UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science
Software Engineering Curriculum

Chingiz Mammadli

A Systematic Review of Traceability in Requirements Engineering of Socio-technical Systems: Industrial Practices and Needs

Master's Thesis (30 ECTS)

Supervisor(s):

Ishaya Peni Gambo, PhD

Tartu 2023

A Systematic Review of Traceability in Requirements Engineering of Socio-technical Systems: Industrial Practices and Needs

Abstract:

Requirement traceability (RT) is useful for determining the effects of differences between various artefacts in the software development lifecycle. RT assures the whole system's quality and accuracy. It also helps the change management process by providing a logical relationship between software development practice deliverables. This thesis examined important methods in current RT activity and classified existing RT techniques based on their domains and requirements. Furthermore, we examined the tools, industrial practices, approaches, and their contributions and trade-offs and highlighted prospective future study topics with important results. Empirical, qualitative, quantitative, agile-based, hybrid, blockchain, natural language processing (NLP), and machine learning (ML) techniques are examined using 46 research studies. The systematic literature review has identified (i) the practices that are popular in literature for RT of socio-technical systems (STSs) and their implications in the industry and (ii) the new trends and approaches that are adopted by the literature and industry. The results and findings from the literature show the variability of requirements and their impact on the overall system. Also, the systematic review (SR) has tried to cover the research gaps of previous literature reviews that only covers the domain-specific literature and case studies. Overall, the SR has analysed the work done in the past five years in the field of RT in STSs. The SR concludes that various practices need to be explored for further studies, which include blockchain, NLP, and ML.

Keywords:

Requirement traceability, Requirement engineering, SLR, Empirical, Industry, Socio-technical systems, Software engineering

CIRCS: P170 Computer science, numerical analysis, systems, control

Sotsiaaltehniliste süsteemide nõuete projekteerimise jälgitavuse süstemaatiline ülevaade: tööstustavad ja vajadused

Lühikokkuvõte:

Nõuete jälgitavus (inglise keeles: requirement traceability; lühend: RT) tuleb kasuks tarkvaraarenduse elutsükli mitmesuguste artefaktide vaheliste erinevuste mõjude määramisel. RT tagab, et kogu süsteem oleks kvaliteetne ja korrektne. Lisaks aitab see muudatuste juhtimise protsessile kaasa, kuna pakub loogilist seost tarkvaraarenduse tavade vahel. Käesolevas lõputöös on vaadeldud olulisi meetodeid hetkel olemasolevas RT tegevuses ning olemasolevad RT tehnikad on klassifitseeritud nende valdkondade ja nõuete alusel. Samuti on uuritud tööriistu, tööstustavasid, lähenemisviise ning nende panust ja kompromisse. Tähtsamate tulemustega on esile tõstetud ka tuleviku uuringuteemad. 46 uurimistöö abil on uuritud empiirilisi, kvalitatiivseid, kvantitatiivseid, väleduspõhiseid, hübriid-, plokiahela-, loomuliku keele töötlemise (inglise keeles: natural language processing; lühend: NLP) ja masinõppe (inglise keeles: machine learning; lühend: ML) tehnikaid. Süstemaatiline kirjanduse ülevaates on tuvastatud (i) sotsiaal-tehniliste süsteemide (inglise keeles: socio-technical systems; lühend: STS) kirjanduses populaarseimad tavad ja nende poolt avaldunud mõju tööstusele ning (ii) uued trendid ja lähenemisviisid, mida kirjandus ja tööstus kasutavad. Vaadeldud kirjanduse tulemused ja leiud näitavad nõuete varieerumist ja nende avaldatud mõju kogu süsteemile. Süstemaatiline ülevaade (inglise keeles: systematic review; lühend: SR) on lisaks püüdnud uurida neid kohti, mis varasemates kirjandusülevaadetes on katmata jäetud, uuringus on hõlmatud vaid valdkonnapõhine kirjandus ja juhtumiuuringud. Uurimustöös on SR kokkuvõtlikult analüüsinud STS-ides viimase viie aasta jooksul RT valdkonnas tehtud tööd. SR-is on järeldatud, et edaspidiste uuringute puhu. tuleb uurida plokiahelat, NLP-d ja ML-i hõlmavaid erisuguseid tavasid.

Märksõnad: Nõuete jälgitavus, Nõuete projekteerimine, SLR, Empiiriline, Tööstus, Sotsiaaltehnilised süsteemid, Tarkvaratehnika

CERCS: P170 Arvutiteadus, arvanalüüs, süsteemid, juhtimine

Contents

List of Tables

List of Figures

1 Introduction

Requirements engineering implicates actions for identifying, detailing, and maintaining a set of requirements for a specific system (Ncube, 2018). Throughout a project development lifecycle, requirements engineering assists in maintaining an evolution of requirements. It concerns all the operations involved in requirements gathering and management. Requirement management is a part of requirement engineering concerning requirements documentation, traceability, and change management.

Requirements traceability assurance throughout the system's lifetime is one of the primary tasks of requirement engineering. According to various standards, such as the Institute of Electrical and Electronic Engineers (IEEE) and Capability Maturity Model Integration (CMMI), traceability is an important activity in requirement engineering.

Requirement Traceability is a helpful technique to determine the consequences of variations among several artifacts of the software development lifecycle. Requirements traceability ensures the quality and correctness of the entire system. It also improves the change management process by specifying a logical relation between the deliverables of the software development practice. Each software development life cycle has to perform various requirements analysis, design, development, and testing. Each phase of the software development process must follow some artifact for successful accomplishment. Therefore, according to (Parizi, 2014), traceability is the ability to follow and describe software artifacts.

Gotel & Finkelstein (1997) first defined requirements traceability as:

"The ability to describe and follow the life of a requirement from its origins, through its development and specification to its subsequent deployment and use, and through periods of ongoing refinement and iteration in any of these phases"

Pre-requirements specification traceability and post-requirements specification traceability are the two main aspects of requirements traceability (Krause *et al*., 2022). The first aspect of requirements tractability, i.e., pre-requirement specification, deals with facets of the life of requirements before the inclusion in the Software Requirement Specification document (SRS) (Krause *et al*., 2022). The main sources for Pre-requirements Specification (Pre-RS) traceability are business rules, stakeholders, and previous documents. Post-requirements Specifications (Post-RS), on the other hand, deals with facets of the life of requirements after their inclusion in SRS (Krause *et al*., 2022). Fulfillment of requirements is ensured in Post-RS traceability. Thus, in Post-RS traceability, system test cases confirm that the requirements are satisfied by the system.

Besides the above-described two aspects of requirements traceability, researchers classified tractability into Backward-From Traceability, Backward-To Traceability, Forward-From Traceability, and Forward-To Traceability (Lin & Chen, 2019). Table 1 summarises the description of all requirements traceability classes. Initially, traceability of requirements was considered for safety and mission-critical systems only. However, due to the proven advantages of requirements traceability, it is adopted now in most types of systems development.

Table 1: Classes of requirements traceability

Requirements Traceability Type	**Description**
Backward-From Traceability	Establish the connection between requirements and their sources, such as business rules, stakeholders, and previous documents.
Forward-From Traceability	Establish the connection between requirements and components of design and implementation.
Backward-To Traceability	Establish backward connection of implementation and design components to requirements.
Forward-To Traceability	Establish the relation between other concerning documents and related requirements, such as manuals covering the functionality of the system.

Well-known advantages of requirements traceability are:

- Impact analysis can be performed efficiently with the help of requirements traceability. For instance, if someone changes requirements, other affected components with this change can be traced easily.

- Requirement sources can be identified accurately with the help of requirements traceability. For instance, whether a group of stakeholders, a single person, or a particular document issues the requirement.
- Verification of test cases can be performed with requirements traceability, e.g., which particular test case validates the specific requirement.
- Overall project's progress can be tracked with requirements traceability, i.e., it can be analysed how many requirements are fulfilled and how many are still to be implemented.
- Requirements traceability helps in tracing the interdependent relationship between requirements and other artifacts.

Another focused area in this systematic review is Socio-technical systems (STS). In this context, STS is referred to as the relationship between technology and people, including software, hardware, data, laws, regulations, and physical surroundings (Maltji *et al.*, 2019). Nowadays, socio-technical systems play a crucial role for enterprises in supporting their critical business processes. This has created a new opportunity and research focus for the requirement engineering community and software development community in general. Integration of requirement analysis with industrial needs has already started to appear in the literature on requirement engineering (Ghozali *et al.*, 2019). Most of the approaches, among them primarily focus on new system development. However, the problem of traceability in Requirements Engineering of Socio-technical Systems in an industrial context is less well understood.

With the enhancement and complexity in the propagation of software systems, it has become complex to capture the requirement traces immediately. Various tools and models have been proposed by researchers regarding requirements traceability. However, according to our knowledge, there is a lack of research studies in the analysis and summarization of the latest tools, technologies, and models available for Traceability in Requirements Engineering of Socio-technical Systems according to the industrial practices and needs.

1.1 Aim and Objectives

1.1.1 Research Aim

The primary aim of this systematic review is to investigate the latest approaches, tools, technologies, and models in the latest studies (2016-2022) related to traceability in requirements engineering of socio-technical systems as per the latest industrial needs and practices.

1.1.2 Research Objectives

The main objectives of this systematic review are:

- Finding the latest studies related to traceability in requirements engineering of socio-technical systems.
- Critical review and analysis of the latest approaches, tools, technologies, and models proposed by the researchers in identified literature.
- Discussion of pros and cons of approaches, tools, technologies, and models proposed by the researchers in identified literature.
- Investigate what is being used in the industry and see if there are any differences between proposed approaches in literature and approaches being used in the industry and why.

We have performed a systematic literature review of the latest tools, technologies, and models in this research work. In this systemic literature review, we aim to recognize the latest studies (2016-2022) in the area of requirements traceability of socio-technical systems as per the latest industrial needs and practices to answer the following research questions:

1.2 Report Organisation

Following the introduction to systematic literature in Section 1, the adopted methodology for this systematic literature review with the definition of search protocol is given in Section 2. Results of the systematic review are managed in Section 3 of this report according to the research questions aimed to answer in this research. Section 4 has the systematic literature review, section 5 has analysis and results of the studies has been evaluated in this section. Section 6 discusses the Research Questions in details and evaluation which are made on the basis of these RQs. Section 7 formulates the research findings which are concluded by evaluating the studies and in the end Section 8 is devised with the specification of future research directions.

2 State-of-the-Art

The software development environment encompasses the technical components of software development and the social aspects (Stojanov et al., 2020). People, policies, activities, and even less tangible items like aims and concepts are all instances of STS' components. Tracing a requirement may be done in two ways: to learn about the analysis phase before including it in the requirements specification or to learn about its usage after it has been elicited and included in the requirements specification (Yoshino and Matsuura, 2020).

2.1 Role of Traceability in Socio-technical Systems

In socio-technical systems, the analysis must address not just individual users and their activities but also how to organise and manage interactions across business structures; moreover, information technology needs should ideally complement industrial objectives. While enterprise models can reflect the problem, they do not provide active direction for determining IT requirements to support industrial activities.

A substantial amount of work is invested to generate a clear requirements specification with the appropriate degree of information. As a result, requirements are regularly revised and amended, new requirements arise from old ones, and some are abandoned. The ability to track the links between requirements is called inter-requirements traceability (Pohl, 2010). Traceability of extra-requirements captures the links between requirements and other artefacts.

2.2 Importance of Requirement Traceability in Socio-technical Systems

In the absence of RT in building STSs, key personnel regularly leave projects, which can result in knowledge loss if traces are not traceable or are collected insufficiently or in an unorganized manner. Additionally, it may lead to misunderstandings, and poor judgments, which might lower the system's overall quality and add to costs and delays.

Even though many requirements engineering challenges are recognised to have their origins in complicated social concerns, few methodologies for analysing socio-technical system needs have arisen.

Ontology-based approaches can manage variable knowledge (Murtazina and Avdeenko, 2019). However, there is insufficient explanation of how quickly knowledge about any single area may be elicited. In brief, no guidelines for a natural start of requirement traceability have been offered thus far. Current industry requirements are diverse, and compatible approaches are desired.

A variety of other techniques are used for RT, including swarm techniques, modern techniques based on machine learning, dynamic techniques to handle outdated links, hybrid techniques (Gul *et al.*, 2021; Islam, Manning, and Cullen, 2021), scalable techniques for catering to project phases and their traceability, and lightweight techniques to overcome frequent changes in requirements (Sultanov and Hayes, 2010). Although approaches based on Information Retrieval (IR) (Zhang, Wan, and Jin, 2016) are prevalent in the business, several studies have shown their mild impact. Also, the agile development (Santos *et al.*, 2018) approaches are already highly popular, but their potential to benefit the industry has not yet been thoroughly explored.

Additionally, in literature and industry, several frameworks are adopted for requirement traceability in socio-technical systems. The RE-Trace framework for requirement traceability in STS combines graph-based and natural language processing approaches to find connections between requirements. Promoting traceability in STSs is possible using the value stream mapping (VSM) approach (Zhang, Wan, and Jin, 2016), which is used to depict the flow of value in a system. The links between various needs are traced through the flow of value in a system, represented in a VSM traceability method as a series of stages or activities.

Further, the socio-technical systems engineering (STSE) method in (Ngowi and Mvungi, 2018) highlights the significance of considering the system's technical and social components while designing and developing socio-technical systems. In order to enable requirement traceability, STSE employs methodologies like use cases and stakeholder analysis. Using the Quality Function Deployment (QFD) approach (Gotel and Finkelstein, 1994), technical requirements for a system or product are converted from those of the client. To enable traceability in STS, QFD focuses on requirement traceability and uses tools like a requirement traceability matrix, and customer needs analysis. These frameworks offer a systematic method for discovering and tracing links between requirements, which can help facilitate RT in STSs. It is crucial to thoroughly assess the project's requirements and select the best framework.

2.3 Research Gaps

It is critical to carefully examine software aspects that will aid in operating and tracking equipment use. Because STSs are embedded in industries and organisations, the following industrial elements influence their design and operation: industrial changes, changes in the structure of requirements and processes, inter-process changes, and so on. Therefore, a diverse literature review is required rather than exploring the one dimension and modeling techniques of traceability in RE of STSs.

2.4 Our Contribution

In addition to assessing the research concepts, study questions, and supporting data, we contribute by offering more examples from various disciplines, tool assistance, and real research outcomes. Additionally, employing extensive literature research, we undertake an empirical, qualitative, quantitative, data-driven, model-driven, and agile-based study for RT from an industry perspective. Because we have studied different and elicited a more significant number of criteria, our approach has made the understanding and modeling of numerous traceability approaches relevant and more accessible for further study.

Remarkably, inconsistencies in user needs may be discovered more readily with tool aid than with more traditional techniques. Some socio-technical models for RE have been developed that define systems in terms of the people, occupations, and roles they play. STSs analysis techniques provide a human-centered investigation of how the technology system impacts society. To analyse such models and research the quantitative, qualitative, as well as empirical part of the RT in RE has been explored, execution and assessment of RT must continue to rely on experience; nevertheless, few analytical approaches for such studies have been released. The previously undertaken SLRs mostly concentrated on the condensed aspect and methodologies for either qualitative or quantitative traceability. The most contemporary trends in traceability are ignored and not fully investigated. This research has studied traceability methodologies and approaches from a dynamic perspective by conducting a thorough review of the work done in the field.

3 Methodology

This research aims to follow the best systematic literature review guidelines. We followed the systematic literature review process proposed by (Kitchenham & Brereton, 2013). We reported review findings for the research studies published from 2016 to 2022. The adopted research method is comprised of 5 steps. In subsequent sections, a detailed description of each phase involved in the planned research method is given.

3.1 Search Protocol Definition

Based on the primary research questions, a review protocol is developed by keeping the best systematic literature review guidelines in mind. The review protocol defines the background of Traceability in Requirements Engineering of Socio-technical Systems, and the research questions discussed ahead. The review protocol includes the key search terms, rejection and selection criteria of the literature, search method, and synthesis of extracted data from research. Some alternative search terms for the topic were also considered, such as "Requirements Specifications Engineering", "Social-Technical Systems", "Tracing Requirements Engineering", and so on. Eventually, the following search string was developed to extract the relevant literature:

("Requirement Engineering") AND ((Traceability OR Trace OR model* OR approach* OR tool* OR techniques* OR analysis* OR needs OR practices OR manage*) OR ("system requirement" or "socio-technical" or industry or needs or systematic or review or requirements or specification or engineering))*

To validate the coverage of the search, the above-denoted string was tested in the index database of Scopus.

