# A Review of Secondary Study on the Verification and Validation of Blockchain Applications: Preliminary Results

Mitja Gradišnik[1], Tina Beranič[1] and Muhamed Turkanović[1]

[1]Faculty of Electrical Engineering and Computer Science - University of Maribor, Koroška cesta 46, Maribor, Slovenia

**Abstract**

In recent years, the development of blockchain technologies, especially programmable smart contracts, has experienced an exceptional growth. This growth is mainly due to the potential these technologies show in various industrial fields. Due to the recognized numerous advantages of blockchain for business environments, there arises a need for their introduction into software solutions, which must be built with high quality to be useful for users. A growing body of literature is found addressing validation and verification techniques. In this paper we conduct review of secondary studies. We conducted an systematic literature review of secondary studies to gain a more refined understanding of good practices for quality assurance, from which conclusions could improve the development process of blockchain-based applications. The systematic search yielded 377 studies of which 37 are selected for further analysis. The literature review revealed that, in quality assurance, formal verification techniques and static analysis are important alongside testing. Due to the immutability of smart contracts, it is recommended to use techniques complementarily. The analysis of individual internal quality aspects revealed that most attention in research field was paid to security, as this is the key attribute that determines whether the implementation of blockchain-based software solutions achieves its intended goals.

**Keywords**

Blockchain-based apps, smart contracts, verification, validation, testing, security

## 1. Introduction

In recent years, blockchain technologies have attracted significant attention from academia and industry, as they can fundamentally transform how businesses operate [1, 2]. Although blockchain technologies initially emerged in the financial sector, their transformative potential has been recognised across various domains beyond cryptocurrencies. In addition to their foundational role in the financial industry, including applications in decentralised finance and lending, blockchain technologies are increasingly being leveraged in areas such as supply chains and logistics, healthcare, and governance [1, 3, 4, 5]. Software developers in these domains are increasingly adopting blockchain technologies because of their fundamental properties, such as decentralisation, immutability, and transparency, which collectively enhance security, ensure data integrity, and support automated processes [1, 6, 7].

One of fundamental characteristic of blockchain is decentralisation. Decentralization creates an environment in which all participants equally contribute to creating and managing records and collectively hold ownership of them. In blockchain networks, each participant keeps a copy of the records [1]. Complete data transparency is inherently achieved because all participants maintain a complete copy of the records. Effective data sharing among participants relies on the immutability of all records, making it a core feature of these systems [8]. Immutability means that data can only be written to and read from the blockchain. Unlike traditional databases, operations such as modifying or deleting data are not supported. Decentralisation, immutability, and transparency are the cornerstones for establishing trust among participants in blockchain systems, eliminating the need for a trusted third party [9]. These properties ensure that no single party controls the data, that records cannot be altered, and that all transactions are visible and verifiable by all stakeholders. The elimination of the need for a central authority paves the way for innovation in electronic commerce and enables more efficient data exchange between organisations.

*SQAMIA 2025: Workshop on Software Quality, Analysis, Monitoring, Improvement, and Applications, September 10–12, 2025, Maribor, Slovenia*

mitja.gradisnik@um.si (M. Gradišnik); tina.beranic@um.si (T. Beranič); muhamed.turkanovic@um.si (M. Turkanović)

Blockchain technologies would have limited practical value without their programmability, which enables the development of custom applications and the integration of solutions into existing information systems [5]. The key component enabling blockchain programmability is the smart contracts. The rapid expansion and widespread adoption of blockchain technology have positioned smart contracts as a fundamental component of secure and automated digital transactions. Smart contracts, often described as self-executing agreements, run directly on blockchain networks, eliminating the need for intermediaries and enhancing transparency [6].

