Check for updates

Received: 26 August 2023 | Revised: 11 October 2023 | Accepted: 9 June 2024

DOI: 10.1002/smr.2707

SPECIAL ISSUE - EMPIRICAL PAPER

Journal of Software: Evolution and Process

WILEY

Trustworthy and collaborative traceability management: Experts' feedback on a blockchain-enabled framework

Selina Demi[1] | Mary Sánchez-Gordón[1] | Monica Kristiansen[1] | Xabier Larrucea[2,3]

[1]Department of Computer Science and Communication, Østfold University College, Halden, Norway

[2]Faculty of Informatics, University of the Basque Country (UPV/EHU), Vitoria, Spain

[3]Straumann Group, Madrid, Spain

Correspondence
Mary Sánchez-Gordón, Østfold University College, Halden, Norway.
Email: mary.sanchez-gordon@hiof.no

Funding information
Østfold University College

Abstract

Blockchain technology has attracted significant attention in both academia and industry. Recently, the application of blockchain has been advocated in software engineering. The global software engineering paradigm exacerbates trust issues, as distributed and cross-organizational teams need to share software artifacts. In such a context, there is a need for a decentralized yet reliable traceability knowledge base to keep track of what/how/when/by whom software artifacts were created or changed. This study presents a blockchain-enabled framework for trustworthy and collaborative traceability management and identifies benefits, challenges, and potential improvements based on the feedback of software engineering experts. A qualitative approach was followed in this study through semistructured interviews with software engineering (SE) experts. Transcripts were analyzed by applying the content analysis technique. The results indicated the emergence of five categories, further grouped into three main categories: experts' perceptions, blockchain-based software process improvement, and experts' recommendations. In addition, the findings suggested four archetypes of organizations that may be interested in blockchain technology: distributed organizations, organizations with contract-based projects, organizations in regulated domains, and regulators who may push the use of this technology. Further efforts should be devoted to the integration of the proposal with tools used throughout the software development lifecycle and leveraging the potential of smart contracts in validating the implementation of requirements automatically.

KEYWORDS
blockchain technology, content analysis, experts' judgment, neural distributed ledger, requirements engineering

1 | INTRODUCTION

The field of software engineering (SE) has been defined by the IEEE, as "*the application of engineering to software*". Traditionally, SE projects have been characterized by co-located team members with the same culture and language and a homogeneous software toolset with limited features to support collaboration.[1] However, the emergence of the "global software engineering" (GSE) paradigm which refers to software being developed by various stakeholders that are distributed across geographical, organizational, temporal, linguistic, and social-cultural boundaries[2,3]

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.
© 2024 The Author(s). Journal of Software: Evolution and Process published by John Wiley & Sons Ltd.

demands greater support for collaboration, coordination, and information sharing.[1,4] This becomes complex in the context of mistrust and power struggles among remote stakeholders in inter-organizational software projects.[5] The increased complexity has been attributed to the distance produced between collaborative entities, due to organizational borders.[6] Previous studies have advocated traceability as a viable way to bridge this distance.[6]

According to Spanoudakis and Zisman,[7] traceability refers to "*the ability to relate artifacts created during the development of a software system to describe the system from different perspectives and levels of abstraction with each other, the stakeholders that have contributed to the creation of the artifacts, and the rationale that explains the form of the artifacts*". Traceability contributes to keeping track of what requirements have been specified, implemented, and tested,[8] hence supporting requirements coverage and verification analysis.[7,9] In particular, traceability contributes to maintenance tasks, by facilitating the analysis of changes' implications and integration and by enhancing the comprehension of the system.[8,10,11] The clear and consistent documentation of systems enabled by traceability[7] has been imposed by software process improvement (SPI) models and standards, for instance, CMMI (Capability Maturity Model Integration), ISO (International Organization for Standardization) 26262, and ASPICE (Automotive Software Performance Improvement and Capability dEtermination). Indeed, a particular emphasis has been put on the importance of traceability in safety-critical systems, as improper specification, design, implementation, and testing can cause life-threatening accidents.[8] Moreover, safety-critical systems are characterized by a large number of artifacts created by distributed teams, including inter-organizational teams.[8] For instance, Maro et al.[8] pointed out that requirements for a typical high-end car amount to approximately 100 million lines of code.

Conventional centralized repositories used to store and share software artifacts cannot be relied upon, in the context of interorganizational and complex software projects composed of a set of distributed stakeholders that do not trust each other.[12] Despite the security measures that can be taken, such as access control policies, centralized approaches cannot ensure reliability, immutability, and availability of software artifacts. Therefore, decentralized yet trustworthy solutions are needed to challenge existing centralized approaches. In this scenario, blockchain technology has been proposed as a technology capable to fulfill decentralization and trustworthiness requirements.[12–14] Blockchain is a peer-to-peer (P2P) distributed ledger technology (DLT) that stores transactions in an immutable chain of blocks. While blockchain has been widely considered a DTL in literature,[15–17] the use of cryptography for secure data transmission and records' immutability in a decentralized context has enabled this technology to overcome former DLTs.[15] Therefore, from a data structure perspective, blockchain is immutable and append-only, meaning that new blocks get appended to the end of the ledger by specifying and pointing to the hash of the previous block. The main utility of blockchain lies in the disintermediation and decentralization of transactions while retaining security.[18] This has been enabled by integrating a set of core existing technologies, such as hashing, timestamping of transactions, digital signatures based on asymmetric cryptography, P2P networks, and distributed consensus mechanisms.[15,19] Hence, blockchain provides a distributed yet trusted mechanism in which parties can verify that the order and transaction timestamps have not been modified.[20] Ultimately, from a business perspective, the utility of blockchain lies in streamlining and automating business processes and legal constraints.[15] As outlined in Swan's book[18]: "*Blockchain: Blueprint for a new economy*", blockchain can be used to register and track assets of any type, beyond financial assets, for example, physical properties, votes, ideas, and health data.

In this study, blockchain technology is used to register and track software artifacts of any type and size, in a decentralized, yet secure manner. This study builds on top of our previous work[21] and aims to identify challenges, benefits, and potential improvements of the proposed framework and prototype by means of SE experts' feedback. Ultimately, the study contributes to the enhancement of the existing knowledge in the promising yet still limited blockchain-enabled SE field.

This paper is organized as follows: Section 2 presents our proposed solution which is composed of two main components: (i) a framework to guide the implementation process of blockchain technology in SE projects and (ii) a prototype for trustworthy and collaborative traceability management that enables sharing software artifacts in a decentralized way while retaining security. Section 3 explains the research method adopted in this study to provide feedback on the proposed solution. The research method section entails the following subsections: research design (data collection and sampling), data analysis, and ethical considerations. Section 4 unveils the emergent categories which are discussed in Section 5. Section 6 concludes the study and proposes directions for future research.

2 | SOLUTION PROPOSAL

2.1 Framework

This section presents a novel framework that aims to facilitate the implementation of blockchain in inter-organizational software projects. Figure 1 depicts the framework which follows an iterative process consisting of the following phases:

1. **Identify the importance and need for distributed requirements traceability (RT).** The first phase of the framework consists of defining the need of RT in inter-organizational software projects. Ensuring trustworthy RT is important in inter-organizational software projects as a proof-of-quality and correctness to avoid expensive disputes.[6] Moreover, traceability enables tracking the project's progress and verifying its adherence to legal regulations. While it is true that traceability information should be shared among collaborative entities, previous research has

20477481, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2707 by Readcube (Labtiva Inc.), Wiley Online Library on [11/07/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