3.2 Research Questions

Notably, even though the software engineering (SE) community has recognised the advantages of RT, these advantages are not acknowledged at the industrial level, and the current approaches and practices for RT are still in their infancy. Traceability is either not used or only used when standards demand it. Therefore, we need an extensive and systematic review to determine the extent of work done from both industrial and research/academic perspectives to uncover the gaps, missing links, and unseen opportunities in the requirements traceability of STSs.

Consequently, the primary purpose of this systematic review aligns with those of Babar et al. (2009) and Kitchenham et al. (2010). More specifically, in this thesis, we seek to have clearer and broader perspectives of industrial and research practices and identify how the approaches, models, tools, and techniques used for traceability contribute to solving problems in SE. Also, the thesis further seeks to understand how requirements are traced and managed throughout the software development lifecycle. To accomplish the goal of this study, the following research questions (RQs) were addressed:

RQ1: What practices are proposed in the literature for traceability in requirements engineering of socio-technical systems?

RQ1-i: What is their contribution to solving software engineering problems?

RQ1-ii: What are the benefits/drawbacks of these practices?

RQ2: What practices are being used for requirements traceability of socio-technical systems in the industry?

RQ3: Do practices used in literature overlap with those used in industry? If not, what are the root causes of such discrepancy?

3.3 Defining Inclusion and Exclusion Criteria

Table 2 defines the inclusion and exclusion criteria of this systematic literature review. Some limitations in this criteria are that we only considered the studies in English and some that fulfilled the inclusion criteria but were not accessible.

Table 2: Literature inclusion and exclusion criteria for systematic review

Inclusion Criteria	Exclusion Criteria
• Studies relevant to the domain of requirement engineering. • Studies investigating the traceability in requirement engineering of systems. • Studies reviewing the tools, technologies, approaches, and models for requirement engineering traceability. • Studies discussing the challenges in requirement engineering traceability. • Studies discuss the requirement engineering processes for socio-technical systems. • Studies discussing the industry requirements and practices for requirement traceability. • Studies published in the year 2016 to 2022.	• Short studies or positions • Studies in languages other than English • Studies published before 2016 • Studies covering the generic aspect of traceability and modeling.

3.4 Systematic Search

The search string defined was tailored to fit the following four research databases:

- IEEE Xplore Digital Library
- Scopus
- ACM Digital Library, and:
- Elsevier Science Direct

Abstract, keywords, and titles were searched in the metadata fields with slightly different queries due to search possibilities differences in research databases. A set of 554 research studies was acquired in the initial search. We applied the inclusion and exclusion criteria given in Table 2 to retrieve the relevant literature only. Three filters were used in this regard.

- Filter 1: The extracted research studies were analysed by parsing their titles, abstracts, and conclusion. Irrelevant research studies were eliminated upon applying this filter.
- Filter 2: Keep a check for repeated studies and duplicate literature.
- Filter 2: Keeping in mind the selection criteria, the full content of the research studies from filter 1 was looked at.

Table 3 shows the search results and the number of research studies selected per phase.

Table 3: Number of research studies selected per phase

Digital Research Library	**Papers Retrieved**	**After Filter 1**	**After Filter 2**	**After Filter 3**
IEEE Xplore Digital Library	95	26	26	11
Scopus	213	42	42	16
Elsevier Science Direct	114	24	24	9
ACM Digital Library	132	20	20	7
Manual Search		4	4	3
Total	**554**	**116**	**116**	**46**

3.5 Manual Search

A manual search was also conducted to find the grey literature, such as dissertations, white papers, unpublished research studies, enterprise documents, and so on. According to the Multivocal Literature Review Approach, grey literature is also important (Garousi, 2016). A manual search on Google scholar was also performed to extract more relevant literature that was not indexed by ACM digital library, IEEE Xplore, Elsevier Science Direct, and Scopus. After completing the manual search, we retrieved 3 more relevant research studies, as shown in Table 3 above.

3.6 Quality Assessment

To enhance the quality of this systematic literature review, high-quality research studies are considered. To filter out the low-quality material, an assessment of quality was performed. After this test, some content was filtered out.

Figure 1 illustrates the search process adopted in this systematic literature review.

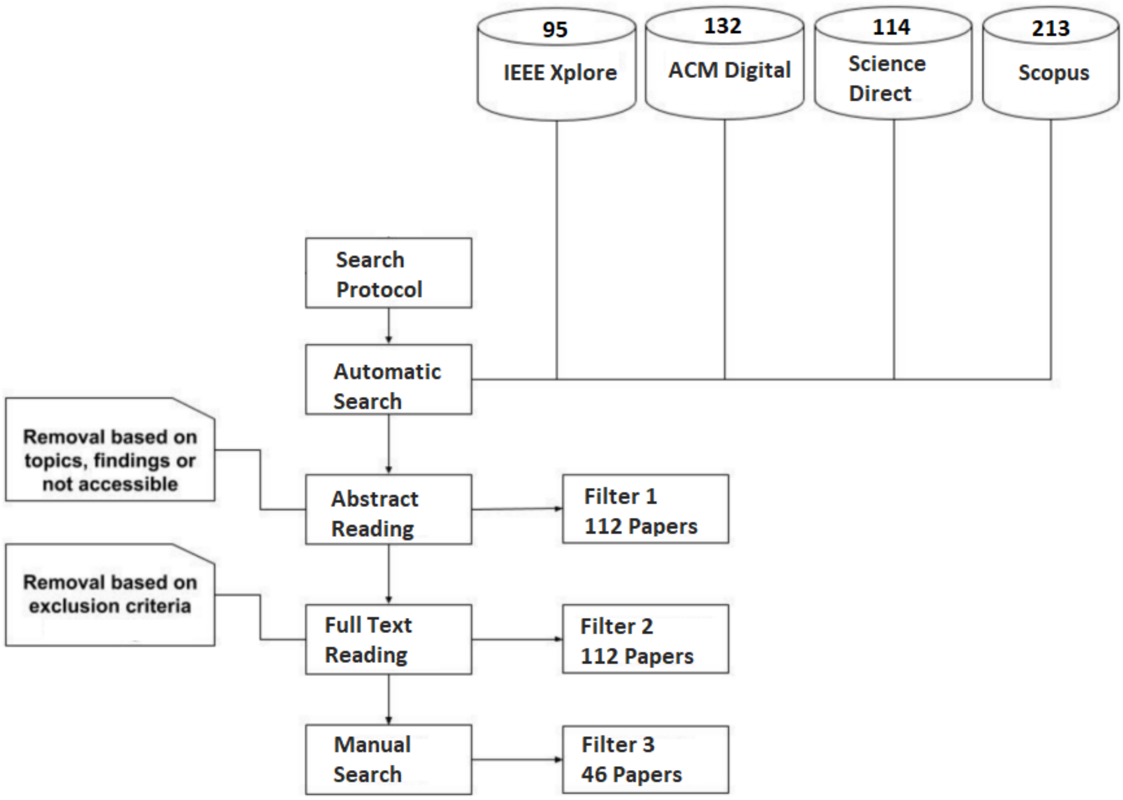

Figure 1: Search process adopted in this systematic literature review

3.7 Extraction of Data

Data is extracted while reading from the sources that provide relevance to the research objective. The papers formulated the strategies for traceability in the RE and the concerns it modeled. The data is extracted with reference to the year of publication, author, and methods. The tools and techniques are also taken into account. The data extraction is modeled below in Table 4 to give insight relevant to the research questions.

Table 4: Data extraction

Source	Relevance	Research question
Publishing date	Article link	RQ1, RQ2, RQ3
Author name	Article link	RQ1, RQ2, RQ3
Methodology used	Author and taxonomy	RQ1
Literature practices	Industrial and literature perspective	RQ1, RQ2
Setbacks	Literature review	RQ1
contribution	Problem and solutions	RQ1
Techniques	For RE	RQ2 and RQ3
Tools	For RE	RQ2

3.8 Threat Analysis for Research Validity

The section analyses the factors that can propose a threat to the validity of the research presented.

- **Access to the material:** The selected areas for gathering material sometimes cannot be accessed. The issue was handled by sorting and filtering the articles with respect to various parameters and devising the research approach and objective afterward.
- **Limited papers for review:** The traceability practices from the past decade have now become conventional and outdated in some sense; the research has tried to build a coherent approach to maintain the links between the previous approaches as well by selecting papers bout methodologies and techniques for traceability in RE and discussing the new approaches to answer the research questions in contrast with the previous ones.
- **Criteria for research reliability**: Research reliability is the important factor for threat validation of this research. The research questions are formulated by considering the fact that what is the key objective for our research and how we will be able to achieve that objective. The structure of the report is devised and formulated by keeping in account the strategy discussed.

The section 4 formulates the SLR to analyze the research questions that are raised in the report.

3.9 Analysis

For an overview total of 46 papers have been selected. The analysis of the paper with respect to the year has been shown in the graph in Figure 2. The studies are selected by using search strings and papers which are published in the past 5 years were taken into account.

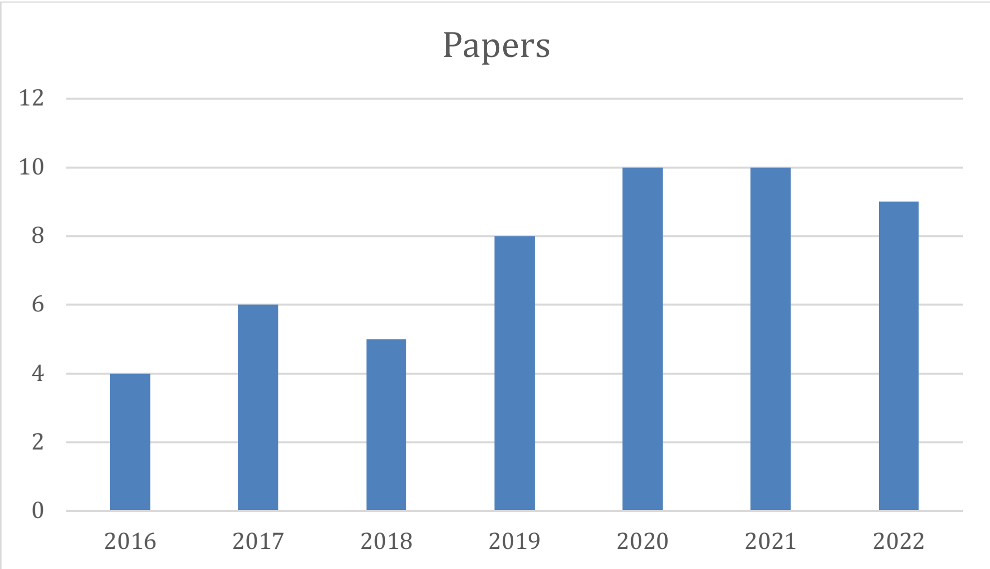

Figure 2: Papers reviewed w.r.t year

4 Systematic Literature Review

The section discusses the techniques used in the RE of STSs. The techniques that are evaluated are empirical, qualitative as well as quantitative. The research papers selected are thoroughly analysed in this section with respect to the research questions. The approaches used in the industry and literature are compared and discussed. The tools that are evaluated and studied in the literature are also discussed and thoroughly studied. The criteria for effectiveness and usage is decided with scores of 1, 2 and 3 for the techniques and tools that are visibly adopted frequently in the industry or literature. The score of 1 is given fir the least adoptive techniques in the reviewed literature, score of 2 is given to moderately used tool and techniques and score of 3 is given for the widely adopted tools and techniques and also currently being used for the solutions of RT issues in the literature as well as the industry. The quality criteria is usually followed in industry 4.0 (Chiarini, 2020) with score or grade and for effective SLR (Paul *et al.*, 2021). The approach is adopted here to evaluate the effectiveness of the literature selected for the dissertation.

4.1 Research Question 1: What practices are proposed in the literature for traceability in requirements engineering of socio-technical systems?

The research question is formulated in light of the approaches that are frequently discussed in the literature. The literature of past 5 years has been studied and hierarchy has been made about the mostly used techniques in past, present and the future. The RT in STS is vital and various authors gave adopted diverse methodologies for an effective traceability. The sub-questions to the RQ1 are devised. The tradeoffs and contributions are also discussed in this section.

- **What is their contribution to solving software engineering problems?**
- **What are the benefits/drawbacks of these practices?**

The overview of traceability requirement techniques in socio-technical systems has been done in this section, as RQ1 has two sub-questions, which will be discussed in the section below.

It has been suggested that traceability can be used to reduce the impact of changes in the requirements (Sutcliffe, 2000). Traceability helps in the repair and the effective analysis of the response of the system to the changed requirement. Various methods have been proposed over the years for the traceability of RE such as empirical , ML, quantitative, qualitative and agile techniques (Torkar et al., 2012). Conventional ML techniques are insufficient for RE; therefore, NLP has paved the way for traceability and RE (Zhao *et al.*, 2021). The static traceability approach has been used. It used source code and textual structures to figure out the traceability links in the high- and low-end artifacts. Formula-based approaches have been used by (Sultanov and Hayes, 2013) to recover the traceability links by the researchers by taking the version and history into account, the repositories play an important role in it. The drawback of this (Sultanov and Hayes, 2013) research was that the two files could be co-dependent on each other, and changing one can impact the other, but the research fails to address such codependency.

The (Rubasinghe, Meedeniya and Perera, 2018a) proposed tracing requirements across the development process by providing traceability between artefacts in source code, analysing and tracing unit test scripts, and building scripts. When evaluated against the core network analysis, the approach performed with 71% accuracy. The research proposed by (Rubasinghe, Meedeniya and Perera, 2021) on software traceability reveals drawbacks, including a restriction to a small number of artifact kinds, a lack of automation, and an inability to handle continuous integrations. Using a prototype called SAT-Analyser, this study seeks to solve such problems by supporting heterogeneous artifacts' traceability in DevOps contexts. This study contributes to the suggested traceability process model, which includes artifact change detection, change effect analysis, and change propagation.

The authors (Khlif, Kchaou and Bouassida, 2022) proposed a scenario-based technique to design a control structure to build traceability between artefacts to lay a strong basis for textual description in the traceability. The technique worked well in terms of precision and recall. Estimation of the Number of Remaining Links (ENRL) proposed by (Falessi *et al.*, 2017) a method for estimating the number of surviving positive connections in a prioritised list of possible traceability links created by a natural language processing-based recovery approach. The authors evaluated the accuracy of their technique on three datasets using a variety of machine learning classifiers and NLP algorithms, as well as traceability linkages across different types of software artefacts, including requirements, built use cases, design patterns, source code, and test cases. The authors (Zogaan *et al.*, 2017) recommended three ML sub-techniques for traceability in RE: manual analysis, mining, and large data analysis. The suggested approaches' fundamental tasks were analyzing and mining large-scale repositories to establish traceability between connections and retrieve lost trace linkages.

The authors (Guo, Gibiec and Cleland-Huang, 2017) provided three query augmentation strategies for identifying query mismatch in data retrieval, the first of which trains a classifier to replace terms learned from a training set of regulation-to-requirements trace links for the original question. The second replaces the original inquiry with web mining phrases, while the third augments query terms with a manually built domain ontology. All three techniques were evaluated using security criteria linked to ten healthcare-related requirements descriptions. The categorising strategy yielded the most accurate findings. The tradeoff was faced regarding cost and performance on some parameters.

The ontology techniques by (Alkhammash, 2020) and (Mosquera *et al.*, 2022) support automated reasoning for tracing the artifacts and establishing new traces. The blockchain-based approach to support agile methodologies is proposed by (Lenarduzzi *et al.*, 2018). The validity of the outcomes is performed by the smart contract to monitor the development of the entities throughout the process. The authors (Demi, 2020a) developed a technique for modeling varied organisational backgrounds, opposing aims, and organisational borders, which contribute to trust concerns that hamper traceability implementation in such situations. The authors suggest a blockchain-enabled architecture for requirements traceability in this study. It offered a comprehensive and trustworthy view of artefacts and traceability connections, provided an incentive mechanism for developers of traceability links, verified the authenticity and quality of traceability links through voting mechanisms, facilitated understanding of traceability information through query services, and enabled the interactive graphical representation of traceability links.