On one hand, these features offer various advantages to businesses. On the other hand, they increase the complexity of verifying the correctness of applications that incorporate them, particularly the correctness, reliability, security, and expected behaviour of applications [8]. For the successful integration of blockchain technologies into enterprise software solutions, it is essential to manage and effectively verify the quality of the developed applications throughout the development process. Given the relative novelty of these technologies, there is a lack of established guidelines for implementing an effective quality assurance process. Due to the specific characteristics of blockchain systems, generic software engineering approaches are often inadequate, and quality assurance processes must be tailored accordingly [8]. Among these characteristics, immutability is a fundamental property that significantly impacts the design, development, and verification of blockchain-based applications. Identifying an effective quality evaluation process is the central focus of this research.

## 2. Background

### 2.1. The architecture of decentralised applications

A central component that enables the programmability of blockchains is undoubtedly smart contracts. Smart contracts typically do not function as standalone applications, but they serve as the core component of decentralised applications (DApps). They contain solely the program logic, implementing both the business logic and the access of smart contracts to data [10]. Decentralised applications are true applications in the complete sense of the word, as they encapsulate smart contracts and expose their functionalities externally, usually through a graphical user interface or an API. Decentralised applications consist of two main components: (1) a front-end, usually implemented as a web application using HTML, CSS, and JavaScript, and (2) a back-end, which includes smart contracts typically written in Solidity or other related languages such as Vyper, or Rust [11]. It is important to emphasise that the architecture presented serves as a generic reference model for decentralised applications and that, in enterprise settings, data are rarely stored exclusively on the blockchain. Instead, organisations typically adopt hybrid storage strategies that combine blockchain storage with additional databases, such as relational or document-oriented databases, to manage and persist data more efficiently.

On the outside, decentralised applications are somewhat like web applications [11]. However, their internal architecture is significantly more complex. The front-end with the user interface and the back-end with smart contracts are separate and independent components communicating through message exchanges. While its deployment location does not restrict the front-end of a decentralised application, smart contracts must run within virtual machines on the nodes of the blockchain network.

Smart contracts are executed in the isolated environment of blockchain networks. As the core component enabling programmability, they inherit immutability, a fundamental property of blockchain technology [12]. In practice, immutability means that smart contracts cannot be easily modified or upgraded once deployed. However, the ability to change, upgrade, and maintain software is one of the fundamental attributes of internal software quality. As such, the immutability of smart contracts poses a significant challenge to established and well-practised approaches in software development, including software quality assessment [13].

### 2.2. Validation and verification

Verification and validation are core processes for ensuring quality, both using testing for achieving set quality goals [14]. The aim of the processes is to support and assist in building quality into the system during the product life cycle [15]. While validation focuses on checking if the developed item corresponds to stakeholders' needs, verification discovers if the item is developed in a proper way, following specifications, specified requirements, and other documents [16].

When employing static and dynamic testing approaches, validation and verification is supported [14]. Static testing evaluate test item without execution, while dynamic testing involve exciting source code for the testing purposes [14]. Static testing range from reviews, model verification and static analysis, while, on the other hand, dynamic testing includes specification-, structured-, and experience-based approaches [14].

## 3. Related work

This research presents a review of literature reviews in the quality assessment of applications based on blockchain technologies. As part of the review, we systematically analysed and synthesised existing secondary research that addresses such applications' validation, verification, or testing. Following a preliminary literature review, we did not find any tertiary literature reviews that directly address the field of quality assessment of blockchain applications. However, the broader literature review revealed several related tertiary studies that generally address validation and verification in software engineering. We identified two related tertiary studies briefly reviewed below due to their relevance to the software product quality assessment field.

Garousi and Mäntylä [17], in their tertiary study, provide a comprehensive overview of the state of accumulated knowledge on software testing during the period from 1994 to 2015 when the study was conducted. After the conducted review, the study pool included 101 secondary studies. Their study aims to systematically map the secondary studies in the field of testing. The research can serve as a summarising index of relevant testing information that supports evidence-based decision-making in any given area of software engineering. The research offers insights into the most frequently addressed testing methods (e.g. model-based approach, regression testing) and software products that are of most significant interest within the testing domain.