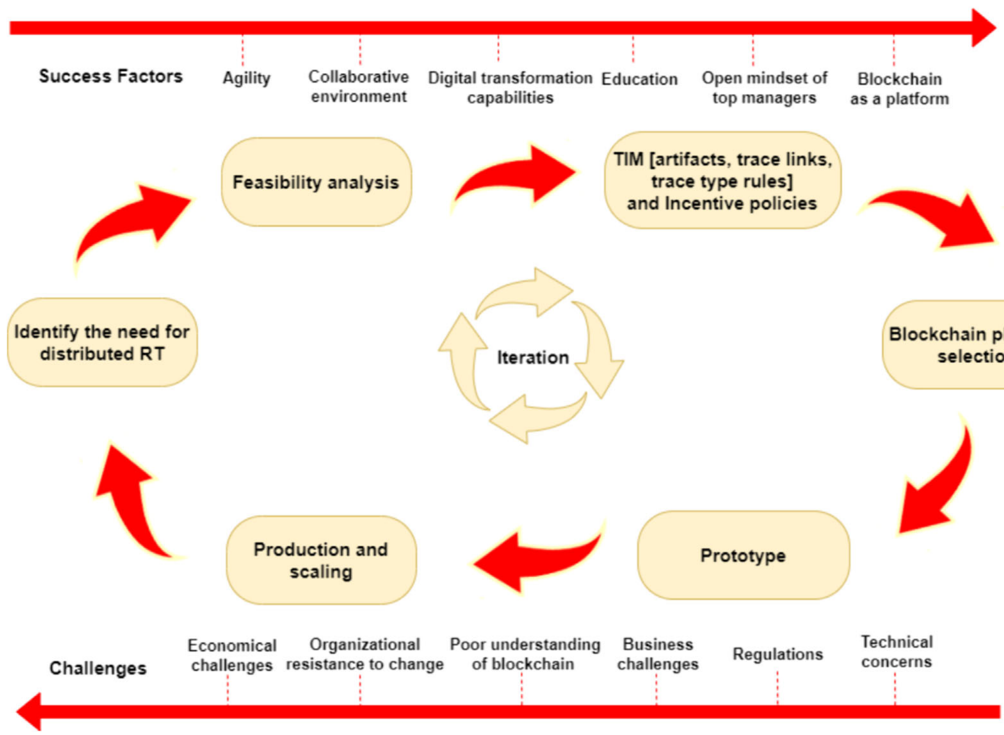


FIGURE 1 Implementation framework for blockchain-enabled traceability management.[22]

reported on the resistance of entities to disclose confidential information and trust the information shared by other entities due to organizational boundaries.[6] This implies a set of requirements for traceability information: available, shared, and reliable (trusted). After defining requirements, organizations need to select a technical solution that has the potential to enable an available, shared, and trusted traceability knowledge base which provides transparency and visibility to all legitimate participants regarding what/how/when software artifacts were created and/or modified and by whom.

2. **Feasibility analysis.** The second phase entails analyzing whether blockchain can be a feasible technical solution to achieve trustworthy RT and compare it with existing technologies, for example, traceability matrices, relational databases, and graph traceability repositories. The analysis should consider two dimensions: (i) legal dimension. Blockchain can serve as a single source of truth to prevent expensive disputes by clients who may claim that the software product was not delivered the way they expected. (ii) Blockchain-as-a-platform. Blockchain can contribute to the creation of a trustworthy ecosystem that consists of a variety of software development lifecycle (SDLC) stakeholders that do not necessarily trust each other, for example, customers, distributed teams, third-party vendors, and regulators. Success factors and challenges that need to be considered to implement successfully blockchain for RT are depicted in Figure 1 and explained in our previous paper.[22]
3. **Traceability information model (TIM) and incentive policies.** The next phase focuses on defining TIM which means defining the types of artifacts to be traced and trace links. This has been considered a best practice to ensure consistent results in multistakeholder projects,[23,24] and it helps in designing the blockchain-enabled prototype. Incentive policies can also be formulated to motivate stakeholders to engage in traceability tasks. It is noteworthy that these incentive policies have not been adopted in the proposed prototype (See Section 2.2).
4. **Blockchain platform selection.** In a previous study,[22] authors identified a set of factors to facilitate the blockchain platform selection process: network accessibility, that is, the choice between public and private blockchain, may depend on the nature of software to be developed, transaction fees, consensus mechanisms, programmability, and community of developers. This study focuses on the development of large-scale software as a collaborative effort of a set of diverse yet known participants. Therefore, the platform should allow the responsible party for the project to decide who to share software artifacts with. In addition, the platform should be performance-efficient and sustainable which goes against existing competitive-based consensus mechanisms, such as proof-of-work or proof-of-stake, and efficient from an economical perspective, that is, lack of transaction fees. Therefore, the authors selected NDL (Neural Distributed Ledger) ArcaNet as a collaborative P2P platform that fulfills the aforementioned requirements.[21]
5. **Prototype, production, and scaling.** The last two phases rely on developing a prototype that relies on blockchain and shifting it into production. However, in our previous paper, we found out that most blockchain projects stop at the prototype stage,[22] due to the innovation–production gap. While addressing this gap is important for both academic and industrial settings, it is not within the scope of this study. The authors of this study focused on developing a prototype that allows software development life cycle (SDLC) stakeholders to register

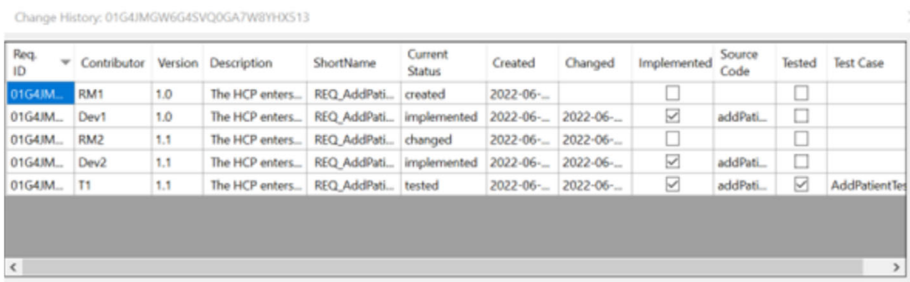
20477481, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2707 by Readcube (Labtiva Inc.), Wiley Online Library on [11/07/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

requirements and their related source code files and verification files collaboratively and securely. Future work should be dedicated to making this prototype fully functional and shifting it into production. A demo of the prototype has been presented in a video that can be accessed online[*] and explained in Section 2.

2.2 Prototype

In order to trace the lifecycle of requirements created and/or changed by diverse and distributed stakeholders throughout the SDLC, the authors developed the BC4RT (Blockchain for RT) prototype. The implementation of this prototype relies on the use of a P2P collaborative platform, namely, NDL ArcaNet. The primary advantage of this platform compared with other blockchain platforms is the capability to protect and transfer securely digital assets of any size and type that are represented in the form of tokens.[25] Figure 2 presents a sample of the user interface of the proposed prototype. Additional information regarding the prototype is provided in our previous paper.[21]

The proposed prototype relies on the concept of requirements as valuable and unique digital assets represented by digital tokens. Requirement tokens are created and/or updated collaboratively by SDLC participants who share a secret key. Tokens and operations on them are stored in a secure token repository which is transparent and visible to network parties and validated by trusted certifiers. The underlying block structure of the prototype is depicted in Figure 3.



Change History: 01G4JMGW6G4SVQ0GA7W8YHXS13

Req. ID	Contributor	Version	Description	ShortName	Current Status	Created	Changed	Implemented	Source Code	Tested	Test Case
01G4JM...	RM1	1.0	The HCP enters...	REQ_AddPati...	created	2022-06-...		☐		☐	
01G4JM...	Dev1	1.0	The HCP enters...	REQ_AddPati...	implemented	2022-06-...	2022-06-...	☑	addPati...	☐	
01G4JM...	RM2	1.1	The HCP enters...	REQ_AddPati...	changed	2022-06-...	2022-06-...	☐		☐	
01G4JM...	Dev2	1.1	The HCP enters...	REQ_AddPati...	implemented	2022-06-...	2022-06-...	☑	addPati...	☐	
01G4JM...	T1	1.1	The HCP enters...	REQ_AddPati...	tested	2022-06-...	2022-06-...	☑	addPati...	☑	AddPatientTes

FIGURE 2 User interface sample: “Trace requirements' life cycle”.

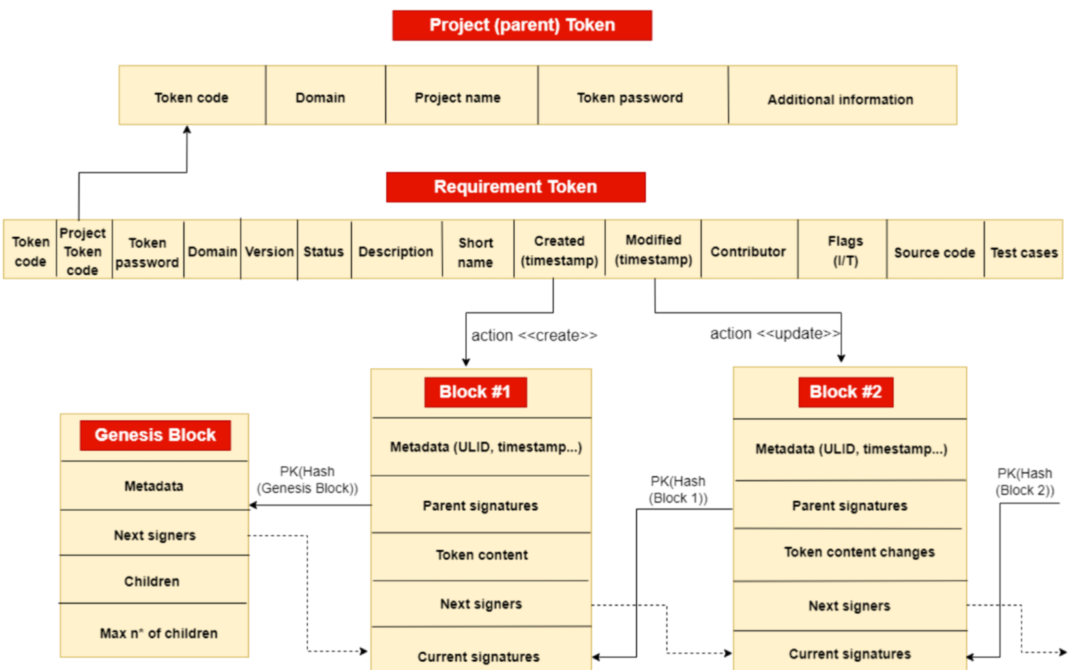


FIGURE 3 Underlying blocks' structure.[21]