The author (Adjepon-Yamoah, 2016) proposed cloud-based Reactive Middleware that offers services for managing Global Software Development (GSD) projects in the direction of reliable change management in RE and traceability. The traceability of requirements necessitates the exploration of technological artefacts as well as social structures. Also, the research by (Cepulis and Niu, 2018) studied the socio-technical graphs to describe the important linkages a patch should embody to help with requirements traceability. An approach based on spreading activation extracts a relevant collection of these interactions as a patch. The results suggest using this approach produces meaningful patches with less extraneous information. Table 5 discusses the approaches of the authors mentioned in the table and the criteria it adopted, giving grades 1, 2, and 3 for the normal, more, and most adapted.

Table 5: Techniques and usage

Techniques	**Reference**	**Issue resolved**	**Usage in Literature**
Latent Semantic Indexing (LSI)	(Mäder, Olivetto and Marcus, 2017)	Links traceability	3
Space Vector Model	(Singh and Kumar, 2020)	Empirical validation for traceability	2

ANN and LSI	(Wang *et al.*, 2018)	Identify polysemous	3
OWL ontology	(Murtazina and Avdeenko, 2019; Alkhammash, 2020)	RE and traceability	3
Socio-technical graphs	(Niu *et al.*, 2018)	Traceability	3
Onto-Trace	(Mosquera *et al.*, 2022)	RE and traceability	3
RNN(bidirectional)	(Guo, Cheng and Cleland-Huang, 2017)	Traceability	3

The visual representation of the discussed techniques is shown in Figure 3.

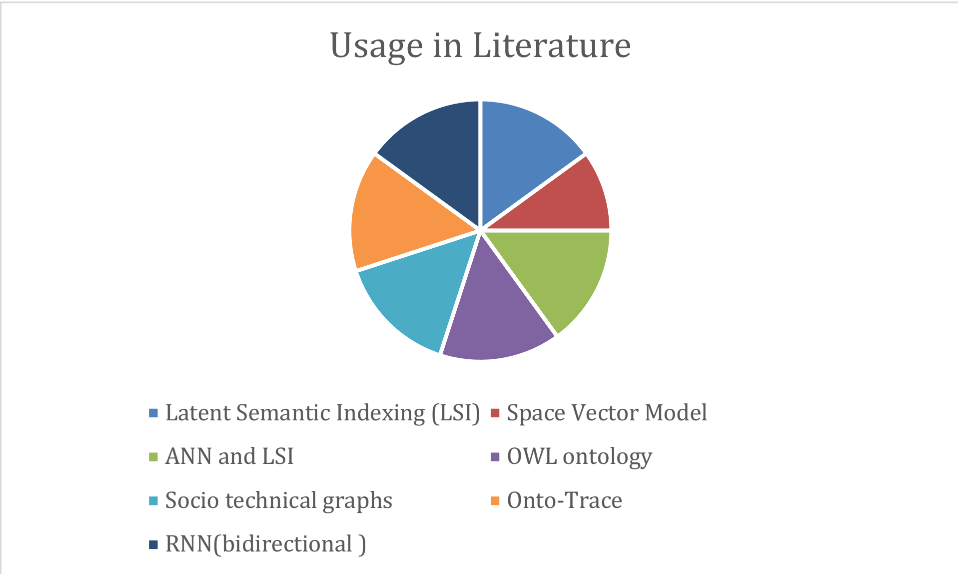

Figure 3: Techniques and usage

The empirical approaches followed by NLP and ML are used in literature the techniques performed well in the context of analysis. Various techniques have been proposed for traceability by taking big data into account. Big data analysis requires cost-effective methods and a good score with respect to precision and recall. The blockchain-based solutions are proposed to model the security of RE artifacts. Because of the diverse set of scattered stakeholders and a range of tools, such features complicate integration, collaboration, coordination, and confidence and limit accurate and trustworthy requirement traceability. The blockchain-based approaches are the most recent in such a manner(Demi, 2020a). Several research studies have used XML and UML to trace RE in socio-technical systems

(Montecchi and Gallina, 2017; I. Rubasinghe, Meedeniya, and Perera, 2018b). Various studies have used UML-based techniques for modeling and tracing the artifacts in RE (Sena Marques, Siegert and Brisolara, 2014; Min, 2016; Montecchi and Gallina, 2017).

The author proposed a metamodel that represents NFRs and their relationships and is independent of any programming paradigm. The author also provided a solution that uses XML-based metamodel representations and XQuery queries to express tracing information. The consistency of the approach is yet to be tested (Kassab, Ormandjieva and Daneva, 2008). The scalability of the blockchain is currently an area under deep research, and it is yet to be adequately explored. The criteria for effective problem-solving have been devised in Figure 4 and modeled in Table 6.

Table 6: Effectiveness criteria

Techniques	**Reference(s)**	**Effectiveness criteria**
ML	(Carvalho *et al.*, 2019)	3
SAT-Analysis	(Rubasinghe, Meedeniya and Perera, 2021)	2
Natural Language	(Falessi *et al.*, 2017)	3
Agile	(Santos *et al.*, 2018) ,(Murtazina and Avdeenko, 2018; Ågren *et al.*, 2019; Hidayati and Rochimah, 2020)	3
IR based	(Capobianco *et al.*, 2013; Ali *et al.*, 2015)	2
RTM	(Tomlinson, 2017; Murtazina and Avdeenko, 2019; Gul *et al.*, 2021)	3
XML and UML	(Khan and Salah, 2018; Saravanan and Bama, 2019; Gul *et al.*, 2021)	3

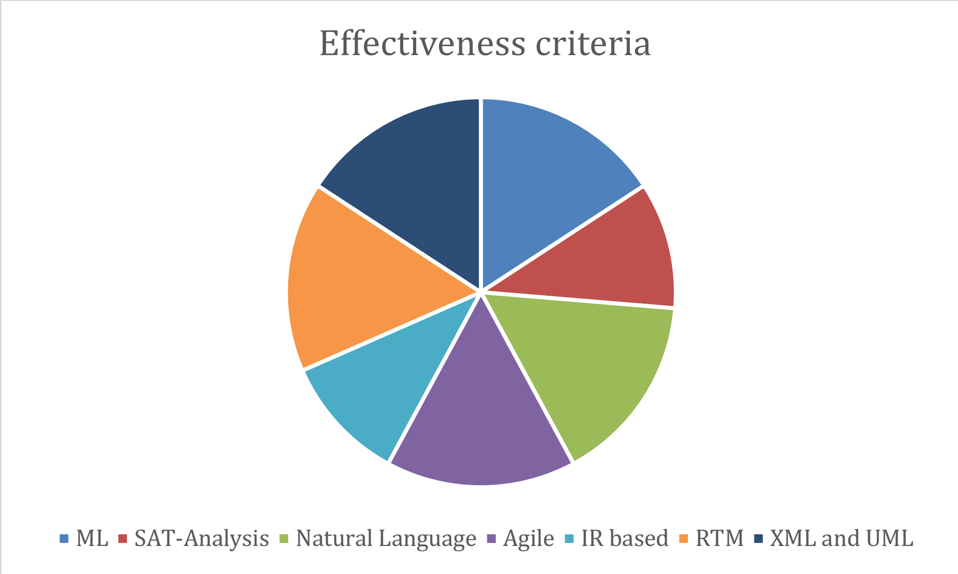

Figure 4: Effectiveness of techniques in literature

The requirement dependency model explains the dependency of the extracted features on the outcomes and their probability of occurrence. The meta-model of traceability has five subdivisions: planning the traceability data, collecting the data, and assessing the traceability data. It suggests a more generic approach to the traceability of the artifacts. Blockchain-based traceability model is the recent approach to keep the artifacts secure and traceability unchanged (Tufail *et al.*, 2018). The domain-specific model deals with the semantics of the approach. Tracking non-functional requirements have mostly been ignored. (Mader, Gotel and Philippow, 2009).

In general, data mining-based tracing approaches usually generate interesting trace connection patterns from artifact pairings(Zhao *et al.*, 2021). For example, the affinity mining approach extracts compatible data from pre-existing needs linkages and gives information. This data comprises domain glossaries and relationships between term-occurring patterns and trace linkages, and it aids in creating new traces (Iqbal, Elahidoost and Lúcio, 2018; Niu *et al.*, 2018; Dalpiaz and Niu, 2020; Charalampidou *et al.*, 2021a). Agile-based approaches need an automated tool for traceability, creating links to the user stories with the backlogs and managing the track of the approaches studies (Murtazina and Avdeenko, 2018; Bandara *et al.*, 2019; Alkhammash, 2020; Mosquera *et al.*, 2022). Rule-based methods use languages like Z, Prolog, and X-Query to represent Unified Modeling Language (UML) and SRS diagrams. To trace links between various RE artifacts. Combining the various high-end algorithms with basic IR approaches can create an automated phenomenon for RE in industry, and this lays the foundation of swarm methodology. (Victoria and Caraballo, 2020; Adithya and Deepak, 2021). Table 7 summarizes the percentage of models reviewed in the literature.

Table 7: Models reviewed in the literature

Models discussed in the literature	**References**	**Reviewed literature %**
Meta model of traceability	(Montecchi and Gallina, 2017; Dassisti *et al.*, 2019; Al-Dhaqm *et al.*, 2021)	15
Blockchain-based traceability model	(Demi, 2020a, 2020b; Demi *et al.*, 2022)	20
Process Traceability model	(Akman *et al.*, 2016; Arbain *et al.*, 2020)	15
Domain-specific Semantic model	(Al-Dhaqm *et al.*, 2021)	10
Traceability Information model	(Nair, De La Vara and Sen, 2013; Gul *et al.*, 2021)	10
Machine learning-based model	(Iqbal, Elahidoost and Lúcio, 2018; Pessoa *et al.*, 2022)	30

The literature analysis of the techniques has been done in the section in which 60 percent of the techniques are related to NLP, ML, and blockchain, and the rest of the techniques are covered in the rest of the sections (5 and 6) in detail. The current techniques have used multiple traceability approaches and modeled the pros and cons discussed in table 8.

4.1.1 Contributions and Tradeoffs

Table 8: Contributions and tradeoffs

Approach	**References**	**Tradeoff**	**Problem solved**
Empirical approaches	(Singh and Kumar, 2020)	Cost-effective but lags in a few performance metrics	Traceability in RE
ML	(Wang *et al.*, 2018; Zhao *et al.*, 2021)	Performed well in precision and recall but lags for big data analysis if the dataset is not appropriately trained.	Traceability in RE, link generation, trace links, modeling, and management.

NLP	(Zhao *et al.*, 2021)	Popular and effective	Traceability, text analysis, query modeling
Blockchain-based Traceability approaches	(Demi, 2020a)	A popular trend for security but poses issues with scalability	Security and tracing

The proposed methodologies have been further divided into the approaches they have proposed for the industry.

4.2 Research Question 2: What practices are being used for requirements traceability of socio-technical systems in the industry?

There are numerous approaches that are regularly utilised in the industry for requirements traceability in STSs. These approaches may be used to aid in the discovery and monitoring of requirements linkages in a software development project. This section is formulated to analyse the practices used in the industry for RT. In addition, we present the tools and methods which are frequently adopted for effective results.

4.2.1 Analysis of Models and Approaches:

The section explains the tools, approaches, frameworks, techniques, methods, methodologies, models or modeling languages, and other activities are among them from the industrial usage perspective.

Modern traceability methods are not applicable without the tools. It is not cost-effective and takes time. The tools for traceability are mostly off the market or as presented as complete requirements management package. Seeing such minimal interest in security issues about trace artifacts is odd. Given the implications of such modifications, security methods such as rule-based access control are required for traceability to restrict who can edit and trace data (Batot, Gérard and Cabot, 2022). Various companies used Java-based platforms such as JIRA to maintain the record of all the requirements and maintain tracking. JIRA can keep track of all requirements and integrate all the traceability, track backs, and changes. The entire lifecycle is maintained and is effective in tracing. The only drawback is that there is no direct way of managing traceability.

4.2.2 Models

Table 9: Models for traceability in RE

Dependency model	**Reference**	**Usage in industry**
Meta model of traceability	(Montecchi and Gallina, 2017; Dassisti *et al.*, 2019; Al-Dhaqm *et al.*, 2021)	2
Blockchain-based traceability model	(Demi, 2020a, 2020b; Demi *et al.*, 2022)	3
Process Traceability model	(Akman *et al.*, 2016; Arbain *et al.*, 2020)	2
Domain-specific Semantic model	(Al-Dhaqm *et al.*, 2021)	2
Traceability Information model	(Nair, De La Vara and Sen, 2013; Gul *et al.*, 2021)	1-2
Machine learning-based model	(Iqbal, Elahidoost and Lúcio, 2018; Pessoa *et al.*, 2022)	3

The traceability model, as discussed in Table 10, is used to trace the types of artifacts and the relations of the artifacts with each other (Esparteiro Garcia and C. R. Paiva, 2016). Various blockchain-based traceability models are applied in the industry (Ho *et al.*, 2021). There is a need for effective RE because software-intensive systems are growing more significant in industrial projects (Demi, Sanchez-Gordon, and Colomo-Palacios, 2021) and because more creative and high-quality systems must be released to market rapidly. It develops the need for efficient requirement engineering (Demi, Sánchez-Gordón and Kristiansen, no date). Figure 5 shows the review of models with respect to the percentage studied, and Table 9 shows the usage of the model in the industry on criteria 1,2 and 3 for normal, more used, and currently being used, respectively as explained in section 4.

Table 10: Traceability methods used in socio-technical systems

Methods	Reference	Usage in industry
IR-based tracing methods	(Capobianco *et al.*, 2013)	1
Data-mining based methods	(Bandara *et al.*, 2019)	3
Rule-based methods	(Gul *et al.*, 2021)	2
Swarm based methods	(Sultanov and Hayes, 2010)	3
Agile oriented	(Murtazina and Avdeenko, 2019; Hidayati and Rochimah, 2020)	3
Hybrid methods	(Gul *et al.*, 2021; Islam, Manning and Cullen, 2021)	3
RTM	(Murtazina and Avdeenko, 2019; Gul *et al.*, 2021)	3

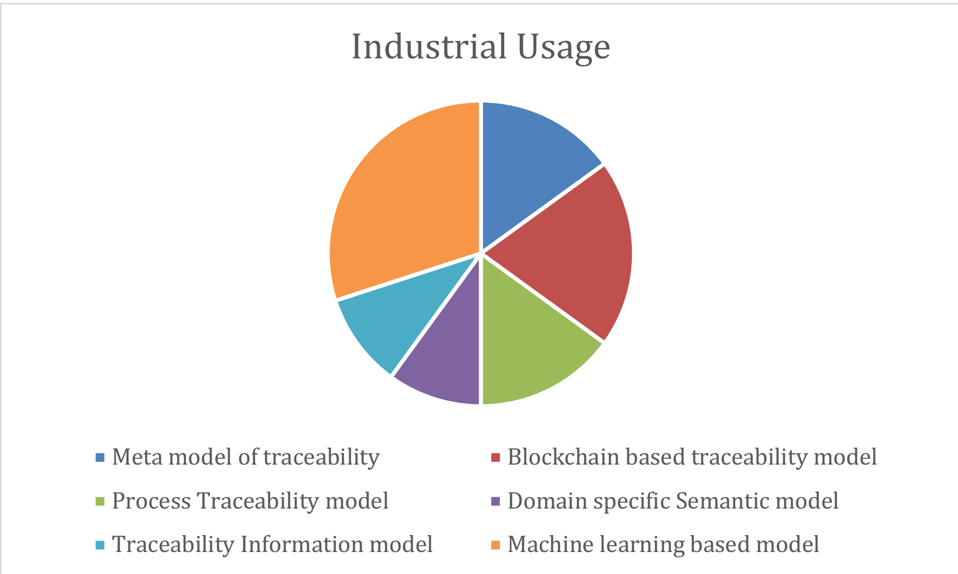

Figure 5: Pie chart for percentage analysis

Documents for requirements and other artifacts are usually textual, and information retrieval techniques have become the most preferred methods for generating trace links in IR-based tracing methodology (Zhang, Wan and Jin, 2016). Queries and documents are commonly used to refer to requirements documents and target artifacts(B. Wang *et al.*, 2018). The usage of methods in the industry, as shown in Table 10, are given the criteria of 1,2 and 3 for normal to more and primarily being used, respectively. It is feasible to connect a need to several implementation artifacts and an artifact to multiple requirements using the traceability tool (Mustafa and Labiche, 2017). A simple traceability information model comprises two sorts of entities: traceable objects and traceability interactions between them. It also specifies which sorts of artifacts are to be tracked to which categories of related artifacts and through what type of traceability links. Figure 6 shows the percentage of the literature and techniques reviewed in the study with respect to the methods.