In their study, Tran et al. [18] focus on the assessing of the quality of testing artifacts. The main objective of the study is to develop a comprehensive model for capturing the factors of test case quality, which are relevant for various perspectives. As part of their literature review, they identified 49 relevant secondary studies published between 2008 and 2019. Based on a review of secondary literature, the authors present the factors that describe the different contexts in which the quality of test cases is studied. The authors also provide a comprehensive model for test case quality, which defines the quality attributes of their measurements, all based on existing research and the international standard ISO/IEC 25010:2011 [19].

## 4. Research method

In designing this review of literature reviews, we follow the methodology for conducting a standard systematic literature review, following the example of related research in software engineering [17, 20]. The study clearly defines the research questions we aim to answer. It outlines the search strategy for identifying secondary studies and explicitly states the inclusion and exclusion criteria. The study describes the procedure for selecting relevant secondary studies and the data extraction process. The continuation of the research focuses on analysing the data collected from secondary studies and synthesising knowledge in the field of validation and verification of blockchain applications. This synthesis provides answers to the previously defined research questions.

## 4.1. Goal of the study

Since blockchain technologies are relatively novel compared to other software products, the quality assurance process for such solutions is not yet as well-established as it is in developing conventional applications, e.g. web or mobile applications. Due to the many specific characteristics of blockchain technologies, software product validation and verification processes must be adopted and adapted accordingly. An efficient quality assurance framework is essential for successfully executing IT projects involving blockchain technologies.

This study is based on reviewing secondary sources, mainly systematic literature reviews and systematic mapping studies, aiming to form a comprehensive picture of the risks and available methods in ensuring the quality of blockchain applications. There are several advantages in using secondary literature sources instead of primary sources. Firstly, secondary sources summarize and synthesize knowledge from a larger number of primary studies, which provides more condensed information on the topic under consideration. The second reason is the systematic nature of the reviews conducted and the critical evaluation of the sources examined. Topics that recur more frequently can be considered more important in the field. Lastly, it is necessary to consider the breadth of the field of quality research. Although blockchain application development is relatively young, it would be difficult to fully address primary studies given the wide range of quality attributes considered, which the quality assurance process requires.

## 4.2. Research questions

Software quality assurance can be approached from two perspectives: (1) the process perspective (top-down perspective), which focuses on the validation and verification workflow along with appropriate techniques and methods, and (2) the product-oriented perspective (bottom-up), which addresses specific quality challenges in the software itself and seeks effective solutions. A common ground between both perspectives lies in the selection of approaches and methods to ensure the quality of blockchain-based applications. Therefore, in this study, we examine the field of software verification and validation from both the process-oriented and the challenge-driven points of view. In line with this approach, we have formulated the following research questions:

- **RQ1:** How can we validate and verify blockchain applications?
- **RQ2:** Which quality attributes of blockchain applications are directly addressed in secondary studies?

Research question **RQ1** provides approaches and state-of-the-art techniques that can be used during the development process of blockchain applications to verify and validate quality aspects of blockchain applications. Within the scope of research question **RQ2**, we identify challenges in quality attributes of blockchain-based applications. The research question identified aspects of internal quality of blockchain applications that require special attention during development process. The frequency of occurrence of each quality attribute in the secondary sources also suggests the importance of individual quality aspects.

## 4.3. Search process

The search for secondary literature was conducted to identify existing systematic reviews, mapping studies, and survey articles that address the validation and verification of blockchain applications. The following search string was used:

(”smart contract” OR ”blockchain application” OR ”dApps”) AND (”testing” OR ”validation” OR ”verification” OR ”detection”) AND (”review” OR ”mapping” OR ”survey” OR ”research direction”).