20477481, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2707 by Readcube (Labtiva Inc.), Wiley Online Library on [11/07/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Figure 3 shows the creation of two types of tokens: project token as parent token, and requirement token as the child of project token. In order to track the lifecycle of tokens, it is necessary for each token to issue its own signed blockchain ledger. The two types of tokens consist of a set of fields:

i. Parent token: token code expressed in ULID (Universally Unique Lexicographically Sortable Identifier) format, project domain, project name and password, and additional information about the project.
ii. Requirement token: token code expressed in ULID format, project token code that points to the parent token, token password, requirement domain, version, current status of the token (created, changed, implemented, tested), requirement description and short name, timestamp of when the requirement token was created and/or modified, a contributor who performed the create or update action on the token, flags (implemented and/or tested), source code files, and test cases.

The <create> action enables the issuance of block #1 which consists of the following fields: ULID, timestamp, previous block signatures, content of requirement token's fields, signers of the next block, and current block signatures. The consequent blocks issued as a result of the <update> action contain only token changes, not the whole token content. Finally, network parties provide a genesis block randomly for any first block. The genesis block (metadata, hashes) is hosted by trusted nodes to prevent the illegitimate recreation of history from the very beginning and to ensure immutability.

3 | RESEARCH METHOD

3.1 Research design: Sampling and data collection

This paper adopts a qualitative approach to get feedback on benefits, challenges, and potential improvements of the proposed framework and prototype from the perspective of SE experts. The reason behind this choice lies in the fact that the goal of the paper is not to achieve generalizability of findings. In such a case, a quantitative approach with a larger number of participants would have been more suitable. Instead, this paper aims to interpret recommendations of SE experts as requirements for future improvements of the prototype; therefore, a qualitative approach is more aligned with this aim, as it can unveil rich and in-depth information.[26]

A series of semistructured interviews were carried out with SE experts in order to provide feedback on the proposed framework. The authors chose semistructured interviews as the main data collection technique, because of their flexible nature that enables uncovering rich contextual information.[26] Interviews were conducted through Zoom and recorded upon interviewees' consent. An interview guide which can be accessed in an online repository[27] was used to guide the interview process within the 45 min timeframe allocated to each participant. A set of predefined open-ended questions were designed to capture experts' opinions and impressions regarding the different content areas of the proposed framework. The interviews started with background questions aimed to probe for interviewees' professional and academic experience in the SE domain. The following questions aimed to probe for interviewees' perceptions of the proposed framework, experts' opinions on the usefulness and challenges of the framework, and their recommendations for potential improvements. Additional questions were asked to the interviewees when the interviewer perceived the need for clarification or the potential for the pursuit of new relevant insights. The informed consent, a sample of questions, and a video demonstrating the prototype were sent to the participants prior to their interview.

The experts were purposively selected, due to their professional and academic experience in SE. Purposive sampling is the most common methodology for participant selection in SE research.[28,29] The experts have a broad experience in a variety of SE dimensions, such as requirements engineering (RE), software development, testing, and SPI in both organizational and academic settings.Table 1 shows that About 70% (7) of the experts have over 10 years of experience in SE in academia and/or industry. From an academic perspective, 90% (9) of the experts hold a PhD degree and have teaching experience in SE. Since the experts have broad and diverse experience, defining their roles was not trivial. However, the authors focused on their current roles and found out that 60% (6) are employed as SE researchers and/or professors and 40% (4) are focused on quality assurance and SPI. Finally, although more women in SE were invited to participate in this study, the majority of the experts (70%, 7) who agreed to participate were men .

3.2 Data analysis

The abundance of raw data generated from the interviews drives the need for a systematic analysis technique that distills words into content-based categories. Therefore, the authors chose content analysis as the most suitable analysis technique, due to its content-sensitive nature[30] and research design flexibility.[31] According to Krippendorff,[30] content analysis is an empirically grounded research technique of an exploratory nature.

20477481, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2707 by Readcube (Labtiva Inc.), Wiley Online Library on [11/07/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 1 Experts' demographics.

Characteristic	Value ($n = 10$), n (%)
Gender	
Male	7 (70%)
Female	3 (30%)
Education level	
PhD	9 (90%)
Master's degree	1 (10%)
Experience (years)	
0–10	3 (30%)
>10	7 (70%)
Professional experience in SE	
No	2 (20%)
Yes	8 (80%)
Teaching experience	
No	1 (10%)
Yes	9 (90%)
Role	
SE researcher/professor	6 (60%)
Quality assurance/process improvement and assessment expert	4 (40%)

Abbreviation: SE, software engineering.

This technique has been used for many years as a scientific tool to make replicable and reliable inferences from semantic data to their specific context. Although the roots of content analysis can be traced back to the 19th century when it was mainly used to analyze newspaper articles and political speeches,[31] this method has been recently used also in technical fields, such as SE.[32] In this study, the authors adopted the content analysis process proposed in Elo and Kyngäs,[33] as it has achieved a high impact in terms of citations (#25643, Google Scholar, accessed on 06.10.2023). It is also worthy to note that 357 citations were related to SE. The authors adapted the content analysis process,[33] as depicted in Figure 4 and explained in the following paragraphs:

1. **Preparation phase.** This phase started with the selection of the unit of analysis. The unit of analysis can be a word or theme, a unit of meaning composed of one or more sentences, the number of participants, or whole interviews. The authors in Graneheim and Lundman[34] pointed out that a narrow analysis unit, such as one word, may lead to fragmentation and suggest whole interviews as the most suitable unit of analysis, as they are enough large to obtain a sense of the whole and enough small to serve as a context for meaning units throughout the analysis process. The suggestion was followed by this interview study, and whole interviews were selected as the unit of analysis. Next, the authors uploaded the transcripts in Nvivo and read thoroughly all the transcripts to make sense of the data.
2. **Organizing phase.** The second phase aims to organize the qualitative data. The first author read again the transcribed text and broke it down into sentences or paragraphs that represent meaning units. Then, the researcher assigned a code to each meaning unit. This procedure has been referred to as open coding.[33] In this study, the open coding was done iteratively, and the list of codes was updated continuously, as more relevant data became available. Codes were collected in coding sheets, along with meaning units and condensed meaning units. The number of words used in extended meaning units was reduced and condensed meaning units were created.[35] The coded material was divided into three main content areas based on interview questions, as suggested in Graneheim and Lundman.[34] Codes were then grouped and categories were generated.
   To reduce the number of categories, they can be further grouped and this process which is referred to as abstraction can continue until a reasonable explanation has been achieved.[35]
3. **Reporting phase.** The last phase entails detailed descriptions of the categories' contents. Previous literature has highlighted the importance of demonstrating results-data links to enhance the reliability of the study.[35,36] A few examples that demonstrate how the categories emerged in this study are depicted in Appendix A (Table A1). Further, the categories are explained in detail in Section 4.