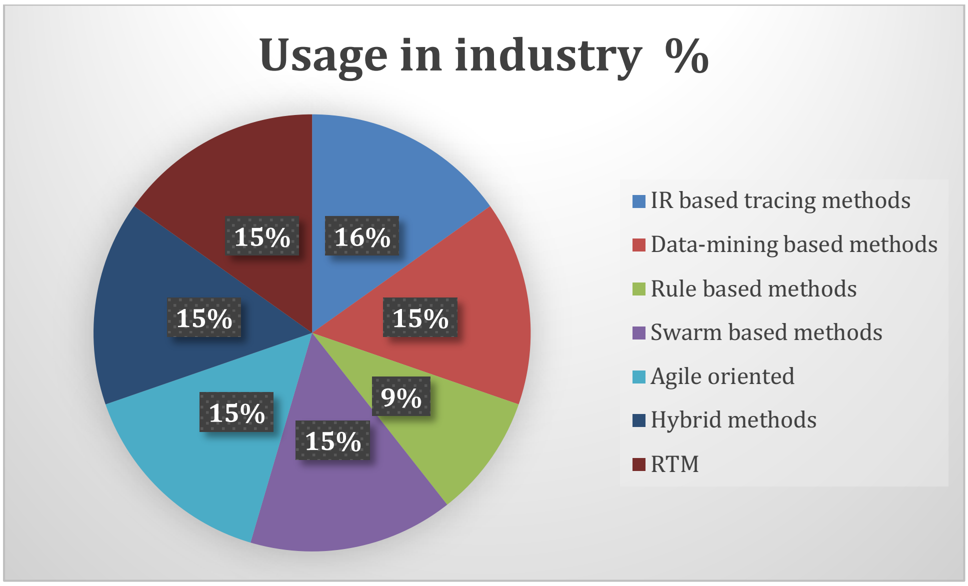

Figure 6: Model comparison based on the current usage

4.2.3 Tools

Table 11: Tools for modeling traceability

Tools	Solutions	Reference	Usage in Industry	Reviewed Literature %
XML	Eliminating the poor tool support	(Hayes, Dekhtyar and Payne, 2018; Corallo *et al.*, 2020; Gul *et al.*, 2021; Batot,	3	3

		Gérard and Cabot, 2022)		
UML	Traceable artifacts and traceable relations of artifacts	(Santos *et al.*, 2018; Yoshino and Matsuura, 2020; Khlif, Kchaou and Bouassida, 2022)	3	4
TFS	Design and traceability of RE	(Ali *et al.*, 2015; Khlif, Kchaou and Bouassida, 2022)	2	2
JIRA	Requirement management and traceability	(Akman *et al.*, 2016)	3	2
EA	Source code traceability	(Tufail *et al.*, 2018; Mengist, Buffoni and Pop, 2021)	2	2
SVN	Source code traceability	(Akman *et al.*, 2016)	2	2
JAMA	Requirement management and traceability	(Tufail *et al.*, 2018; Mengist, Buffoni and Pop, 2021)	2	2
SAT analyser	Traceability of artifacts	(Rubasinghe, Meedeniya and Perera, 2018)	3	1

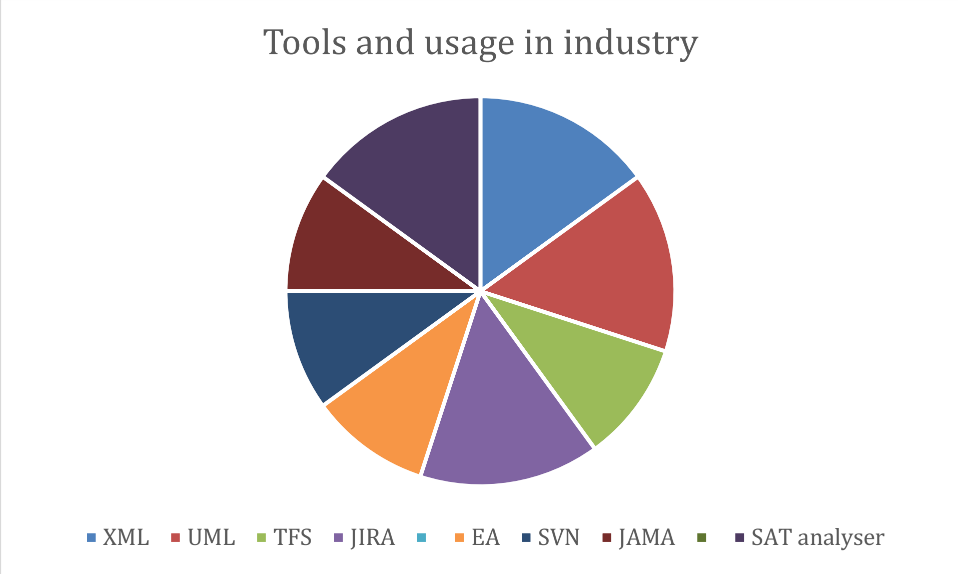

Figure 7: Tools and usage

The tools for traceability are adopted for the effective RE traceability analysis in the system Keeping requirements traceable necessitates the use of well-designed development tools and techniques. Requirement traceability allows clients' requirements to be tracked, and the linkages between requirements, design decisions, and code can be retrieved to reuse previously built products or analyse the consequences, which helps businesses to reduce costs (Akman *et al.*, 2016). UML diagrams are used for sequentially tracing, documenting, and building artefacts. We may find a context element that may be influenced by modifications and visualise it in a diagram. As a result of changes to the context element, we may identify elements connected to it via any dependency relationships that may need to be modified, either directly or indirectly (Bauer and Herder, 2009; Pessoa *et al.*, 2022). The literature has been studied to extract the tools currently being used based on the more used and mostly used with scale 2 and 3, respectively, in Table 11, and usage is shown in Figure 7.

4.2.4 Techniques

Table 12: Techniques in traceability

Techniques	**Reference**	**Status of usage**
NLP based techniques	(Zhao *et al.*, 2021)	3
Block-chain based techniques	(Demi *et al.*, 2022)	3

UML based techniques	(Yoshino and Matsuura, 2020; Khlif, Kchaou and Bouassida, 2022)	3
IR based Techniques	(Ali *et al.*, 2015; B. Wang *et al.*, 2018)	2
Machine learning-based techniques	(Iqbal, Elahidoost and Lúcio, 2018; Santos *et al.*, 2018; Dassisti *et al.*, 2019; Pessoa *et al.*, 2022)	3
Agile development techniques	(Murtazina and Avdeenko, 2019; Hidayati and Rochimah, 2020; Maro *et al.*, 2022)	2

Table 12 summarizes a number of techniques popular in literature and currently being used in the industry with respect to the usage, as shown in Figure 8.

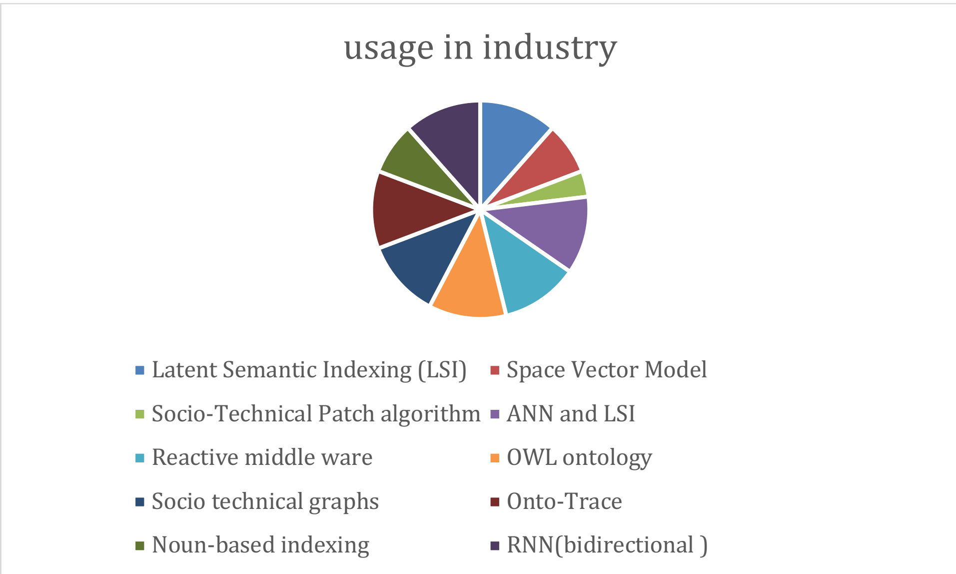

Figure 8: Summary of a few techniques

4.3 Research Question 3: Do practices used in literature overlap with those used in industry? If not, what are the root causes of such discrepancy?

System analysts generally utilize requirement management tools to do manual requirements tracing. To create the Requirement Traceability Matrix, they manually analyse each pair of requirements listed in the tools. Traceability analysis is a feature of most contemporary requirement management tools, including JIRA and Rational DOORS (Torkar *et al.*, 2012). With so many variables, the manual method is time-consuming and prone to errors.

Furthermore, such domain expertise may be lost during the system development and evolution life cycle due to requirement alterations, dispersed teams, or system reorganisation. Numerous automated strategies have been proposed in the literature to deal with this issue. In the last ten years, the industry has shifted to various integrated automated and hybrid systems built on efficient requirement tracing tools. In industry and literature, several traceability techniques may overlap or be used in conjunction with one another. The requirement traceability matrix (Torkar *et al.*, 2012) shows the relationships between different requirements, such as functional, non-functional, and quality requirements. The matrix can be used to trace the relationships between different requirements and to ensure that all requirements are being addressed in the project.

Linking and tagging (W. Wang *et al.*, 2018) is a technique that involves attaching a unique identifier to each requirement and linking related requirements together. This allows for easy tracking and management of requirements as the project progresses.
Use cases (Santos *et al.*, 2018) describe the interactions between a system and its users and can be used to trace the relationships between different requirements.
Change management is a robust process that can help ensure that requirement changes are properly tracked and managed throughout the project.
Using Version Control (Mukherjee *et al.*, 2011) software can help track changes to requirements and facilitate collaboration among team members. Several tools are available that can help with requirement traceability, such as requirement management software and project management software. These tools often include features such as version control (Krüger *et al.*, 2019), linking and tagging (W. Wang *et al.*, 2018), and change management (Jayatilleke and Lai, 2018) to support requirement traceability.
Several tools are frequently employed in business and literature to enable requirement traceability in requirement engineering.

A tool to track and manage requirements is JIRA (Akman *et al.*, 2016), a project management application used to assist requirement traceability. It has capabilities like linking and tagging, version control, and change management.

A tool for requirements management that is specially made for requirement engineering is called DOORS (Demi, 2020b), and Rational Requirements Composer by IBM ("IBM products," n.d.) is a requirements management application that supports requirement traceability with capabilities including linking and tagging, version control, and change management. ReqPro (Abo, 2010) is a requirements management application that supports requirement traceability. CaliberRM (Martin, n.d.) is a requirements management platform that supports RT in industrial systems.

The effectiveness, dynamics, and needs of the system and the purposes for which it is being utilised are taken into account while choosing techniques and methods. The use cases in particular scenarios are frequently discussed in literary practices. When it comes to industry, extensive and varied criteria and artefacts are developed. It is crucial to properly assess the project's requirements and select a technology to facilitate RT.

There are several reasons why industry may not adopt traceability approaches from literature frequently:

Complexity: Some traceability methods, including those based on machine learning, may be difficult to apply and need specialist knowledge. This could make it challenging for businesses, especially those with limited resources or modest budgets, to use them. While significant research has been done on using machine learning to RT, these techniques are still developing and are not yet commonly deployed in the industry. Given the continued development of machine learning techniques, requirement traceability is expected to be used more frequently in the future (Fucci, Alégroth and Axelsson, 2022).

Lack of understanding and standard: Certain traceability techniques might not be well-known or broadly accepted in the industry, especially if they are new or have not yet been widely applied in the field. Adopting these strategies may be challenging for corporations as a result. A lack of standardisation can make it challenging for companies to select the optimal traceability strategy for their purposes since there may not be a clear standard or industry consensus on the best approach (Fournier and Petrillo, 2020).

Cost: Acquiring and maintaining specialist software or other resources, which certain traceability approaches may require, can be expensive for enterprises. Organizations may find it challenging to defend the expense of these methods as a result(Li, Vaughn and Saiedian, 2002).

In general, it is crucial for businesses to thoroughly assess the requirements of their projects and select traceability methods that are appropriate for their goals and available resources.

5 Analysis and Results Findings

We discuss the review's results in this section in light of the three main research questions and each of their sub-questions. The results can help us comprehend the general state of the RT research community and pinpoint the difficulties facing RT activities in the context of suggested methodologies. The quality of each tracing approach has been evaluated, together with the extent to which industry practitioners may use it in industrial practice, from 2016 to 2022. The findings will be identified and analysed in this section.

5.1 RQ1: Implications and Findings

The papers we analysed in our SLR to determine the answer to research question 1 were based on rigorous literary content and methodologies. To answer this topic, the study is widely separated into the researchers' traceability methodologies for the literary setting. The approaches are searched from a larger perspective, and methodologies for each method are studied. The analysis discovered that empirical procedures dominated the literature over the last 5 years, with about 50% - 60% of the strategies being hybrid, integrated with empirical models, or exclusively based on them. The Information Retrieval based methods were popular in the last decade as they involved modeling techniques that were feasible and cost-effective to use yet takes time and had their tradeoff when compared to fully or semi-automated traceability techniques. Agile development techniques are still popular and extensively used to integrate various ML and NLP techniques, covering 60 % of the RQ1 and RQ2. The trend of blockchain techniques for traceability in integration with various empirical techniques is covered in the literature with most recent research (Lenarduzzi *et al.*, 2018; Ho *et al.*, 2021; Demi *et al.*, 2022) as the technology is thriving and various case studies are devised based on these methods.

5.1.1 Contribution of Practice

From the literature, we have analysed that agile methods and techniques place emphasis on adaptability and quick iteration. They are intended to enable quick alterations and tweaks to the development process in response to new information or changing needs.

One of the primary advantages of agile methodologies is their emphasis on traceability, or the ability to track and document a project's progress and history. Traceability is frequently achieved in agile development through numerous tools and methodologies. These techniques are well suited for collaborative approaches, as shown in (Lenarduzzi *et al.*, 2018; Santos *et al.*, 2018; Murtazina and Avdeenko, 2019) for traceability and system development. The IR-based methods have been found effective in domain-specific and less diverse environments. The integrated IR methods have shown better results for case studies (Ali *et al.*, 2015; W. Wang *et al.*, 2018). The requirements with variable nature are traced with the NLP and ML approaches, and these models have shown better results for rule-based approaches (Zhao *et al.*, 2021). These approaches have covered the maximum part of the literature review, as shown in Table 7, and their effectiveness criteria, as shown in Figure 4.

5.1.2 Findings for the Tradeoffs of Practices

As shown in Table 8, the tradeoffs of the approaches and methods the ML approaches have shown better results when the LSI and text classification techniques are applied (Zhao *et al.*, 2021). As ML techniques depend on the dataset and its credibility, these techniques are yet to be explored in the industry context. The empirical procedures that were modeled using agile methodologies and UML had much better outcomes and are currently being researched.

5.2 Implication and Findings for RQ2: Review Analysis of Industrial Practices

Research question 2 focuses on industrial practices for requirement traceability. In this SLR, we have analysed the tools, techniques, models, and methods for RT in socio-technical systems with popular industrial approaches. The papers that are reviewed for the industrial approaches mainly used empirical and agile methods and covered 70% of the practices that are currently being used in the industry (Sena Marques, Siegert, and Brisolara, 2014; Dassisti *et al.*, 2019; Bougdira, Akharraz and Ahaitouf, 2020; Fucci, Alégroth and Axelsson, 2022). The studies include the review practices and case studies designed for traceability in RE. The 20 relevant studies to support the industrial context have been identified. The literature and industrial aspect of approaches has been discussed in RQ3.

5.2.1 Analysis of Tools, Techniques and Models

Traceability is crucial in preserving consistency in the system since it may be utilised to infer a logical relationship between diverse artefacts at any stage of the development life cycle. The industrial approaches are dynamic, and various stakeholders are involved with heterogeneous requirements. Section 4.2 has discussed the techniques, models, and tools that are being used by the industry to create an interoperable approach. The studies have shown various automated tools used in RT integration. The popular agile methods with UML modeling and trace linking technique have shown better results in diverse environments, as discussed by (Santos *et al.*, 2018; Khlif, Kchaou and Bouassida, 2022). The question has been covered in approximately 20 papers reviewed in this SLR.

5.3 Findings for RQ3: Analysis of Literary and Industrial Practices

Traceability approaches usually take into account the context of the development effort and procedures, which acts as their distinguishing feature. Real-time and efficient strategies are needed to tackle the multi-traceability industry dilemma. The methods of industries are highly dependent on empirical and agile approaches with automated tools for traceability, as discussed in 10% of the articles reviewed from the year 2017 to 2022. The approach is widely addressed in articles that are published between 2018 and 2020. The majority of literary methods are theoretical and involve tradeoffs when applied in the industry. The diverse demands of industry necessitate a variety of techniques. The methods of literature applied in the industry need skills and technical expertise, such as NLP and ML-based models. The literature has covered feasible approaches such as IR-based approaches as covered in (Zhang, Wan and Jin, 2016) and (I. Rubasinghe, Meedeniya and Perera, 2018b) that are cost-efficient but have a time constraint.

5.3.1 Industrial and Literary Practices a Contradictory Analysis

Based on the research, it has been determined that requirement traceability is often utilised in industrial activities to guarantee the quality and efficacy of a system or product. Requirement traceability is used in software development to ensure that the program complies with client requirements and that any modifications to the requirements are correctly recorded and communicated. As discussed in studies (Bauer and Herder, 2009; Peraldi-Frati and Albinet, 2010; Bougdira, Akharraz and Ahaitouf, 2020), the industrial case studies of socio-technical systems involve rigorous requirement change and retracing. The blockchain model is implemented for requirements' security and collaboration for multiple stakeholders but also faces scalability issues, as we have seen in (Lenarduzzi *et al.*, 2018; Demi, Sanchez-Gordon and Colomo-Palacios, 2021; Ho *et al.*, 2021; Demi *et al.*, 2022; Farooq, Ahmed and Emran, 2022). The literature covered the studies for this question in section 4 and section 6 in detail.

6 Discussions

The section explains the implications based on the findings of the study, as discussed in the results section (see Section 5).

6.1 Existing Practices in Literature for Traceability in Requirements Engineering of Socio-technical Systems

In section 4, we have analysed the approaches in the literature adopted for RT in STS. An in-depth discussion of the approaches, methods, and frameworks with respect to their contributions and tradeoffs will also be done here. The hybrid techniques are quite popular in the literature, and various authors have adopted them to improve the traceability of various artifacts. Because socio-technical systems frequently feature complex connections between technical and social components, a single traceability approach would not be adequate to capture all of these linkages, making hybrid approaches effective in these systems. Authors (Eyal Salman, Seriai and Dony, 2023) proposed traceability in socio-technical systems using a hybrid technique that combines Latent Semantic Indexing (LSI) and Scenario-Based Probabilistic Ranking (SPR). The method utilises SPR to rank the likelihood of distinct needs being connected based on how similar their characteristics are after using LSI to extract features from the text of the requirements and detect semantic relationships between different requirements.

The authors (Wang, Li and Yang, 2019) have proposed S2trace based on the idea of using sequential semantics, or the meaning of words and phrases in context, to identify relationships between requirements and code. The authors (Saini *et al.*, 2021) proposed techniques by combining natural language processing with graph-based methods. NLP methods are used to extract data from text-based requirements, while graph-based methods are used to illustrate the connections between various requirements. The impact combining these two methods can offer a more comprehensive understanding of how requirements relate to one another in a socio-technical system

The authors (Putro and Wibowo, 2018; Hidayati and Rochimah, 2020) have combined use cases and requirement traceability matrices. It displays the connections between several categories of requirements, including quality, non-functional, and functional requirements. Qualitative techniques are also quite popular in literature, such as use cases, prototyping, focus groups, and version control(Mukherjee *et al.*, 2011).

Use cases are used to show the connections between various criteria by illustrating how a system interacts with its users. A more comprehensive understanding of the connections between the needs in a socio-technical system may result from integrating these two techniques. Use cases are broad explanations of how a technology will be used. Connecting requirements to the specific activities a user will do with the system aid in establishing traceability(White, Krinke and Tan, 2020).

Authors (Paiva, Maciel and da Silva, 2020) proposed the technique known as RSL, which incorporates prototype transformation procedures, notably a transformation from requirements into test cases (specified in RSL) and another from test cases into test scripts to track the tests' performance following the requirement. Another qualitative strategy for establishing traceability is prototyping, which involves making a physical or digital model of the system and its needs. This enables interested parties to understand how the criteria will be put into practice and how they interact with one another.

In the model-based design of CPSs, the authors (Mengist, Buffoni and Pop, 2021) proposed a method with a use case to validate it and a prototype for automatically establishing and maintaining the relevant traceability links between diverse artefacts ranging from requirement models through design models to prototype and validation results throughout the product life cycle. A neural blockchain prototype for requirements traceability (BC4RT) has been created. To put the prototype into practice, NDL ArcaNet, a neural blockchain platform, is employed (Demi *et al.*, 2022). The suggested prototype offers a comprehensive and reliable view of software artefacts, changes to requirements, and trace linkages. The increased cooperation, trust, and communication among stakeholders brought about by the increased visibility may improve the efficiency and quality of software development.

6.1.1 Contributions of Existing Practices to Solving Software Engineering Problems

Basic traceability techniques such as forward traceability, reverse traceability, and lateral and temporal traceability are widely adapted to solve software engineering issues that occur in traceability. Agile development methods have contributed a lot to enhancing focus on adaptability, teamwork, and quick prototyping while creating software and other products. Requirement traceability, or the capacity to track the connections between requirements and the rest of the development process, is a crucial component of agile development. This is crucial in socio-technical systems because they entail complex relationships between people, processes, and technology. Although blockchain is yet to be explored thoroughly, the research can be groundbreaking for the RE as blockchain systems demand specific SRE techniques since standard approaches often have problems with RT, client confidentiality, and Negotiation of requirement (RN), which calls for further refinement.

Additionally, the SRE architecture may use blockchain technology as an infrastructure to provide transparency, security, and dependability (Farooq, Ahmed and Emran, 2022). Table 5 in section 4 shows the techniques with respect to the solutions these techniques offer, and Table 6 shows the techniques' effectiveness in requirement traceability. There are several NLP (Natural Language Processing) techniques that can be used to support requirement traceability in socio-technical systems. Named entity recognition (NER) (Zhang *et al.*, 2021), which includes recognising and naming certain entities, such as persons, companies, and places in a written document is a popular NLP approach for requirement traceability. Finding stakeholders and figuring out the size of the system's requirements can both benefit from this. Text categorization approaches may be used to group requirements and design components according to their content and purpose. This may aid in the organisation and prioritisation of requirements, as well as the identification of any possible gaps or inconsistencies in the system design. The techniques such as ANN and LSI (Wang *et al.*, 2018) impact the development process and requirement tracing. ANN might be used to categorise requirements according to their content and purpose and to predict how changes to one requirement affect other requirements or design components. LSI is used to recognise significant concepts and phrases in a text and categorise them according to their significance.

This can be beneficial for organising and categorising requirements and design aspects and spotting any gaps or inconsistencies in system design. The reviewed papers show that the literature has covered about 50% of traceability's qualitative and quantitative techniques. The papers published in the last decade have extensively focused on IR-based methods and UML models for traceability in RE.

6.1.2 Benefits and Drawbacks of Existing Practices

The review shows, as mentioned in Table 8, the benefits of the discussed techniques. The empirical techniques have been widely adopted in the last decade, as explained in Section 4. Empirical approaches can provide various advantages in the context of RT as it provides Increased accuracy. These techniques are accurate and ensure that the final system fits the demand. It provides predictability and improved usability. By gathering data on how users interact with a system, empirical approaches can assist in identifying and addressing any usability concerns, resulting in a more user-friendly system. By evaluating data from previous systems, empirical approaches can assist in finding patterns and trends that can be utilised to create more accurate forecasts about future system performance and behaviour.

While empirical approaches have several potential drawbacks to be aware of, such as time and cost, collecting and interpreting data from actual systems or user experiences may be time-consuming and costly, especially if user studies or field trials are involved. Limitation of scope is another drawback to this approach, as it often requires gathering data from a small sample of people or systems that may or may not be representative of the total population (Charalampidou *et al.*, 2021a). As a result, empirical research findings may not be generalizable to other people or systems. Ethical factors such as getting information permission and preserving participants' privacy and confidentiality must typically be taken into account while using such approaches. The ML and NLP techniques are quite popular in the literature, and the industry has shown a potential shift towards these approaches, but the techniques offer their tradeoffs. NLP and ML are dependent on data quality; the findings of the analysis may be inaccurate if the data is of low quality or does not adequately reflect real-world circumstances(Zhao *et al.*, 2021).

Due to their restricted interpretability, some NLP and ML algorithms, such as deep learning models, might be challenging to comprehend how they came to their findings. This might make it difficult to explain or defend an analysis's conclusions and uncover and eliminate any errors or biases.

Because the algorithms lack flexibility, they may be unable to adapt to changing needs or conditions without substantial extra development work. Additionally, such deployment requires specific expertise and skills that may not be easily available inside an organisation. Blockchain is an area under extensive study but has scalability issues. The literature related to the blockchain is still in its infancy.

6.2 Industrial Practices for Requirements Traceability of Socio-technical Systems

The industry practices that are adapted for RT actively and passively are discussed below with reference to the literature review.

6.2.1 Models and Methods

The models used in the industry are discussed in table 9. The models for traceability that are adopted in the industry, such as Meta Models and UML models, are quite popular. Relationships between various artefacts and aspects, such as requirements and design elements, or the influence of maintenance actions on the entire system, are included in the meta-model. The studies (Peraldi-Frati and Albinet, 2010; Dassisti *et al.*, 2019; Jadoon, Shafi, and Jan, 2019) have adapted the meta-model in integration with the techniques for the purpose. Other studies from industries have adapted UML models for RT (Sena Marques, Siegert, and Brisolara, 2014; Min, 2016; Khlif, Kchaou, and Bouassida, 2022). The UML modeling methods have influenced the industry in the past decade by integrating the ontology-based approaches and methods (Khlif, Kchaou and Bouassida, 2022).

The methods and their industrial usage is shown in table 10. The IR-based methods implemented in (Capobianco *et al.*, 2013; Zhang, Wan and Jin, 2016; B. Wang *et al.*, 2018; W. Wang *et al.*, 2018) show a search capability that allows users to search for certain requirements or other artefacts based on keywords or other information is how IR approaches are utilised for traceability. This can assist stakeholders in rapidly locating key requirements and comprehending how they connect to other sections of the system. The other popular hybrid and integrated industrial methods are discussed in (Wang, Li and Yang, 2019; Gul *et al.*, 2021; Islam, Manning and Cullen, 2021; Khlif, Kchaou and Bouassida, 2022) with the diverse industrial requirements of hybrid techniques often performs better. Agile approaches are built on iterative development, adaptability, and collaborative concepts and favour quick delivery of functioning software above comprehensive, advanced design. The agile models for traceability through epochs and lightweight trace method are discussed in (Krüger *et al.*, 2019), and integrated agile models with UML and NLP are discussed in (Santos *et al.*, 2018; Khlif, Kchaou and Bouassida, 2022).

6.2.2 Tools

Tools with respect to their industrial usage, as shown in Table 11, explain the current and past trends and industrial shift towards more diverse and dynamic approaches. A complete UML modeling tool for creating and managing UML diagrams and models, as well as other forms of visual models like Business Process Modeling Notation (BPMN) and ArchiMate, are used (Penicina, 2013). Visual paradigm, use case diagrams discussed in (Khlif, Kchaou and Bouassida, 2022) are used to generate and manage requirements traceability by developing visual models that depict the links between various requirements and the system as a whole. The NLP tools and their usage analysis with the RE perspective are done by (Zhao *et al.*, 2021). The Industrial tools that are used for traceability as discussed in (Mengist, Buffoni and Pop, 2021). The papers (Nair, De La Vara and Sen, 2013; Tufail *et al.*, 2018; Zhao *et al.*, 2021; Batot, Gérard and Cabot, 2022) used XML and UML for modeling and RT. The industrial tools that support RT, such as DOORS ("IBM products," n.d.) and IBM change, are used (Abo, 2010). Although JIRA provides efficient RT, the plugins require cost; therefore, they are unsuitable for low-budgeted projects (Akman *et al.*, 2016). The SAT analyzer tool for NLP is used in (I. D. Rubasinghe, Meedeniya and Perera, 2018; Rubasinghe, Meedeniya and Perera, 2021; Khlif, Kchaou and Bouassida, 2022) as NLP can be used to track source codes and diagrams created using the UML to create the structured format of the document. Literature has adapted various approaches using this hybrid technique and tool support.

RETRO.Net is integrated to track the artefacts in text and XML format for effective RT (Hayes, Dekhtyar and Payne, 2018). RTM (Torkar *et al.*, 2012) is used in integrating ontology approaches in 13 papers discussed in the literature. The RTM is widely adapted in the industry because of its flexibility as it tracks the progress of each requirement as well as ensures the coverage of all the requirements. The tool caliberRM (Martin, n.d.) is a highly functional industrial tool for RT as it is suitable for complex socio-technical models. It tracks the complex artifacts and links as well as modification proposals. It provides a variety of ways to keep track of all needs, combining these features to assess the effects of modification proposals. Empirical tools such as SVN are also used in the literature (Ali *et al.*, 2015; Akman *et al.*, 2016).

Figure 7 shows the tool used in the industry. The RTM has been observed to be the most used tool in the industry and literature in integrating NLP models and approaches. The IBM tools are now in the trend towards requirement traceability and management and are yet to be fully explored by literature and industry.

6.2.3 Techniques

As shown in Table 12, the usage of techniques in industry reflects that the standard graphical notation to model the systems by UML has been adopted by the industry in the past decade. The technique involves using case diagrams to describe the interactions, sequence, and class diagram to model the structure and relationships of the artefacts. The integration of the UML with various empirical approaches, as discussed by the authors (Penicina, 2013; Charalampidou *et al.*, 2021b; Mengist, Buffoni and Pop, 2021), is explored extensively in the industry. The other technique that is extensively used in the industry is IR-based RT. These techniques involve a keyword search that enables users to look for specifications or other artefacts using particular terms or phrases and a full-text search that locates the artefacts by searching the full-text phrases (Ali *et al.*, 2015). Meta data that provides the trace of an artefact by searching its status and link analysis are discussed in the literature and industry in past decades (Wang *et al.*, 2018). The techniques that are still being used extensively in the industry are Agile techniques and approaches. Agile approaches are built on iterative development, adaptability, and collaborative concepts, and they favour quick delivery of functioning software above comprehensive, advanced design. Therefore the highly impactful models are Agile driven, as discussed in the literature with respect the industrial practices. The Agile models and techniques are being used in integration with the new approaches, such as blockchain-based techniques for RT and security of the artefacts (Lenarduzzi *et al.*, 2018; Murtazina and Avdeenko, 2018, 2019; Santos *et al.*, 2018; Hidayati and Rochimah, 2020; Demi, Sanchez-Gordon and Colomo-Palacios, 2021). The techniques have shown impactful results in the literature and industrial practices and are still being explored further.

6.2.4 Approaches

The approaches that are quite popular in the industry for requirements traceability are discussed in section 4. The Agile hybrid approach dominates the industry in this area too. The

Agile based traceability approaches are integrated with NFR, Ontology-based approaches (Murtazina and Avdeenko, 2019), UML-based approaches, blockchain (Lenarduzzi *et al.*, 2018), and IR-based approaches as discussed in the papers (Arbain *et al.*, 2020; Maro *et al.*, 2022). The model-based approach entails developing and maintaining visual representations of the system, such as UML diagrams or flowcharts. These models may be used to depict the relationships between various system components and how they interact with one another. These approaches are noted in multiple papers and with various data-driven approaches as well, that includes NLP-based approaches that integrate with LSI and SVN (Akman *et al.*, 2016) and other empirical approaches. Extensive literature has been proposed using Agile-based approaches and models.