20477481, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2707 by Readcube (Labtiva Inc.), Wiley Online Library on [11/07/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

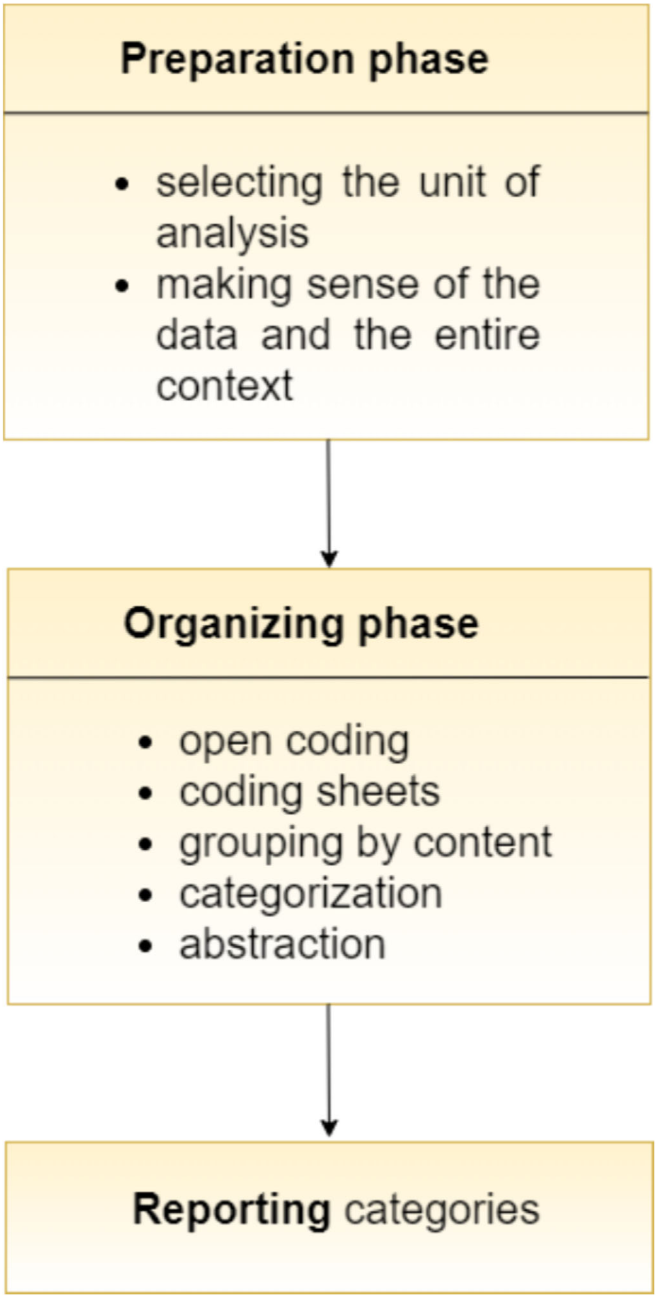


FIGURE 4 Content analysis process, adapted from Elo and Kyngäs.[33]

3.3 Ethical considerations

A detailed informed consent form was sent to all the participants prior to their interviews. This form was approved by the Data Protection Officer at Østfold University College to ensure General Data Protection Regulation (GDPR) compliance and can be accessed online.[27] The informed consent outlined that participation in the study was voluntary and anonymous. In addition, the informed consent form provided information regarding the purpose of the study, motivation of the selection of interviewees, information regarding recording, storage and deletion of data, benefits and risks associated with the study, and rights of interviewees. Although the study did not collect or handle sensitive data, it was registered and sent for approval to the Norwegian Center for Research Data (NSD).

20477481, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2707 by Readcube (Labtiva Inc.), Wiley Online Library on [11/07/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

4 | RESULTS

This section presents SE experts' feedback on the proposed framework and prototype. The analysis of the data suggested five categories, which were then classified into three core categories: experts' perceptions, blockchain-based SPI, and experts' recommendations. The categories are explained in the following section.

4.1 Experts' perceptions

The experts were presented with the proposed framework and prototype and were probed into unveiling their impressions on such a proposal. In general, the experts expressed positive feedback about the proposal and the transformative potential of blockchain technology in the SE domain. A set of terms were used by the experts to describe the blockchain-enabled RT framework and prototype: *innovative* (EX7), *interesting* (EX1, EX4, EX10), *promising and sound* (EX1), *rudimentary but functional* (EX2), *necessary* (EX4), *feasible* (EX1, EX3), *customizable and worthy to try* (EX1), *automation system* (EX5), *useful for RT* (EX8), and *makes a lot of sense to use it for requirements management and traceability* (EX9).

EX3 perceived the blockchain-enabled prototype as a distributed version control system that enables configuration management activities: "*what we have here is a distributed version control system, for me it seems like a technological solution for enabling configuration management activities. For doing that, I think it is a good approach … Conceptually, a technological solution like blockchain fulfills the necessities of configuration management*". Similarly, EX2 considered the approach as useful for enhancing change management: "*it's potentially useful as a concept because it clarifies or makes obvious when that change happened. That's an obvious major benefit here because this can help to reduce uncertainty in the future about why a change happened, how it happened, or when it happened*". Finally, the prototype was perceived as easy to use by two of the experts (EX3, EX4), as follows:

> "*it is more the idea to provide non-functional aspects like traceability than waging a very complex application*"
>
> (EX3).

> "*it is simple to use but useful, because most of the times we try to provide very complex interfaces that are difficult to use … I have some commercial tools related to requirements traceability and they are difficult to use because you can only have the requirements, architecture elements, and test cases*"
>
> (EX4).

4.2 Blockchain-based SPI

SPI relies on the underlying values of business focus, people and organizational change.[37] These values emerged as subcategories in the data analysis process and are explained in the following sections.

4.2.1 Business

Identifying the organizational pain points before deciding to implement blockchain technology has been outlined by EX5. In fact, this expert suggested: "*you should forget about blockchain at the very beginning when you try to adapt for an organization. You do not need to mention the blockchain, but everything should be done with the requirements of the blockchain*". Moreover, the experts agreed that the main benefit of the proposed blockchain system is traceability.

> "*Traceability would be heightened to the use of the type of framework you are proposing*"
>
> (EX2)

> "*We need to trace what is happening, so traceability is very important in the development of any engineering system*
>
> " (EX3)

> "*It will definitely improve requirements traceability …*"
>
> (EX9)

20477481, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2707 by Readcube (Labtiva Inc.), Wiley Online Library on [11/07/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Likewise, EX6 highlighted the importance of traceability and revealed how traceability is currently addressed in the automotive domain: "*you will find several configuration management tools which are available in the market, for most of OEM (Original Equipment Manufacturer), or most of automotive suppliers to use. So, basically by using configuration management concepts, versioning techniques, baselining techniques, you can freeze a certain set of requirements, trace it, store your traceability reports, have a baseline that cannot be changed … This is being stored on the cloud*". In addition to traceability, blockchain enables the instant sharing of information among distributed stakeholders in a trustworthy and secure manner (EX1, EX3, EX7), ensures transparency over the SDLC (EX1, EX3, EX5, EX7), nonrepudiation (EX1), reliability and availability of information (EX9), improved communication between customers and development teams, and code reusability, as source code artifacts are stored on blockchain (EX7). These benefits may potentially provide software organizations with significant returns, due to time-saving and enhanced project quality (EX7, EX9).

However, the experts raised several concerns regarding the implementation of blockchain in organizational settings, too. EX5 stated that blockchain may not be suitable for all organizations. The expert made an analogy between the adoption of blockchain and agile methodologies: "*In Turkey, many companies try to adapt to Agile, but they do not have the culture … They think Scrum is the holy grain that solves all the problems*". Furthermore, despite mentioning the inherent limitations of blockchain systems, such as performance (EX2, EX3, EX5) and costs (EX4, EX5, EX7), the experts discussed primarily the perception of documentation as overhead. EX5 raised the concern that highly complex documentation of requirements may face resistance from software engineers, as it may require effort and affect the project's efficiency. "*It is good to trace back, but it could be too much details. You need to avoid at some stage this kind of high complex documentation. You are creating enormous amount of documentation or processed work … That could affect the efficiency and speed*".

EX2 claimed that the high documentation of requirements can be perceived as anti-agile software development: "*It is our presupposition that this (proposed prototype) is happening in an environment where people actually take the effort of documenting requirements to a reasonably high level. There are a lot of settings today where that doesn't happen. Or maybe what you get is a user story, one sentence on a poster stuck on a wall and moved along a Kanban board and this is your requirement. So, these people inherently do not want to be typing stuff into systems and tracking requirements. It is almost anti-Agile software development, and anyone who is realistic about building enterprise grade software systems, which is many, they know they have to document requirements and track changes*" (EX2). EX8 was skeptical and claimed that neither software engineers nor project managers are willing to document RT, due to the high workload. According to EX8, who has experience in developing software for the banking domain, project managers have on average nine projects to manage. EX2 also noted that the prototype is unlikely to be adopted by organizations that do not document requirements. "*There's a certain end of the market that isn't going to want to do this. For people who don't document requirements other than writing them down on posters and putting them up on the wall. You're not going to convince them to start documenting and they're going to tell you that they don't need to know that much about requirements. Now that doesn't mean they're right, but that will be their philosophical position on the matter*" (EX2). Likewise, EX10 confirmed the reluctance to document artifacts manually: "*If I would need to add every file I created for a software manually to the system, I would never do it in such a way to document it …*".