6.3 Ascertaining Overlap in Existing Literature and Industrial Practices

We discussed this in two parts by taking into account the findings from the research. The practitioners use traceability in the industry to track the changes in requirement analysis and the impact of that change by tracking the links between artefacts. Tracing links is an approach that is popular in the industry as industrial approaches widely deal with real-time phenomenon with multiple stakeholders involved (Wohlrab *et al.*, 2020). Overcoming the challenge of multi traceability industry requires real-time and effective approaches. The approaches in the literature are mostly theoretical and come with tradeoffs. With the diverse requirements of the industry, the approaches also need to be diverse. The context of the development effort and processes is always incorporated into traceability techniques and serves as their defining characteristic. Such as integrating agile traceability methods with different empirical and qualitative methodologies. The empirical approaches involve case studies, surveys, and field studies (Ali *et al.*, 2015; Zhang, Wan, and Jin, 2016; Mäder, Olivetto and Marcus, 2017; Singh and Kumar, 2020; Charalampidou *et al.*, 2021a). These approaches can overcome the collaborative challenges in traceability.

Also, traceability is driven by the nature of the system and the regulations related to it. Ontology-based and cloud-based traceability methods are employed in the industry to handle this diversity. It allows users to exchange and access remotely traceable data. The authors (Bougdira, Akharraz and Ahaitouf, 2020) use the ontology-based technique to integrate heterogeneous data. The study (Adithya and Deepak, 2021) creates a traceable cloud-based platform using semantic and ML approaches.

The literature also focuses on linking and tagging with RTM and other modeling and data-driven approaches. The literature has also adopted Agile and IR-based techniques massively, which are quite popular in the industry as well(Ali *et al.*, 2015). As agile models are flexible, they provide collaborative integration with continuous testing, making them suitable for dynamic industrial approaches for RT in STSs. In contrast, IR-based methods have been used in past decades, and new automated traceability tools have changed the dynamics for the better. On the contrary, the literature has devised various NLP (e.g., NLP4RE) and ML approaches (Zhao *et al.*, 2021). Such approaches have yet to handle big data analysis in RT. Blockchain is the new trending technology, and various hybrid techniques are proposed for RT using blockchain(Lenarduzzi *et al.*, 2018; Demi, 2020a; Ho *et al.*, 2021; Li and Kassem, 2021). The overlapping occurs for various reasons, but the industrial approaches are primarily dynamic compared to the literary approaches. The reasons for these discrepancies are explored in the subsection below.

6.3.1 The Root Causes of Discrepancy:

While there may be some similarities between the traceability techniques used in literature and those in industry, there also may be significant differences. These factors might be some of the fundamental reasons for these discrepancies:

Different objectives and aims: Organizations in the industry may have different priorities and goals than academic researchers, which might affect the traceability techniques adopted. This means different organisations may face various challenges and have other resources than academic researchers, which may affect the methods of traceability that are utilized.

Diverse phases of development: While the traceability techniques used in literature may represent more unique or experimental methods, those employed in industry may reflect the status of advancement in the subject at this time.

Different acceptance levels: While certain traceability techniques employed in literature may not yet be extensively used in industry, some may have more widespread adoption than others.

Different degrees of expertise: Traceability approaches may be more specialised among academic researchers than among practitioners in industry, which might result in variations in the methods employed.

The techniques and a few overlapping trends are discussed below:

6.3.1.1 NLP Approaches and Industrial Interoperability:

Finding defects, developing software artefacts, and tracking linkages between models are all purposes of NLP research coupled with agile approaches like user stories. Nonetheless, system analysts can apply NLP to manage the traces. It might be challenging since rigorous assessment procedures and NLP approach experimentation are required to give high-quality results. Understanding the context of a phrase is tough, as it is with other NLP techniques. The dependency on the dataset and semantics is sometimes challenging while processing dynamic artefacts and requirements(Raharjana, Siahaan and Fatichah, 2021).

6.3.1.2 Blockchain-based Methods and Their Discrepancy With Respect to Industry:

As stated in the preceding sections, blockchain has been adopted in the literature and the industry for RE techniques, although it has scalability concerns. Because blockchain consensus is computationally intensive, it has yet to be researched further for usage in real-time systems(Fournier and Petrillo, 2020).

6.3.1.3 Machine Learning-based Approaches and Interoperability:

Notably, RT is a highly difficult problem from the information retrieval and text mining perspectives. The ML models are data-driven implementation timeframes, which might vary based on what is implemented. Integration issues, a lack of understanding of cutting-edge technology, and usability and compatibility with other systems and platforms make ML difficult (Iqbal, Elahidoost and Lúcio, 2018). Another issue that must be considered for industrial practices is the expertise and cost related to such approaches.

6.3.1.4 Industrial Requirements for Traceability in Socio-technical Systems:

The preferences of the industry in terms of methods and approaches can be summarised below:

Model-based approach: This method entails developing and maintaining visual representations of the system, such as UML diagrams or flowcharts. These models may be used to depict the relationships between various system components and how they interact with one another.

Data-driven method: This strategy entails gathering and analysing data from the system, such as log files or performance measurements, in order to understand the behaviour and interactions of the system's various components.

Collaborative approach: This technique involves incorporating stakeholders in the creation and execution of the system, such as users and developers. This contributes to ensuring that the system serves all stakeholders' demands while encouraging openness and traceability throughout the development process.

In general, it is crucial for businesses to thoroughly assess the requirements of their projects and select traceability techniques that are compatible with their requirements and available resources.

6.4 Quality Score of Papers

Table 14 comprehensively details the expected quality ratings for each selected item. Twenty (20) of the chosen articles are worth three or more points. The papers with industrial approaches and comparisons have been given points above 4 and 5 points, including 4 review articles. This shows that the chosen articles' quality is commendable, the score of 3.5 and 4 is suggesting that we were able to address at least two of the three RQs in almost 30 articles out of 46. The answer to Q1 can be found in every article, and Q3 can be found in the majority of articles. Research question 2 and RQ3 are addressed in more than half of the articles which are given the score of 1 and are partially addressed in some which are given score of 0.5. Thus, to sum up, RQ1 and RQ2 are covered in most of the studies with a score of 3. The overlapping techniques were covered in 10 of these articles and given a score of 2 for RQ3.

6.4.1 SLR approach

An iterative technique is employed for literature filtering. The first step is to answer RQ1. According to the methodology, the articles that include literary practices are categorised and screened. Current trends are noted, and SLRs from the previous three years are also assessed. The spiral technique has helped address RQs 2 and 3 since it highlights the ongoing interaction between literary and commercial practices.

7 Research Findings

1. This study successfully finds papers that offer any form of empirical, qualitative, and quantitative data pertaining to traceability and comprehending their qualities.
2. The study supports the need for more research on the efficacy of various traceability frameworks and approaches in socio-technical systems, including qualitative and quantitative effectiveness measurements.
3. This SLR analyses the trade-off and the effects of thriving traceability tools and methodologies in sociotechnical systems.
4. The majority of the suggested methods are not adequately assessed for a practitioner to draw judgments only on the study. The literary and industrial practices are investigated and analysed in this SLR in simultaneous and comparative ways.
5. Numerous studies have merely taken into account one aspect of the issue, such as domain-specific techniques. The approaches that are supported by industrial case studies and literary techniques have been examined in this study.
6. We presented a systematic review of the literature to classify the traceability techniques, highlight their benefits and drawbacks, and classify the methodologies and frameworks utilised. We also identified the software traceability tools and their usage with hybrid techniques to demonstrate the traceability approaches.
7. The blockchain and its integration with hybrid techniques have been explored in this SLR.
8. The aspects presented in the study indicate several ways in which more work on requirement traceability approaches, models, frameworks, and tools in socio-technical systems is required.

Our results may impact how requirement traceability research is conducted in the future, enabling researchers and designers to think about the issue more clearly and create models and frameworks. For evolving needs, requirement traceability is crucial. As needs change, maintaining traceability becomes more difficult and expensive in the interim.

8 Conclusion and Future work

In this thesis, we first carried out a thorough literature review to fully comprehend requirement traceability and provide answers to the research questions formulated.

Numerous papers on literary techniques, industrial practices, trade-offs, and contributions to requirement traceability may be found while searching for the answers to the RQs. Using integrated tools and industrial practices, studies of the key approaches in current RT operations were conducted, and existing RT techniques were categorised according to their distinct domains and specifications. The body of literature and its impact on RT. The possibilities and drawbacks of different traceability approaches have been thoroughly studied. Various methodologies have been evaluated, and research conclusions have been developed.

Various strategies from various domains, including information retrieval, machine learning, swarm techniques, and empirical, agile, and hybrid methodologies, have been demonstrated to be successful in tracing issues. Using these principles in the RT area has increased efficiency while decreasing human effort, according to experience. As a result, introducing new ideas in new domains, such as blockchain in evolutionary computation, or expanding current disciplines, such as deep learning in machine learning, is both feasible and viable.

The organised and interrelated components of the study may be used as a starting point by field researchers to investigate a range of research goals from social or technological perspectives. The systematic evaluation of traceability management in industrial projects is a pertinent area for future study.

References

Garousi, V., Felderer, M. and Mäntylä, M.V., 2016, June. The need for multivocal literature reviews in software engineering: complementing systematic literature reviews with grey literature. In Proceedings of the 20th international conference on evaluation and assessment in software engineering (pp. 1-6).Ghozali, R.P., Saputra, H., Nuriawan, M.A., Utama, D.N. and Nugroho, A., 2019. Systematic literature review on decision-making of requirement engineering from agile software development. Procedia Computer Science, 157, pp.274-281.

Gotel, O. and Finkelstein, A., 1997, January. Extended requirements traceability: Results of an industrial case study. In Proceedings of ISRE'97: 3rd IEEE International Symposium on Requirements Engineering (pp. 169-178). IEEE.

Kitchenham, B. and Brereton, P., 2013. A systematic review of systematic review process research in software engineering. Information and software technology, 55(12), pp.2049-2075. Krause, J., Kaufmann, A., Riehle, D. and Jung, M., 2022, August. The Benefits of Pre-Requirements Specification Traceability. In 2022 IEEE 30th International Requirements Engineering Conference (RE) (pp. 166-177). IEEE.

Lin, F. and Chen, H., 2019. Comparative Study of Requirements Traceability in Facing Requirements Change: Systematic Literature Study and Survey. Maltji, M., Von Solms, S. and Marnewick, A., 2019. Socio-technical systems cybersecurity framework. Information & Computer Security.

Ncube, C. and Lim, S.L., 2018, August. On systems of systems engineering: A requirements engineering perspective and research agenda. In 2018 IEEE 26th International Requirements Engineering Conference (RE) (pp. 112-123). IEEE.

Parizi, R.M., Lee, S.P. and Dabbagh, M., 2014. Achievements and challenges in state-of-the-art software traceability between test and code artifacts. IEEE Transactions on Reliability, 63(4), pp.913-926.

Abo, R. (2010) 'Introduction to a Requirements Engineering Framework for Aeronautics', *Journal of*

Software Engineering and Applications, 03(09), pp. 894–900. Available at: https://doi.org/10.4236/jsea.2010.39105.

Adithya, V. and Deepak, G. (2021) 'OntoReq: An Ontology Focused Collective Knowledge Approach for Requirement Traceability Modelling', in A.M.A. Musleh Al-Sartawi, A. Razzaque, and M.M. Kamal (eds) *Artificial Intelligence Systems and the Internet of Things in the Digital Era*. Cham: Springer International Publishing, pp. 358–370.

Adjepon-Yamoah, D.E. (2016) 'Towards Dependable Change Management and Traceability for Global Software Development'. Available at: http://arxiv.org/abs/1608.05981.

Ågren, S.M. *et al.* (2019) 'The impact of requirements on systems development speed: a multiple-case study in automotive', *Requirements Engineering*, 24(3), pp. 315–340. Available at: https://doi.org/10.1007/s00766-019-00319-8.

Akman, S. *et al.* (2016) 'Experience report: implementing requirement traceability throughout the software development life cycle', *Journal of Software: Evolution and Process*, 28(11), pp. 950–954. Available at: https://doi.org/10.1002/smr.1824.

Al-Dhaqm, A. *et al.* (2021) 'Digital Forensics Subdomains: The State of the Art and Future Directions', *IEEE Access*, 9, pp. 152476–152502. Available at: https://doi.org/10.1109/ACCESS.2021.3124262.

Ali, N. *et al.* (2015) 'An empirical study on the importance of source code entities for requirements traceability', *Empirical Software Engineering*, 20(2), pp. 442–478. Available at: https://doi.org/10.1007/s10664-014-9315-y.

Alkhammash, E. (2020) 'Formal modelling of OWL ontologies-based requirements for the development of safe and secure smart city systems', *Soft Computing*, 24(15), pp. 11095–11108. Available at: https://doi.org/10.1007/s00500-020-04688-z.

Arbain, A.F. *et al.* (2020) 'Case study on non-functional requirement change impact traceability for Agile software development', *International Journal on Advanced Science, Engineering and Information Technology*, 10(1), pp. 34–40. Available at: https://doi.org/10.18517/ijaseit.10.1.10176.

Arunthavanathan, A. *et al.* (2016) 'Support for traceability management of software artefacts using Natural Language Processing', in *2016 Moratuwa Engineering Research Conference (MERCon)*, pp. 18–23. Available at: https://doi.org/10.1109/MERCon.2016.7480109.

Bandara, M. *et al.* (2019) 'From Requirements to Data Analytics Process: An Ontology-Based Approach', in F. Daniel, Q.Z. Sheng, and H. Motahari (eds) *Business Process Management Workshops*. Cham: Springer International Publishing, pp. 543–552.

Batot, E.R., Gérard, S. and Cabot, J. (2022) *A Survey-driven Feature Model for Software Traceability Approaches*, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer International Publishing. Available at: https://doi.org/10.1007/978-3-030-99429-7_2.

Bauer, J.M. and Herder, P.M. (2009) 'Designing Socio-Technical Systems', in A. Meijers (ed.) *Philosophy of Technology and Engineering Sciences*. Amsterdam: North-Holland (Handbook of the Philosophy of Science), pp. 601–630. Available at: https://doi.org/https://doi.org/10.1016/B978-0-444-51667-1.50026-4.

Bougdira, A., Akharraz, I. and Ahaitouf, A. (2020) 'A traceability proposal for industry 4.0', *Journal of Ambient Intelligence and Humanized Computing*, 11(8), pp. 3355–3369. Available at: https://doi.org/10.1007/s12652-019-01532-7.

Capobianco, G. *et al.* (2013) 'Improving IR-based traceability recovery via noun-based indexing of software artifacts', *Journal of Software: Evolution and Process*, 25(7), pp. 743–762. Available at: https://doi.org/https://doi.org/10.1002/smr.1564.

Carvalho, T.P. *et al.* (2019) 'A systematic literature review of machine learning methods applied to predictive maintenance', *Computers & Industrial Engineering*, 137, p. 106024. Available at: https://doi.org/https://doi.org/10.1016/j.cie.2019.106024.

Cepulis, D. and Niu, N. (2018) 'Creating socio-technical patches for information foraging: A requirements traceability case study', *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC*, 2018-Octob, pp. 17–21. Available at: https://doi.org/10.1109/VLHCC.2018.8506526.

Charalampidou, S. *et al.* (2021a) 'Empirical studies on software traceability: A mapping study', *Journal of Software: Evolution and Process*, 33(2), p. e2294. Available at: https://doi.org/https://doi.org/10.1002/smr.2294.

Charalampidou, S. *et al.* (2021b) 'Empirical studies on software traceability: A mapping study', *Journal of Software: Evolution and Process*, 33(2), pp. 1–28. Available at: https://doi.org/10.1002/smr.2294.

Chiarini, A. (2020) 'Industry 4.0, quality management and TQM world. A systematic literature review and a proposed agenda for further research', *TQM Journal*, 32(4), pp. 603–616. Available at: https://doi.org/10.1108/TQM-04-2020-0082.

Corallo, A. *et al.* (2020) 'A systematic literature review to explore traceability and lifecycle relationship', *International Journal of Production Research*, 58(15), pp. 4789–4807. Available at:

https://doi.org/10.1080/00207543.2020.1771455.

Dalpiaz, F. and Niu, N. (2020) ‘Requirements Engineering in the Days of Artificial Intelligence’, *IEEE Software*, 37(4), pp. 7–10. Available at: https://doi.org/10.1109/MS.2020.2986047.