Other experts emphasized that the overhead created as a result of documenting requirements should not justify the lack of documentation, even in simple projects with a low number of requirements (EX6) or projects that adopt Agile methodologies (EX7). In fact, documenting requirements is required also in Agile software development, although the documentation can be performed module-by-module (EX7). EX7 outlined that documenting requirements and their changes facilitate the maintenance of systems, as it enables the maintenance team to review existing documentation when new features need to be added in the future (EX7). This should encourage software engineers to show empathy towards future maintainers, by documenting software artifacts and their metadata (EX9). While it is true that compassion is a human factor, it is also true that its incorporation in software development may enhance the software development process[38] and consequently have a positive business impact. In addition, the expert emphasized the need to monitor the entire software lifecycle, and this can be achieved solely by putting effort into documenting quality artifacts and metadata: "*When I was working with Rational Rose, the only way that the tool was going to be useful for us, is if we input information to it. If garbage input, we will have garbage output, so it's not only that you have to input the information you have to make sure the quality of the information or metadata you input in the tool*"(EX9).

The importance of documenting quality and accurate artifacts in a timely and continuous fashion was also mentioned by EX10. The expert outlined the risk of incorrect or fake entries being stored on blockchain which can endanger the trust that is expected to be ensured inherently by blockchain technology. In fact, this is not an issue that is specific to the proposed system, as it occurs also in issue tracker systems. EX10 has analyzed the logs of an issue tracker in a specific company where the story cards had to be moved through a board that represented the defined workflow. The expert was interested in the time needed by developers to finish a story card and found out that "*one of the developers needed on average 2 minutes. After talking with the company, we found out that this time is not possible. The developer simply moved the cards through the board, as he frequently forgot to document his work and had the impression that this is not important*". Therefore, it is necessary to enforce mechanisms that ensure that the documentation is done timely and continuously.

The information provided by the experts was useful in defining the archetype of organizations that might be interested in the proposed framework. The data suggested four main archetypes of potentially interested software and customer organizations: (i) organizations composed of teams distributed globally; "*We work in a distributed software development context, and the information and developer teams are distributed over the globe. So, we want to share our information, share data, share the rest of the artifacts of our projects, but organizations first need to think about*

20477481, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2707 by Readcube (Labtiva Inc.), Wiley Online Library on [11/07/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

security aspects. Blockchain is one of the ways that might be helpful to global software development and to make available information for all of us instantly, securely"(EX7); (ii) organizations that are willing to document requirements and their changes, often in contract-based projects. EX2 outlined that the prototype can be adopted in those projects where contracts with customers are established: "*anyone who has customers usually has contracts with the customers and usually requirements change management is a key part of that contract*". If requirements are agreed upon and signed by the customer, their changes can be managed in a legally defensible way, by means of blockchain technology (EX2). As mentioned before, EX2 also claimed that organizations that do not document requirements are unlikely to be adopt the proposed prototype; (iii) customer organizations operating in domains that are subject to regulations. Organizations that operate in safety-critical and highly regulated domains, such as healthcare, aerospace, automotive, defense, and nuclear power generation, can demonstrate adherence to regulations, by storing records on blockchain, as was highlighted by the majority of experts (EX2-EX6, EX9, EX10); and (iv) regulators. EX8 who has experience in developing software for the banking sector emphasized the following: "*I experienced people who came from the central bank and asked for evidence*". Therefore, this expert outlined the potential of companies that develop regulations and policies to push the use of blockchain technology.

4.2.2 People

According to the experts, RT is challenging in global software development, due to cultural (EX1, EX7, EX9), language (EX7, EX9), and religious diversity (EX7). As EX1 stated: "*There are developers in India, China … who have a different attitude towards security issues … The value systems are different. And the issues of trust, security, and transparency are extremely sensitive to these cultural issues*". Blockchain can contribute to address these issues (EX1). However, the main challenge in the broader field of RE lies in communication. In fact, communication issues are inherent human limitations that cannot be addressed solely by technological solutions, such as blockchain (EX2). EX2 elaborated on this issue: "*I am building a software system for you. And you tell me certain things you want and I listen, and then I try to define them as requirements. And then, I implement them. There's a huge gap in this whole space. The first gap arose when you told me what you wanted, because you think you know what you want. You have an idea of what you need, not what you want. And even in knowing what you need, you might not necessarily thought about it thoroughly enough for it to be a software requirement. So, there is a huge communication breakdown here because as humans we only have limited time to communicate, we use a certain amount of words, nearly every word is given to multiple interpretations*". Recent discussions in the SE research community have framed communication as a "*resource to be invested, particularly to refine project requirements*".[39] This framing also outlines the trade-off between investments and their returns and between resource allocation and resource management.

Although software engineers are more likely to understand innovative technologies compared with practitioners in other domains (EX1), in practice, it is difficult to find software engineers who are knowledgeable in blockchain, and it seems that they are not interested in learning this technology (EX5). In this regard, EX5 shared his experience: "*When we tried to have this project (integration of microservices with blockchain) in Turkey, the problem is that we could not really find someone who knows good enough microservices, so we can educate with blockchain. And they are not interested actually. They wanted to use their talent and do something with microservices, but when we tried to teach them blockchain, they said we are not security people. This can be one of the problems when you want to make a real implementation*". According to this expert, it is necessary to enhance the understanding of software engineers about the concept of immutability. "*They need to understand immutability, even though they know immutable databases, they do not really understand immutable blocks concept*" (EX5). Training and upskilling employees with blockchain knowledge have been advocated by our experts (EX4, EX7, EX9).

4.2.3 Change

All the experts discussed resistance to change when introducing a new technology in the SE context. This is not surprising, as humans are inherently resistant to change, due to the effort that change requires and the risks associated with change (EX2). Moreover, EX2 acknowledged that a blockchain-enabled system to store and share software artifacts might face resistance from companies that do not need to document requirements and their changes. "*Companies that do not spend a lot of time documenting requirements, who prefer to write software and get feedback on it, which is not necessarily a particularly smart way to build software, but lots of people do … These companies might just be resistant to documenting anything. And, in that context, you'd get it difficult to convince them to use a tool that had made them formally write up the requirements and every time they changed. They might say it was unnecessary*"(EX2). Furthermore, EX3 pointed out the performance limitations of blockchain-based systems, as a factor that makes SE practitioners reluctant: "*people tend to be reluctant to include things that add in terms of time … If I want to create any artifact and then the process of validating that and write the blocks takes one hour … That could be problematic*".

The experts expressed diverse opinions regarding the conservative trait of software engineers. EX6 who is a SPI expert working in the automotive domain stated that automotive software engineers are aware of new technologies and are not change-resistant, if they are convinced about the benefits of such a change. On the contrary, EX1 who has experience as the head of quality management in a software research center and EX5 who works mainly with developing serious games for the educational domain perceived software engineers as conservative:

20477481, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2707 by Readcube (Labtiva Inc.), Wiley Online Library on [11/07/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

> *"In my experience, there are software engineers that are quite conservative … They are used to use technologies which they are familiar with, and they are quite reluctant to embark on new technologies. That is contradictory to their capacity to change, because they would understand the technology"*
>
> (EX1).

> *"According to my experience, teaching something to developers is very hard. They are called mostly conservatives, sticking to what they know. Then you try to teach them something new, that could affect efficiency and speed"*
>
> (EX5).

To mitigate the resistance to change and push the adoption of the proposed system, EX9 suggested a top-to-bottom approach with policies at the organizational or project level in place. These policies should be enforced in a positive manner, for instance, by rewarding software engineers who are contributing to introduce the new technology throughout the organization (EX9). However, it should also be considered that the implementation of the proposed blockchain system is not solely a technical and organizational decision but on a broader scope; it is a decision of the entire software ecosystem (EX8). EX8 elaborated on this topic, as follows: "*When you provide services to other companies, you want to work with the same technology as the other company. So, it was not your decision, it is a decision of the whole environment or the whole industry. For example, if you change to a database as Oracle, you cannot change to PostgreSQL, even if it is better, because the other company is not going to accept that, the same with management tools*".

4.3 Experts' recommendations

The experts provided their recommendations for the improvement of the proposed framework and prototype. The core recommendation of the experts was the lightweight integration of the proposal with existing commercial tools that are used throughout the SDLC, for instance, Rational Solution for Collaborative Lifecycle Management (EX1), JIRA (EX5, EX10), Kanban boards (EX2), and GitHub and GitLab (EX10). According to the experts, the integration with existing tools may encourage SE practitioners to adopt the blockchain-enabled prototype. As EX2 stated: "*if you want your technology to be adopted or attractive to them (SE practitioners), if it could integrate with existing systems, then this would make a big difference. If you actually had requirements on blockchain, linked to user stories, and if that could be visualized, so you could bring user stories to life as blocks on blockchain on a Kanban board for development, then you'd be giving something that they could use as a more kind of total solution … The last thing you want to say to engineers is here is another place, I want you to do this (record artifacts) … What you really want to do is get them to start documenting in your system and stop documenting somewhere else. And to really make that work, you will have to be able to visualize. For example, you could integrate a timeline on a Kanban board into your software*". Moreover, EX5 pointed out that compilation issues are not included in the proposed prototype and suggested the integration with JIRA to handle such issues. EX5 raised the following concern: "*you are proposing an automation system, but most of the people in industry have automation systems like JIRA and Confluence. So, it's like you are proposing a new religion to them … Why don't we just use another add-on instead of this? JIRA has enormous amount of add-ons …*" Likewise, EX2 suggested the integration with existing tools rather than completely disrupting the existing way of working.

Furthermore, the experts provided the following recommendations:

i. Change rationale. EX2 pointed out that the prototype does not consider the rationale behind changes. "I could log in as a developer, and I could say, I am changing this requirement. But not say, why? And you might even want to have a little bit more around that to see, has it been demonstrated that this is something we want to do? Is there a value in doing this?" (EX2). This expert also highlighted the importance of avoiding the inclusion of extra features that add risks instead of value. This phenomenon has been referred to as gold plating in SE (EX2). Likewise, gold plating was identified as a critical challenge in a recent survey performed by Hoffmann et al.[40] In the RE community, gold plating has been defined as the implementation of features without related requirements, and it has been considered a critical RE problem, particularly in plan-driven and medium-sized organizations.[41]
ii. Smart contracts. EX7 advocated the use of smart contracts to record software requirements specifications and to assess automatically if the requirements are implemented successfully by the teams. Smart contracts can also be used to assess compliance with service-learning agreements (SLA) automatically. As EX8 stated: "In banking, we had SLA 99.5, so you have records of what is your uptime, downtime, time of recovery. But when you show in the reports, there is a main gap here, people lie on that report. I think if you really want to evaluate if people are doing what they say, you can track this on blockchain".
iii. Web-based application. EX3 suggested a web-based application instead of a desktop application, as it offers better accessibility to end-users.
iv. Configuration options. The proposed prototype consists of only four roles: requirements manager, developer, tester, and customer, for simplicity. EX2 outlined that different companies use different naming for different roles. Thus, the expert suggested a feature that allows

20477481, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2707 by Readcube (Labtiva Inc.), Wiley Online Library on [11/07/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

entering new roles in a company. In addition, EX9 suggested the incorporation of options for different types of software processes, for example, agile and DevOps. Similarly, to improve the usability of the system, EX10 recommended considering other representations of requirements, such as providing a use case template for requirements managers to enter requirements, instead of a text field.

v. Evaluation phase. A final stage named "Evaluate" that focuses on evaluating the return on investment was also suggested by EX8.

vi. Present information to customers in a different format. EX4 raised the concern that the prototype offers the same view to all the users. According to EX4, often customers do not have technical knowledge, hence they do not understand technical artifacts. Customers are more interested in the progress of the project than in how requirements have been implemented (EX4). Therefore, information about the project should be presented to customers in a different format compared with the information presented to developers. In this sense, EX10 suggested limiting the presentation of technical information, such as token ULID to stakeholders who do not have a strong technical background. In addition, the expert recommended the use of simple highlighting techniques that show clearly which implementations of requirements were tested successfully and which test cases failed.

vii. Incentivize software engineers to use the system. Incentivizing software engineers to engage in traceability tasks is not trivial due to the high effort/benefits ratio (EX10). Our experts provided some suggestions regarding this topic. For instance, EX9 recommended the inclusion of gamification features to enhance the participation of software engineers, and EX10 mentioned the inclusion of financial benefits for each traceability link created and social incentives that aim to enhance work-life balance.

5 | DISCUSSION

5.1 Overview of main findings

The paradigm shift from co-located software development to global software development augments the complexity of enabling trustworthy and available RT.[6] The solution proposed by literature[8,42,43] to store and share software artifacts in a distributed multistakeholder environment is a centralized repository in which all artifacts are stored and protected by an access control system. This study challenges the conventional centralized way of storing and sharing artifacts by proposing decentralization through blockchain technology. The use of this technology in the SE field has been advocated in recent studies.[44–47] The findings of this study are summarized in Table 2 and suggest a positive perception of SE experts on using blockchain to enhance traceability and transparency over the SDLC, in particular in regulated domains and in contract-based projects. This is in line with the existing literature that emphasizes the importance of proving the specification, implementation and validation of requirements in safety-critical systems, as demanded by standards,[8] and the importance of traceability as proof-of-correctness to avoid expensive disputes in contract-based projects.[6]

The main concern outlined by the experts is the high effort required for the documentation of requirements and their changes in the proposed prototype. According to the experts, the increased effort may lead to the reluctance of SDLC stakeholders to use the blockchain-enabled system. In fact, this is not surprising, as previous literature revealed that developers perceive traceability as an extra and effort-intensive task that disrupts their workflow,[8,9,12,48] and managers perceive traceability as an expensive activity with not enough evidence of the added value.[8] In addition, our experts claimed that the heavyweight documentation of software artifacts required by our prototype may go against the Agile principle of working software over comprehensive documentation.[49] In fact, missing documentation has been identified as a problematic challenge in a recent survey study that asked 192 SE practitioners to rate the criticality and evaluate the frequency of a set of challenges.[40] While it is true that traceability is often seen as unnecessary and unwanted in agile projects, it is also true that agile projects are increasingly adopted in large-scale, distributed, and often safety-critical projects.[50] The experts emphasized the importance of traceability in such projects, despite the effort and costs associated with its creation and maintenance. Therefore, the type of software development process, economical challenges, and organizational resistance to change are important factors that should be considered in the feasibility analysis phase of the blockchain-enabled framework proposed in this study (See Figure 1). The identification of such factors, along with additional challenges and success factors depicted in Figure 1, may encourage organizations to customize the proposed blockchain framework based on their specific needs.

5.2 Limitations

Although the ultimate goal of qualitative research should be generalizability,[51] the issue of generalizability has been considered as complex by qualitative researchers.[52] In fact, qualitative research tends to provide a contextualized understanding of human opinions, feelings, and experiences[22,52] and consequently is subject to external validity threats. Often, the low number of interviewees indicates limited generalizability of the results.[53] However, this study does not aim to achieve generalizability, but the scope is limited to identifying benefits, challenges, and recommendations on future improvements of the framework. Therefore, 10 experts provided sufficient feedback, due to their broad experience in SE (7 of experts with >10 years of experience). However, a limitation is that six of the eight experts who have both academic and industrial years of

20477481, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2707 by Readcube (Labiva Inc.), Wiley Online Library on [11/07/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 2 Overview of main findings.

Categories		Main findings	Experts
Experts' perceptions		• Innovative, interesting and promising	EX1, EX4, EX7, EX10
		• Feasible and customizable	EX1, EX3
		• Useful for RT and change management	EX2, EX3, EX8, EX9
		• Easy to use	EX3, EX4
Blockchain-based SPI	Business	The prototype enhances traceability.	EX2, EX3, EX9
		Improves trustworthy instant sharing of artifacts and transparency over SDLC	EX1, EX3, EX5, EX7, EX9
		Saves costs and enhances project quality	EX7, EX9
		Main concern is the high documentation effort required which can be perceived anti-agile	EX2, EX5, EX8, EX10
		Four archetypes of potential interested organizations: globally distributed, organizations with contract-based projects, organizations in safety-critical domains and regulators	EX2-EX10
	People	Lack of knowledge and interest of software engineers in learning blockchain	EX5
		Train and upskill employees with blockchain knowledge and skills	EX4, EX7, EX9
	Change	Resistance to change due to high documentation efforts required from software engineers and performance limitations of blockchain	EX2, EX3
		Adopt a top-down approach with policies in place to enable blockchain in a positive way	EX9
Experts' recommendations		Lightweight integration with existing SDLC tools	EX1, EX2, EX5, EX10
		Inclusion of change rationale	EX2
		Inclusion of smart contracts	EX7, EX8
		Web-based application instead of desktop-based	EX3
		Allow for configuration options	EX2, EX9, EX10
		Inclusion of evaluation phase to evaluate return on investment	EX8
		Customize the presented traceability information based on stakeholders' needs	EX4, EX10

Abbreviation: SPI, software process improvement.

experience are not currently working in industry. Moreover, although there are no agreed guidelines regarding sample size in qualitative research,[54] this study considers three factors suggested in Morse[55] to determine the number of experts:

i. *Focused scope of the study* (provide feedback on a specific framework for trustworthy and collaborative traceability management)
ii. *Clear nature of the topic* (blockchain for trustworthy traceability management)
iii. *Quality of the data* (rich and in-depth information obtained in over 100 pages of interview transcripts, 84 codes identified)

Another limitation of qualitative approaches is reproducibility which refers to the ability of a researcher to produce the same results, by using the same raw materials as the prior study.[56] To ensure reproducibility, the research methodology is explained thoroughly in Section 3, and a sample of the analysis process can be accessed in an online repository.[27] Finally, it is worthy to mention that the experts' feedback on the framework is limited, because some aspects of the framework, such as blockchain platform selection, may have been overlooked.