Dassisti, M. *et al.* (2019) ‘An approach to support Industry 4.0 adoption in SMEs using a core-metamodel’, *Annual Reviews in Control*, 47, pp. 266–274. Available at: https://doi.org/https://doi.org/10.1016/j.arcontrol.2018.11.001.

Demi, S. (2020a) ‘Blockchain-oriented Requirements Engineering: A Framework’, *Proceedings of the IEEE International Conference on Requirements Engineering*, 2020-Augus, pp. 428–433. Available at: https://doi.org/10.1109/RE48521.2020.00063.

Demi, S. (2020b) ‘Blockchain-oriented Requirements Engineering: A Framework’, in *2020 IEEE 28th International Requirements Engineering Conference (RE)*, pp. 428–433. Available at: https://doi.org/10.1109/RE48521.2020.00063.

Demi, S. *et al.* (2022) ‘A Neural Blockchain for Requirements Traceability: BC4RT Prototype’, in M. Yilmaz et al. (eds) *Systems, Software and Services Process Improvement*. Cham: Springer International Publishing, pp. 45–59.

Demi, S., Sanchez-Gordon, M. and Colomo-Palacios, R. (2021) ‘What have we learnt from the challenges of (semi-) automated requirements traceability? A discussion on blockchain applicability’, *IET Software*, 15(6), pp. 391–411. Available at: https://doi.org/https://doi.org/10.1049/sfw2.12035.

Demi, S., Sánchez-Gordón, M. and Kristiansen, M. (no date) ‘Blockchain for requirements traceability: A qualitative approach’, *Journal of Software: Evolution and Process*, n/a(n/a), p. e2493. Available at: https://doi.org/https://doi.org/10.1002/smr.2493.

Dey, S. and Lee, S.W. (2017) ‘REASSURE: Requirements elicitation for adaptive socio-technical systems using repertory grid’, *Information and Software Technology*, 87, pp. 160–179. Available at: https://doi.org/10.1016/j.infsof.2017.03.004.

Esparteiro Garcia, J. and C. R. Paiva, A. (2016) ‘A Requirements-to-Implementation Mapping Tool for Requirements Traceability’, *Journal of Software*, 11(2), pp. 193–200. Available at: https://doi.org/10.17706/jsw.11.2.193-200.

Eyal Salman, H., Seriai, A.-D. and Dony, C. (2023) ‘Feature Location in Software Variants Toward Software Product Line Engineering’, in R.E. Lopez-Herrejon et al. (eds) *Handbook of Re-Engineering Software Intensive Systems into Software Product Lines*. Cham: Springer International Publishing, pp. 3–30. Available at: https://doi.org/10.1007/978-3-031-11686-5_1.

Falessi, D. *et al.* (2017) ‘Estimating the number of remaining links in traceability recovery’, *Empirical Software Engineering*, 22(3), pp. 996–1027. Available at: https://doi.org/10.1007/s10664-016-9460-6.

Farooq, M.S., Ahmed, M. and Emran, M. (2022) ‘A Survey on Blockchain Acquainted Software Requirements Engineering: Model, Opportunities, Challenges, and Future Directions’, *IEEE Access*, 10, pp. 48193–48228. Available at: https://doi.org/10.1109/ACCESS.2022.3171408.

Fournier, G. and Petrillo, F. (2020) ‘Architecting Blockchain Systems: A Systematic Literature Review’, *Proceedings - 2020 IEEE/ACM 42nd International Conference on Software Engineering Workshops, ICSEW 2020*, pp. 664–670. Available at: https://doi.org/10.1145/3387940.3392196.

Fucci, D., Alégroth, E. and Axelsson, T. (2022) ‘When traceability goes awry: An industrial experience report’, *Journal of Systems and Software*, 192, p. 111389. Available at: https://doi.org/https://doi.org/10.1016/j.jss.2022.111389.

Gotel, O.C.Z. and Finkelstein, A.C.W. (1994) ‘An Analysis of the Requirements Traceability Problem Imperial College of Science , Technology & Medicine Department of Computing , 180 Queen ’ s Gate’, *Proceedings of the First International Conference on Requirements Engineering*, pp. 94–101.

Gul, L. *et al.* (2021) ‘Integrated Traceability Approach for an Effective Impact Analysis’, *Mehran University Research Journal of Engineering and Technology*, 40(2), pp. 346–357. Available at: https://doi.org/10.22581/muet1982.2102.09.

Guo, J., Cheng, J. and Cleland-Huang, J. (2017) ‘Semantically Enhanced Software Traceability Using Deep Learning Techniques’, in *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, pp. 3–14. Available at: https://doi.org/10.1109/ICSE.2017.9.

Guo, J., Gibiec, M. and Cleland-Huang, J. (2017) ‘Tackling the term-mismatch problem in automated trace retrieval’, *Empirical Software Engineering*, 22(3), pp. 1103–1142. Available at: https://doi.org/10.1007/s10664-016-9479-8.

Hayes, J.H., Dekhtyar, A. and Payne, J. (2018) ‘The requirements tracing on target (RETRO).NET dataset’, *Proceedings - 2018 IEEE 26th International Requirements Engineering Conference, RE 2018*, pp. 424–427. Available at: https://doi.org/10.1109/RE.2018.00054.

Hidayati, N.N. and Rochimah, S. (2020) ‘Requirements Traceability for Detecting Defects in Agile Software Development’, in *2020 10th Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS)*, pp. 248–253. Available at: https://doi.org/10.1109/EECCIS49483.2020.9263420.

Ho, G.T.S. *et al.* (2021) 'A blockchain-based system to enhance aircraft parts traceability and trackability for inventory management', *Expert Systems with Applications*, 179, p. 115101. Available at: https://doi.org/https://doi.org/10.1016/j.eswa.2021.115101.

Iqbal, T., Elahidoost, P. and Lúcio, L. (2018) 'A Bird's Eye View on Requirements Engineering and Machine Learning', in *2018 25th Asia-Pacific Software Engineering Conference (APSEC)*, pp. 11–20. Available at: https://doi.org/10.1109/APSEC.2018.00015.

Islam, S., Manning, L. and Cullen, J.M. (2021) 'A Hybrid Traceability Technology Selection Approach for Sustainable Food Supply Chains', *Sustainability*, 13(16). Available at: https://doi.org/10.3390/su13169385.

Jadoon, G., Shafi, M. and Jan, S. (2019) 'A Model-Oriented Requirements Traceability Framework for Small and Medium Software Industries', in *2019 International Arab Conference on Information Technology (ACIT)*, pp. 91–96. Available at: https://doi.org/10.1109/ACIT47987.2019.8991116.

Jayatilleke, S. and Lai, R. (2018) 'A systematic review of requirements change management', *Information and Software Technology*, 93, pp. 163–185. Available at: https://doi.org/https://doi.org/10.1016/j.infsof.2017.09.004.

Kassab, M., Ormandjieva, O. and Daneva, M. (2008) 'A Traceability Metamodel for Change Management of Non-functional Requirements', in *2008 Sixth International Conference on Software Engineering Research, Management and Applications*, pp. 245–254. Available at: https://doi.org/10.1109/SERA.2008.37.

Khan, M.A. and Salah, K. (2018) 'IoT security: Review, blockchain solutions, and open challenges', *Future Generation Computer Systems*, 82, pp. 395–411. Available at: https://doi.org/10.1016/j.future.2017.11.022.

Khlif, W., Kchaou, D. and Bouassida, N. (2022) 'A Complete Traceability Methodology Between UML Diagrams and Source Code Based on Enriched Use Case Textual Description', *Informatica (Slovenia)*, 46(1), pp. 27–47. Available at: https://doi.org/10.31449/inf.v46i1.3306.

Kirikova, M. (2022) 'Continuous Requirements Engineering in Sociotechnical Systems: Challenges and Solutions', *CEUR Workshop Proceedings*, 3107.

Krüger, J. *et al.* (2019) 'Effects of Explicit Feature Traceability on Program Comprehension', in *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. New York, NY, USA: Association for Computing Machinery (ESEC/FSE 2019), pp. 338–349. Available at:

https://doi.org/10.1145/3338906.3338968.

Lenarduzzi, V. *et al.* (2018) 'Blockchain Applications for Agile Methodologies', in *Proceedings of the 19th International Conference on Agile Software Development: Companion*. New York, NY, USA: Association for Computing Machinery (XP '18). Available at: https://doi.org/10.1145/3234152.3234155.

Li, J. *et al.* (2021) 'Multilevel Traceability Links Establishments Between SOFL Formal Specifications and Java Codes Using Multi-dimensional Similarity Measures', in *2021 IEEE 21st International Conference on Software Quality, Reliability and Security (QRS)*, pp. 852–863. Available at: https://doi.org/10.1109/QRS54544.2021.00094.

Li, J. and Kassem, M. (2021) 'Applications of distributed ledger technology (DLT) and Blockchain-enabled smart contracts in construction', *Automation in Construction*, 132(September), p. 103955. Available at: https://doi.org/10.1016/j.autcon.2021.103955.

Li, W., Vaughn, R. and Saiedian, H. (2002) 'Pre-Requirements Traceability', in.

Lin, F. and Chen, H. (2019) 'Comparative Study of Requirements Traceability in Facing Requirements Change: Systematic Literature Study and Survey', (January). Available at: http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1305894&dswid=8014.

Mader, P., Gotel, O. and Philippow, I. (2009) 'Getting back to basics: Promoting the use of a traceability information model in practice', in *2009 ICSE Workshop on Traceability in Emerging Forms of Software Engineering*, pp. 21–25. Available at: https://doi.org/10.1109/TEFSE.2009.5069578.

Mäder, P., Olivetto, R. and Marcus, A. (2017) 'Empirical studies in software and systems traceability', *Empirical Software Engineering*, 22(3), pp. 963–966. Available at: https://doi.org/10.1007/s10664-017-9509-1.

Maro, S. *et al.* (2022) 'TracIMo: a traceability introduction methodology and its evaluation in an Agile development team', *Requirements Engineering*, 27(1), pp. 53–81. Available at: https://doi.org/10.1007/s00766-021-00361-5.

Mengist, A., Buffoni, L. and Pop, A. (2021) 'An integrated framework for traceability and impact analysis in requirements verification of cyber–physical systems', *Electronics (Switzerland)*, 10(8). Available at: https://doi.org/10.3390/electronics10080983.

Min, H.-S. (2016) 'Traceability Guideline for Software Requirements and UML Design', *International Journal of Software Engineering and Knowledge Engineering*, 26(01), pp. 87–113. Available at: https://doi.org/10.1142/S0218194016500054.

Montecchi, L. and Gallina, B. (2017) 'SafeConcert: A Metamodel for a Concerted Safety Modeling of Socio-Technical Systems', in M. Bozzano and Y. Papadopoulos (eds) *Model-Based Safety and Assessment*. Cham: Springer International Publishing, pp. 129–144.

Mosquera, D. *et al.* (2022) 'OntoTrace: A Tool for Supporting Trace Generation in Software Development by Using Ontology-Based Automatic Reasoning', in J. De Weerdt and A. Polyvyanyy (eds) *Intelligent Information Systems*. Cham: Springer International Publishing, pp. 73–81.

Mukherjee, P. *et al.* (2011) 'Traceability Link Evolution with Version Control.', in, pp. 151–161.

Murtazina, M.S. and Avdeenko, T. V. (2019) 'An ontology-based approach to support for requirements traceability in agile development', *Procedia Computer Science*, 150, pp. 628–635. Available at: https://doi.org/10.1016/j.procs.2019.02.044.

Murtazina, M.S. and Avdeenko, T. V (2018) 'Ontology-Based Approach to the Requirements Engineering in Agile Environment', in *2018 XIV International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE)*, pp. 496–501. Available at: https://doi.org/10.1109/APEIE.2018.8546144.

Mustafa, N. and Labiche, Y. (2017) 'The Need for Traceability in Heterogeneous Systems: A Systematic Literature Review', in *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, pp. 305–310. Available at: https://doi.org/10.1109/COMPSAC.2017.237.

Nair, S., De La Vara, J.L. and Sen, S. (2013) 'A review of traceability research at the requirements engineering conferencere@21', *2013 21st IEEE International Requirements Engineering Conference, RE 2013 - Proceedings*, (May 2020), pp. 222–229. Available at: https://doi.org/10.1109/RE.2013.6636722.

Niu, N. *et al.* (2018) 'Requirements Socio-Technical Graphs for Managing Practitioners' Traceability Questions', *IEEE Transactions on Computational Social Systems*, 5(4), pp. 1152–1162. Available at: https://doi.org/10.1109/TCSS.2018.2872059.

Paiva, A.C.R., Maciel, D. and da Silva, A.R. (2020) 'From Requirements to Automated Acceptance Tests with the RSL Language', in E. Damiani, G. Spanoudakis, and L.A. Maciaszek (eds) *Evaluation of Novel Approaches to Software Engineering*. Cham: Springer International Publishing, pp. 39–57.

Paul, J. *et al.* (2021) 'Scientific procedures and rationales for systematic literature reviews (SPAR-4-SLR)', *International Journal of Consumer Studies*, 45(4), pp. O1–O16. Available at: https://doi.org/https://doi.org/10.1111/ijcs.12695.

Penicina, L. (2013) 'Linking BPMN, ArchiMate, and BWW: Perfect match for complete and lawful business process models?', *CEUR Workshop Proceedings*, 1023(April), pp. 156–165.

Peraldi-Frati, M.-A. and Albinet, A. (2010) 'Requirement Traceability in Safety Critical Systems', in *Proceedings of the 1st Workshop on Critical Automotive Applications: Robustness & Safety*. New York, NY, USA: Association for Computing Machinery (CARS '10), pp. 11–14. Available at: https://doi.org/10.1145/1772643.1772647.

Pessoa, M.V.P. *et al.* (2022) 'Model-Based Digital Threads for Socio-Technical Systems', in G. Marques, A. González-Briones, and J.M. Molina López (eds) *Machine Learning for Smart Environments/Cities: An IoT Approach*. Cham: Springer International Publishing, pp. 27–52. Available at: https://doi.org/10.1007/978-3-030-97516-6_2.

Putro, H.P. and Wibowo, A.F. (2018) 'Software Verification and Validation on Object Oriented Software Development Using Traceability Matrix', in *2018 Third International Conference on Informatics and Computing (ICIC)*, pp. 1–5. Available at: https://doi.org/10.1109/IAC.2018.8780518.

Raharjana, I.K., Siahaan, D. and Fatichah, C. (2021) 'User Stories and Natural Language Processing: A Systematic Literature Review', *IEEE Access*, 9, pp. 53811–53826. Available at: https://doi.org/10.1109/ACCESS.2021.3070606.

Repinskaya, R.P. and Eremina, N.S. (1985) 'Effect of Filtration of Small-Scale Disturbances in the H//5//0//0 Field on the Integration of Equations of a Barotropic Nongeostrophic Model.', *Soviet meteorology and hydrology*, 9(1), pp. 31–35.

Rubasinghe, I., Meedeniya, D. and Perera, I. (2018a) 'Automated Inter-artefact Traceability Establishment for DevOps Practice', in *2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*, pp. 211–216. Available at: https://doi.org/10.1109/ICIS.2018.8466414.

Rubasinghe, I., Meedeniya, D. and Perera, I. (2018b) 'Traceability Management with Impact Analysis in DevOps based Software Development', in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1956–1962. Available at: https://doi.org/10.1109/ICACCI.2018.8554399.

Rubasinghe, I., Meedeniya, D. and Perera, I. (2021) 'SAT-Analyser Traceability Management Tool Support for DevOps', *Journal of Information Processing Systems*, 17, pp. 972–988. Available at: https://doi.org/10.3745/JIPS.04.0225.

Rubasinghe, I.D., Meedeniya, D.A. and Perera, I. (2018) 'Software Artefact Traceability Analyser: A Case-Study on POS System', in *Proceedings of the 6th International Conference on Communications and Broadband Networking*. New York, NY, USA: Association for Computing Machinery (ICCBN 2018), pp. 1–5. Available at: https://doi.org/10.1145/3193092.3193094.

Saini, R. *et al.* (2021) ‘Automated Traceability for Domain Modelling Decisions Empowered by Artificial Intelligence’, in *2021 IEEE 29th International Requirements Engineering Conference (RE)*, pp. 173–184. Available at: https://doi.org/10.1109/RE51729.2021.00023.

Santos, N. *et al.* (2018) ‘An Agile Modeling Oriented Process for Logical Architecture Design’, in J. Gulden et al. (eds) *Enterprise, Business-Process and Information Systems Modeling*. Cham: Springer International Publishing, pp. 260–275.

Saravanan, A. and Bama, S.S. (2019) ‘A Review on Cyber Security and the Fifth Generation Cyberattacks’, *Oriental journal of computer science and technology*, 12(2), pp. 50–56. Available at: https://doi.org/10.13005/ojcst12.02.04.

Sena Marques, M.R., Siegert, E. and Brisolara, L. (2014) ‘Integrating UML, MARTE and sysml to improve requirements specification and traceability in the embedded domain’, in *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, pp. 176–181. Available at: https://doi.org/10.1109/INDIN.2014.6945504.

Singh, T. and Kumar, M. (2020) ‘Empirical Validation of Requirements Traceability Metrics for Requirements Model of Data Warehouse using SVM’, in *2020 IEEE 17th India Council International Conference (INDICON)*, pp. 1–5. Available at: https://doi.org/10.1109/INDICON49873.2020.9342245.

Sultanov, H. and Hayes, J. (2010) ‘Application of Swarm Techniques to Requirements Engineering: Requirements Tracing’, in *Proceedings of the 2010 18th IEEE International Requirements Engineering Conference, RE2010*, pp. 211–220. Available at: https://doi.org/10.1109/RE.2010.33.

Sultanov, H. and Hayes, J.H. (2013) ‘Application of reinforcement learning to requirements engineering: requirements tracing’, in *2013 21st IEEE International Requirements Engineering Conference (RE)*, pp. 52–61. Available at: https://doi.org/10.1109/RE.2013.6636705.

Sundaram, S.K. *et al.* (2010) ‘Assessing traceability of software engineering artifacts’, *Requirements Engineering*, 15(3), pp. 313–335. Available at: https://doi.org/10.1007/s00766-009-0096-6.

‘Systematic Literature Review on the Extent of Conflict Resolution Re- search and Practice in Requirements Engineering of Socio-technical Systems : Where are we now ? Süstemaatiline kirjanduse ülevaade konfliktide tuvastamisel ja lahen- damisel kasutusel o’ (no date).

Tomlinson, A. (2017) ‘Introduction to the TPM’, *Smart Cards, Tokens, Security and Applications: Second Edition*, pp. 173–191. Available at: https://doi.org/10.1007/978-3-319-50500-8_7.

Torkar, R. *et al.* (2012) 'Requirements traceability: A systematic review and industry case study', *International Journal of Software Engineering and Knowledge Engineering*, 22(3), pp. 385–433. Available at: https://doi.org/10.1142/S021819401250009X.

Tufail, H. *et al.* (2018) 'A systematic review of requirement traceability techniques and tools', *2017 2nd International Conference on System Reliability and Safety, ICSRS 2017*, 2018-Janua(December), pp. 450–454. Available at: https://doi.org/10.1109/ICSRS.2017.8272863.

Victoria, D. and Caraballo, R. (2020) 'The Division of Computer Science and Engineering'.

Wang, B. *et al.* (2018) *Requirements traceability technologies and technology transfer decision support: A systematic review*, *Journal of Systems and Software*. Available at: https://doi.org/10.1016/j.jss.2018.09.001.

Wang, S., Li, T. and Yang, Z. (2019) 'Exploring Semantics of Software Artifacts to Improve Requirements Traceability Recovery: A Hybrid Approach', in *2019 26th Asia-Pacific Software Engineering Conference (APSEC)*, pp. 39–46. Available at: https://doi.org/10.1109/APSEC48747.2019.00015.

Wang, W. *et al.* (2018) 'Enhancing automated requirements traceability by resolving polysemy', *Proceedings - 2018 IEEE 26th International Requirements Engineering Conference, RE 2018*, pp. 40–51. Available at: https://doi.org/10.1109/RE.2018.00-53.

White, R., Krinke, J. and Tan, R. (2020) 'Establishing Multilevel Test-to-Code Traceability Links', in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*. New York, NY, USA: Association for Computing Machinery (ICSE '20), pp. 861–872. Available at: https://doi.org/10.1145/3377811.3380921.

Wohlrab, R. *et al.* (2020) 'Collaborative traceability management: a multiple case study from the perspectives of organization, process, and culture', *Requirements Engineering*, 25(1), pp. 21–45. Available at: https://doi.org/10.1007/s00766-018-0306-1.

Yoshino, K. and Matsuura, S. (2020) 'Requirements Traceability Management Support Tool for UML Models', *ACM International Conference Proceeding Series*, pp. 163–166. Available at: https://doi.org/10.1145/3384544.3384586.

Zhang, M. *et al.* (2021) 'Recovering Semantic Traceability between Requirements and Source Code Using Feature Representation Techniques', in *2021 IEEE 21st International Conference on Software Quality, Reliability and Security (QRS)*, pp. 873–882. Available at: https://doi.org/10.1109/QRS54544.2021.00096.

Zhang, Y., Wan, C. and Jin, B. (2016) 'An empirical study on recovering requirement-to-code links', *2016 IEEE/ACIS 17th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2016*, pp. 121–126. Available at: https://doi.org/10.1109/SNPD.2016.7515889.

Zhao, L. *et al.* (2021) ‘Natural Language Processing for Requirements Engineering: A Systematic Mapping Study’, *ACM Comput. Surv.*, 54(3). Available at: https://doi.org/10.1145/3444689.

Zogaan, W. *et al.* (2017) ‘Automated training-set creation for software architecture traceability problem’, *Empirical Software Engineering*, 22(3), pp. 1028–1062. Available at: https://doi.org/10.1007/s10664-016-9476-y.

Appendix

I. Glossary

Acronym	Description
RT	Requirements Traceability
NLP	Natural Language Processing
ML	Machine Learning
STS	Socio-technical System
SLR	Systematic Literature Review
IEEE	Institute of Electrical and Electronic Engineers
CMMI	Capability Maturity Model Integration
IR	Information Retrieval
STSE	Socio-technical Systems Engineering
QFD	Quality Function Deployment
ENRL	Estimation of the Number of Remaining Links
LSI	Latent Semantic Indexing
SPR	Scenario-Based Probabilistic Ranking
Pre-RS	Pre-Requirements Specifications
Post-RS	Post-Requirements Specifications
BPMN	Business Process Modeling Notion
SE	Software Engineering
RE	Requirements Engineering
RQ	Research Questions
GSD	Global Software Development
UML	Unified Modeling Language

II. List of papers reviewed in the SLR

Table 13 includes the papers which are reviewed in this dissertation. 46 papers were selected and the title of paper with the author and year is shown below.

Table 13: List of papers reviewed in the SLR

Author Name(s) & Year	**Paper Title**
(Khlif, Kchaou and Bouassida, 2022)	A Complete Traceability Methodology Between UML Diagrams and Source Code Based on Enriched Use Case Textual Description
(Kirikova, 2022)	Continuous Requirements Engineering in Sociotechnical Systems: Challenges and Solutions
(Batot, Gérard and Cabot, 2022)	A Survey-driven Feature Model for Software Traceability Approaches
(Pessoa *et al.*, 2022)	Model-Based Digital Threads for Socio-Technical Systems
(Maro *et al.*, 2022)	TracIMo: a traceability introduction methodology and its evaluation in an Agile development team
(Farooq, Ahmed and Emran, 2022)	A Survey on Blockchain Acquainted Software Requirements Engineering: Model, Opportunities, Challenges, and Future Directions
(Mosquera *et al.*, 2022)	OntoTrace: A Tool for Supporting Trace Generation in Software Development by Using Ontology-Based Automatic Reasoning
(Demi *et al.*, 2022)	A Neural Blockchain for Requirements Traceability: BC4RT Prototype
(Fucci, Alégroth and Axelsson, 2022)	When traceability goes awry: An industrial experience report
(Charalampidou *et al.*, 2021a)	Empirical studies on software traceability: A mapping study
(Paul *et al.*, 2021)	Scientific procedures and rationales for systematic literature reviews (SPAR-4-SLR)
(Mengist, Buffoni and Pop, 2021)	An integrated framework for traceability and impact analysis in requirements verification of cyber–physical systems
(Gul *et al.*, 2021)	Integrated Traceability Approach for an Effective Impact Analysis
(Zhao *et al.*, 2021)	Natural Language Processing for Requirements Engineering: A Systematic Mapping Study
(Charalampidou *et al.*, 2021b)	Empirical studies on software traceability: A mapping study

(Demi, Sanchez-Gordon and Colomo-Palacios, 2021)	Blockchain for requirements traceability: A qualitative approach
(Rubasinghe, Meedeniya and Perera, 2021)	SAT-Analyser Traceability Management Tool Support for DevOps
(Li *et al.*, 2021)	Multilevel Traceability Links Establishments Between SOFL Formal Specifications and Java Codes Using Multi-dimensional Similarity Measures
(Charalampidou *et al.*, 2021a)	Empirical studies on software traceability: A mapping study
(Wohlrab *et al.*, 2020)	Collaborative traceability management: a multiple case study from the perspectives of organization, process, and culture
(Arbain *et al.*, 2020)	Case study on non-functional requirement change impact traceability for Agile software development
(Singh and Kumar, 2020)	Empirical Validation of Requirements Traceability Metrics for Requirements Model of Data Warehouse using SVM
(Hidayati and Rochimah, 2020)	Requirements Traceability for Detecting Defects in Agile Software Development
(White, Krinke and Tan, 2020)	Establishing Multilevel Test-to-Code Traceability Links
(Yoshino and Matsuura, 2020)	Requirements Traceability Management Support Tool for UML Models
(Demi, 2020a)	Blockchain-oriented Requirements Engineering: A Framework
(Alkhammash, 2020)	Formal modelling of OWL ontologies-based requirements for the development of safe and secure smart city systems
(Ågren *et al.*, 2019)	The impact of requirements on systems development speed: a multiple-case study in automotive
(Krüger *et al.*, 2019)	Effects of Explicit Feature Traceability on Program Comprehension
(Murtazina and Avdeenko, 2019)	An ontology-based approach to support for requirements traceability in agile development
(Lin and Chen, 2019)	Comparative Study of Requirements Traceability in Facing Requirements Change: Systematic Literature Study and Survey
(Hayes, Dekhtyar and Payne, 2018)	The requirements tracing on target (RETRO).NET dataset
(Wang *et al.*, 2018)	Enhancing automated requirements traceability by resolving polysemy

(Santos *et al.*, 2018)	An Agile Modeling Oriented Process for Logical Architecture Design
(Lenarduzzi *et al.*, 2018)	Blockchain Applications for Agile Methodologies
(Rubasinghe, Meedeniya and Perera, 2018b)	Traceability Management with Impact Analysis in DevOps based Software Development
(Rubasinghe, Meedeniya and Perera, 2018a)	Software Artefact Traceability Analyser: A Case-Study on POS System
(Wang *et al.*, 2018)	Requirements traceability technologies and technology transfer decision support: A systematic review
(Cepulis and Niu, 2018)	Creating socio-technical patches for information foraging: A requirements traceability case study
(Mäder, Olivetto and Marcus, 2017)	Empirical studies in software and systems traceability
(Mustafa and Labiche, 2017)	The Need for Traceability in Heterogeneous Systems: A Systematic Literature Review
(Dey and Lee, 2017)	REASSURE: Requirements elicitation for adaptive socio-technical systems using repertory grid
(Arunthavanathan *et al.*, 2016)	Support for traceability management of software artefacts using Natural Language Processing
(Zhang, Wan and Jin, 2016)	An empirical study on recovering requirement-to-code links

III. Quality Scores Table

Table 14: Quality score table for papers

Papers reference	**RQ1**	**RQ1&RQ2**	**RQ2**	**RQ2 &RQ3**	**RQ3**	**Total**
(Murtazina and Avdeenko, 2019)	1	1	1	0.5	0.5	4
(Lin and Chen, 2019)	1	1	1	0.5	0.5	4

(Cepulis and Niu, 2018)	1	1	0.5	0	0	2.5
(Mengist, Buffoni and Pop, 2021)	1	1	0.5	0.5	0.5	3.5
(Demi, 2020a)	1	1	0.5	0.5	0	3
(Khlif, Kchaou and Bouassida, 2022)	1	1	0.5	0.5	0.5	3.5
(Batot, Gérard and Cabot, 2022)	1	1	1	0.5	0	3.5
(Hayes, Dekhtyar and Payne, 2018)	1	1	0	0	0	2
(Ågren *et al.*, 2019)	1	1	0.5	0	0.5	3.5
(B. Wang *et al.*, 2018)	1	1	1	0.5	0	3.5
(Alkhammash, 2020)	1	1	0.5	0.5	0.5	3.5
(Krüger *et al.*, 2019)	1	1	1	0	0	3
(Wohlrab *et al.*, 2020)	1	1	0.5	0.5	0.5	3.5
(Gul *et al.*, 2021)	1	1	0	0.5	0	2.5
(Zhao *et al.*, 2021)	1	1	0.5	0.5	0	3
(Dey and Lee, 2017)	1	1	0.5	0.5	0.5	3.5
(Arbain *et al.*, 2020)	1	1	0	0	0.5	2.5
(Mäder, Olivetto and Marcus, 2017)	1	1	0.5	0	0	2.5
(Zhang, Wan and Jin, 2016)	1	1	0.5	0.5	0.5	3.5
(Charalampidou *et al.*, 2021b)	1	1	0.5	0	0	2.5
(Charalampidou *et al.*, 2021a)	1	1	0.5	1	0.5	4

(Singh and Kumar, 2020)	1	1	0.5	0.5	0.5	3.5
(Santos *et al.*, 2018)	1	1	0.5	0	0	2.5
(Lenarduzzi *et al.*, 2018)	1	1	0	0.5	0	2.5
(Maro *et al.*, 2022)	1	1	0.5	0.5	0.5	3.5
(Murtazina and Avdeenko, 2018)	1	1	0.5	0.5	0.5	3.5
(Hidayati and Rochimah, 2020)	1	1	0.5	0.5	0.5	3.5
(Arbain *et al.*, 2020)	1	1	0	0.5	0	2.5
(Ågren *et al.*, 2019)	1	1	0.5	0.5	0.5	3.5
(Demi, Sanchez-Gordon and Colomo-Palacios, 2021)	1	1	0.5	0.5	0.5	3.5
(Wohlrab *et al.*, 2020)	1	1	1	1	1	5
(Jayatilleke and Lai, 2018)	1	1	0.5	2	0.5	5
(B. Wang *et al.*, 2018)	2	1	0.5	0.5	0	4
(Lin and Chen, 2019)	1	1	1.5	0.5	0.5	4.5
(I. Rubasinghe, Meedeniya and Perera, 2018b)	1	1	1	1	1	5
(Rubasinghe, Meedeniya and Perera, 2021)	1	1	0.5	0.5	0.5	3.5

(I. Rubasinghe, Meedeniya and Perera, 2018a)	1	1	0.5	0.5	0.5	3.5
(B. Wang *et al.*, 2018)	1	1	0	0.5	0.5	3
(W. Wang *et al.*, 2018)	1	1	0	0.5	0	2.5
(Santos *et al.*, 2018)	1	1	0.5	0.5	0.5	3.5
(Batot, Gérard and Cabot, 2022)	1	1	0.5	0.5	0.5	3.5
(Zhao *et al.*, 2021)	1	1	0.5	0.5	0.5	3.5
(Farooq, Ahmed and Emran, 2022)	1	1	0.5	0.5	0.5	3.5
(Mengist, Buffoni and Pop, 2021)	1	1	0.5	0.5	1.5	4.5
(Li *et al.*, 2021)	1	1	0.5	0	2.5	5
(W. Wang *et al.*, 2018)	1	1	0.5	0.5	0.5	3.5
(Mustafa and Labiche, 2017)	1	1	1	0.5	1	4.5
(Dey and Lee, 2017)	1	1	1	1	0	4
(Arunthavanathan *et al.*, 2016)	0.5	0.5	0.5	0.5	0	2
(Zhang, Wan and Jin, 2016)	1	1	0	1	0	3

License

Non-exclusive licence to reproduce thesis and make thesis public

I, **Chingiz Mammadli**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:
   1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and
   1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

of my thesis

A Systematic Review of Traceability in Requirements Engineering of Socio-technical Systems: Industrial Practices and Needs,

supervised by Ishaya Peni Gambo, PhD,

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, **05.01.2023**