6 | CONCLUSION AND FUTURE WORK

This study proposes and provides feedback on a blockchain-enabled framework and prototype to keep track of software artifacts created and/or changed by distributed stakeholders throughout the SDLC. The feedback has been acquired by conducting semistructured interviews with SE experts. The experts were selected according to their extensive and diverse professional and/or academic experience in SE. Interviews were recorded, transcribed, and analyzed by using the content analysis technique.

The results indicated the emergence of five categories, which were then grouped into the following three main categories: experts' perceptions, blockchain-based SPI, and experts' recommendations. The results suggested an overall positive perception of experts regarding the blockchain-enabled RT framework and prototype. According to the experts, the framework can be useful for four archetypes of software and

20477481, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2707 by Readcube (Labtiva Inc.), Wiley Online Library on [11/07/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

customer organizations: globally distributed organizations, organizations with contract-based projects, customer organizations in regulated domains, and regulators. Moreover, the experts outlined that software engineers may be reluctant to use the proposed system, as high effort is required in documenting all the software artifacts, their metadata and their changes. The resistance of software engineers should be considered and addressed in the feasibility analysis phase of the blockchain-enabled framework, along with other related challenges, such as economical and regulatory challenges.

Experts' recommendations for the improvement of the framework suggested promising future research avenues. One of the avenues advocated by the experts is the use of smart contracts to register requirements specifications and validate their successful implementation automatically. In addition, smart contracts can be used to assess adherence to service-learning agreements, standards and regulations automatically. These recommendations should be taken into account when implementing a practical traceability management framework built on blockchain technology. Our future work will focus on improving the framework based on experts' recommendations and conducting questionnaires based on Pfleeger and Kitchenham's principles of survey research in SE.[57] The subjects of this questionnaire-based study will be SE practitioners who are involved in the development of safety-critical systems. These experts will be presented with a conceptual document, click-dummy, and a fully functional prototype and will be asked to assess the usability and ultimately, their willingness to use the proposed approach.

ACKNOWLEDGMENTS

This work is part of a PhD project, funded by the Østfold University College.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in figshare (https://doi.org/10.6084/m9.figshare.20903656).

ORCID

Mary Sánchez-Gordón https://orcid.org/0000-0002-5102-1122

ENDNOTES

* https://www.youtube.com/watch?v=p1HgheeOvPQ

REFERENCES

1. Mistrík I, Grundy J, van der Hoek A, Whitehead J. Collaborative software engineering: challenges and prospects. In: Mistrík I, Grundy J, Hoek A, Whitehead J, eds. *Collaborative Software Engineering*. Springer; 2010:389-403. doi:10.1007/978-3-642-10294-3_19
2. Babar MA, Lescher C. Global software engineering: identifying challenges is important and providing solutions is even better. *Information and Software Technology*. 2014;56(1):1-5. doi:10.1016/j.infsof.2013.10.002
3. Kiely G, Butler T, Finnegan P. Global virtual teams coordination mechanisms: building theory from research in software development. *Behaviour & Information Technology*. 2022;41(9):1952-1972. doi:10.1080/0144929X.2021.1909141
4. Stol K-J, Fitzgerald B. The ABC of software engineering research. *ACM Trans Softw Eng Methodol*. 2018;27(3):1-51. doi:10.1145/3241743
5. Damian D, Chisan J. An empirical study of the complex relationships between requirements engineering processes and other processes that Lead to payoffs in productivity, quality, and risk management. *IEEE Trans Softw Eng*. 2006;32(7):433-453. doi:10.1109/TSE.2006.61
6. Rempel P, Mäder P, Kuschke T, Philippow I. Requirements traceability across organizational boundaries - a survey and taxonomy. In: Doerr J, Opdahl AL, eds. *Requirements Engineering: Foundation for Software Quality*. Springer; 2013:125-140. doi:10.1007/978-3-642-37422-7_10
7. Spanoudakis G, Zisman A. Software traceability: a roadmap. In: *Handbook of Software Engineering and Knowledge Engineering*. Vol.3. Recent Advances. World Scientific; 2005:395-428. doi:10.1142/9789812775245_0014
8. Maro S, Steghöfer J-P, Staron M. Software traceability in the automotive domain: challenges and solutions. *Journal of Systems and Software*. 2018;141:85-110. doi:10.1016/j.jss.2018.03.060
9. Wohlrab R, Knauss E, Steghöfer J-P, Maro S, Anjorin A, Pelliccione P. Collaborative traceability management: a multiple case study from the perspectives of organization, process, and culture. *Requirements Eng*. 2018;25(1):21-45. doi:10.1007/s00766-018-0306-1
10. Mäder P, Egyed A. Do developers benefit from requirements traceability when evolving and maintaining a software system? *Empir Software Eng*. 2015;20(2):413-441. doi:10.1007/s10664-014-9314-z
11. Tian F, Wang T, Liang P, Wang C, Khan AA, Babar MA. The impact of traceability on software maintenance and evolution: a mapping study. *Journal of Software: Evolution and Process*. 2021;33(10):e2374. doi:10.1002/smr.2374
12. Demi S, Sanchez-Gordon M, Colomo-Palacios R. What have we learnt from the challenges of (semi-) automated requirements traceability? A discussion on blockchain applicability. *IET Software*. 2021;15(6):391-411. doi:10.1049/sfw2.12035
13. Demi S. Blockchain-oriented requirements engineering: a framework. In: 2020 IEEE 28th International Requirements Engineering Conference (RE). 2020;428-433. doi:10.1109/RE48521.2020.00063
14. Demi S, Sánchez-Gordón M, Colomo-Palacios R. A blockchain-enabled framework for requirements traceability. In: Yilmaz M, Clarke P, Messnarz R, Reiner M, eds. *Systems, Software and Services Process Improvement*. Springer International Publishing; 2021:3-13. doi:10.1007/978-3-030-85521-5_1
15. Belotti M, Božić N, Pujolle G, Secci S. A Vademecum on Blockchain technologies: when, which, and how. *IEEE Communications Surveys Tutorials*. 2019;21(4):3796-3838. doi:10.1109/COMST.2019.2928178

20477481, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2707 by Readcube (Labtiva Inc.), Wiley Online Library on [11/07/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

16. Babich V, Hilary G. Blockchain and other distributed ledger technologies in operations. Foundations and trends® in technology. *Information and Operations Management*. 2019;12(2-3):152-172. doi:10.1561/0200000084
17. Sunyaev A. Distributed ledger technology. In: *Internet Computing*. Springer; 2020:265-299. doi:10.1007/978-3-030-34957-8_9
18. Swan M. *Blockchain: Blueprint for a New Economy*. O'Reilly Media, Inc; 2015.
19. Zheng Z, Xie S, Dai H-N, Chen X, Wang H. Blockchain challenges and opportunities: a survey. *International Journal of Web and Grid Services*. 2018; 14(4):352-375. doi:10.5555/3292946.3292948
20. Di Pierro M. What is the blockchain? *Computing in Science & Engineering*. 2017;19(5):92-95. doi:10.1109/MCSE.2017.3421554
21. Demi S, Colomo-Palacios R, Sánchez-Gordón M, Velasco C, Cano R. A neural blockchain for requirements traceability: BC4RT prototype. In: Yilmaz M, Clarke P, Messnarz R, Wöran B, eds. *Systems, Software and Services Process Improvement*. Springer International Publishing; 2022:45-59. doi:10.1007/978-3-031-15559-8_4
22. Demi S, Sánchez-Gordón M, Kristiansen M. Blockchain for requirements traceability: a qualitative approach. *Journal of Software: Evolution and Process*. 2022;35(9):e2493. doi:10.1002/smr.2493
23. Cleland-Huang J, Berenbach B, Clark S, Settimi R, Romanova E. Best practices for automated traceability. *Computer*. 2007;40(6):27-35. doi:10.1109/MC.2007.195
24. Mader P, Gotel O, Philippow I. 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. 2009;21-25. doi:10.1109/TEFSE.2009.5069578
25. Arca. In: ByEvolution Creative Factory. https://byevolution.com/es/arca-demo-2/. Accessed 8 Jun 2022
26. Oates BJ, Griffiths M, McLean R. *Researching Information Systems and Computing*. SAGE; 2022.
27. Demi S, Sánchez-Gordón M, Colomo-Palacios R. Trustworthy and collaborative requirements traceability: Experts' feedback on a Blockchain-enabled framework. Figshare. *Dataset*. 2022. doi:10.6084/m9.figshare.20903656.v3
28. Lenarduzzi V, Dieste O, Fucci D, Vegas S. Towards a methodology for participant selection in software engineering experiments: a vision of the future. In: *Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. Association for Computing Machinery; 2021:1-6. doi:10.1145/3475716.3484273
29. Amir B, Ralph P. Poster: there is no random sampling in software engineering research. In: 2018 IEEE/ACM 40th international conference on software engineering: companion (ICSE-companion). Pp 344–345. 2018. doi:10.1145/3183440.3195001
30. Krippendorff K. *Content Analysis: An Introduction to Its Methodology*. SAGE Publications, Inc.; 2019. 2455 Teller Road, Thousand Oaks California 91320
31. Harwood TG, Garry T. An overview of content analysis. *The Marketing Review*. 2003;3(4):479-498. doi:10.1362/146934703771910080
32. DeFranco JF, Laplante PA. A content analysis process for qualitative software engineering research. *Innovations Syst Softw Eng*. 2017;13(2-3): 129-141. doi:10.1007/s11334-017-0287-0
33. Elo S, Kyngäs H. The qualitative content analysis process. *J Adv Nurs*. 2008;62(1):107-115. doi:10.1111/j.1365-2648.2007.04569.x
34. Graneheim UH, Lundman B. Qualitative content analysis in nursing research: concepts, procedures and measures to achieve trustworthiness. *Nurse Educ Today*. 2004;24(2):105-112. doi:10.1016/j.nedt.2003.10.001
35. Bengtsson M. How to plan and perform a qualitative study using content analysis. *NursingPlus Open*. 2016;2:8-14. doi:10.1016/j.npls.2016.01.001
36. Polit DF, Beck CT. *Essentials of Nursing Research: Methods, Appraisal, and Utilization*. Lippincott Williams & Wilkins; 2006.
37. Georgiadou E, Siakas K, Berki E, Estdale J, Rahanu H, Ross M. The sociocultural dimension of the software process improvement manifesto: pilot validation by experts. *Journal of Software: Evolution and Process*. 2020;32(11):e2304. doi:10.1002/smr.2304
38. Sánchez-Gordón M, Sanchez-Gordon S, Colomo-Palacios R. Vote item: is "compassionate software development" a topic worth researching? In: 2022 IEEE/ACM 15th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE). 2022;107-108. doi:10.1145/3528579.3529176
39. Matthies C, Sánchez-Gordón M, Jørgensen J, Prechelt L. Communication is a scarce resource!" A Summary of CHASE'22 Conference Discussions. 2022. doi:10.48550/arXiv.2207.00054
40. Hoffmann M, Mendez D, Fagerholm F, Luckhardt A. The human side of software engineering teams: an investigation of contemporary challenges. *IEEE Transactions on Software Engineering*. 2022;1–1(1):211-225. doi:10.1109/TSE.2022.3148539
41. Fernández DM, Wagner S, Kalinowski M, et al. Naming the pain in requirements engineering. *Empir Software Eng*. 2017;22(5):2298-2338. doi:10.1007/s10664-016-9451-7
42. Gotel OCZ, Finkelstein CW. *An Analysis of the Requirements Traceability Problem*. Proceedings of IEEE International Conference on Requirements Engineering; 1994:94-101. doi:10.1109/ICRE.1994.292398
43. Regan G, McCaffery F, McDaid K, Flood D. The barriers to traceability and their potential solutions: towards a reference framework. In: 2012 38th Euromicro conference on software engineering and advanced applications. Pp 319–322. 2012. doi:10.1109/SEAA.2012.80
44. Colomo-Palacios R. Cross fertilization in software engineering. In: Yilmaz M, Niemann J, Clarke P, Messnarz R, eds. *Systems, Software and Services Process Improvement*. Springer International Publishing; 2020:3-13. doi:10.1007/978-3-030-56441-4_1
45. Demi S, Colomo-Palacios R, Sánchez-Gordón M. Software engineering applications enabled by Blockchain technology: a systematic mapping study. *Applied Sciences*. 2021;11(7):2960. doi:10.3390/app11072960
46. Marchesi M. Why blockchain is important for software developers, and why software engineering is important for blockchain software (keynote). In: 2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE). 1–12018. doi:10.1109/IWBOSE.2018.8327564
47. Farooq MS, Ahmed M, Emran M. A survey on Blockchain acquainted software requirements engineering: model, opportunities, challenges, and future directions. *IEEE Access*. 2022;10:48193-48228. doi:10.1109/ACCESS.2022.3171408
48. Winkler S, von Pilgrim J. A survey of traceability in requirements engineering and model-driven development. *Softw Syst Model*. 2010;9(4):529-565. doi:10.1007/s10270-009-0145-0
49. Fowler M, Highsmith J. The agile manifesto software development. 2001;9:28-35.
50. Cleland-Huang J. Traceability in agile projects. In: Cleland-Huang J, Gotel O, Zisman A, eds. *Software and Systems Traceability*. Springer; 2012:265-275. doi:10.1007/978-1-4471-2239-5_12
51. Ayres L, Kavanaugh K, Knafl KA. Within-case and across-case approaches to qualitative data analysis. *Qual Health Res*. 2003;13(6):871-883. doi:10.1177/1049732303013006008

20477481, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2707 by Readcube (Labtiva Inc.), Wiley Online Library on [11/07/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

52. El Hussein M, Hirst S, Salyers V, Osuji J. Using grounded theory as a method of inquiry: advantages and disadvantages. *Qual Rep*. 2014;19:1-15. doi:10.46743/2160-3715/2014.1209
53. Boyce C, Neale P. *Conducting In-Depth Interviews: A Guide for Designing and Conducting In-Depth Interviews for Evaluation Input*. Pathfinder international Watertown; 2006.
54. Coughlan J, Lycett M, Macredie RD. Communication issues in requirements elicitation: a content analysis of stakeholder experiences. *Information and Software Technology*. 2003;45(8):525-537. doi:10.1016/S0950-5849(03)00032-6
55. Morse JM. Determining sample size qualitative health research. 2000;10:3-5. doi:10.1177/104973200129118183
56. Goodman SN, Fanelli D, Ioannidis JP. What does research reproducibility mean? *Sci Transl Med*. 2016;8(341):1-6. doi:10.1126/scitranslmed.aaf5027
57. Kitchenham BA, Pfleeger SL. Principles of survey research part 1: turning lemons into lemonade. *ACM SIGSOFT Software Engineering Notes*. 2001; 26(6):16-18. doi:10.1145/505532.505535

How to cite this article: Demi S, Sánchez-Gordón M, Kristiansen M, Larrucea X. Trustworthy and collaborative traceability management: Experts' feedback on a blockchain-enabled framework. *J Softw Evol Proc*. 2024;e2707. https://doi.org/10.1002/smr.2707

APPENDIX A

TABLE A1 Sample of the coding process.

Meaning unit	Condensed meaning unit	Code	Subcategory	Category
"It could be too much details like most of the time as you're working in the same field, this kind of high complex documentation is one of the things you need to avoid at some stage. So, you are creating enormous amounts of documentation or just processing work …"	Complex and enormous documentation is being created. High amount of processing work (overhead).	Documentation perceived as overhead	Business	Blockchain-based software process improvement
"in my experience, there are software engineers …, who are used to use technologies which they are familiar with, and they are quite reluctant to embark on new technologies. That's a little bit contradictory to their capacity to change, because they would understand the technology."	Software engineers are reluctant to embark on new technologies.	Software engineers' resistance to change	Change	Blockchain-based software process improvement
"One think that is missing is how to evaluate the return on investment of this technology. I think it is missing a sixth step «Evaluate», because you have the prototype, but it is important to know how much it is going to cost me because organizations are very concerned about that."	Missing final step: evaluate return on investment.	Add a final phase «Evaluate»	Experts' recommendations	Experts' recommendations

20477481, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/smr.2707 by Readcube (Labtiva Inc.), Wiley Online Library on [11/07/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License