The stated query is written in a generic form. The actual queries were adapted to the specific online academic libraries used. This query was designed to capture various terms related to blockchain-based

applications, including (1) smart contracts and decentralised applications; (2) quality assurance activities, such as testing, validation, verification, and detection; and (3) typical descriptors of secondary studies, such as reviews, mapping studies, and surveys. The literature search was conducted using the following online academic libraries:

- ScienceDirect (https://www.sciencedirect.com/),
- IEEE Xplore (https://ieeexplore.ieee.org/),
- SpringerLink (https://link.springer.com/),
- Scopus (https://www.elsevier.com/products/scopus),
- ACM Digital Library (https://dl.acm.org/).

These databases were selected based on their extensive coverage of peer-reviewed publications in the fields of computer science, software engineering, and emerging technologies, including blockchain. By combining multiple databases, the search process aimed to maximise the breadth of the collected literature and reduce the risk of overlooking relevant secondary studies.

## 4.4. Inclusion and exclusion criteria

The number of retrieved papers by online academic libraries is reduced by specifying a strict number of inclusion and exclusion criteria. In the study, only peer-reviewed papers from journal and conferences are included. Given the relative youth of the blockchain research field, we did not impose any time restrictions on the search. All sources relevant to the domain were considered. Only English-language papers were included, including surveys, literature reviews and mapping studies addressing the validation, verification, or testing of blockchain-based applications. The complete list of adopted inclusion and exclusion criteria is presented in Table 1.

**Table 1**
Inclusion and exclusion criteria

| | **Inclusion Criterion** | | **Exclusion Criterion** |
|---|---|---|---|
| I1 | Papers must be written in English | E1 | Papers fully or partially written in languages other than English |
| I2 | Literature reviews, mapping studies and surveys | E2 | Primary empirical studies, technical reports, books, and book chapters |
| I3 | Focus on validation, verification, or testing of blockchain applications | E3 | Do not address these quality aspects or are not related to blockchain |
| I4 | Full text available | E4 | Papers without full text available |
| I5 | Reseach area of computer science | E5 | Other areas, such as economics, mathematics |

## 4.5. Validity threats

In the following section, we discuss potential threats to the validity of this study.

The first possible threat to the validity of this research is that it may miss relevant secondary studies in the field. We mitigated this risk by careful development and evaluation of our search strings. A related validity threat is caused by our decision to exclude grey literature from the study. The study, therefore, represents an exclusively academic perspective on the topic under consideration, excluding industry reports. However, since we reviewed secondary and not primary studies, the risk of excluding relevant but not peer-reviewed material is low. Since our data extraction in this study is based on secondary studies, relevant information about verifying and validating blockchain applications available in primary studies may no longer be available in secondary studies. This thread is inherent to any review based on secondary studies. We accept this threat as a trade-off for the breadth of the research domain that can be covered through secondary studies.

# 5. Results of the study

This section presents and elaborates on the study's results, answering the two research questions introduced in Section 4.2.

## 5.1. Overview of selected studies

The literature search was carried out in June 2025, resulting in 37 unique papers published since 2019. The formulated search query was executed across a selection of online scientific databases. Table 2 presents the number of papers retrieved from each individual database.

**Table 2**
Number of studies returned by each library

| Online Library | Search results |
|---|---|
| ScienceDirect | 13 |
| IEEE Xplorer | 120 |
| SpringerLink | 55 |
| Scopus | 182 |
| ACM Digital library | 7 |
| Total | 377 |

The initial set of 377 papers retrieved from various online academic libraries was compiled and reviewed. After screening the titles and abstracts for relevance, 73 papers were selected for further consideration. Applying the inclusion and exclusion criteria and removing duplicates reduced the set to 59 unique papers. Finally, after a full-text review, 37 papers were deemed eligible and included in the study. Table 3 provides a list of the selected literature, along with details on its id, publication year, type, and source. All identified studies in the table are marked with a unique identifier in the format *Sx*, where *x* represents the sequential number of the study. This identification method was introduced to enable simpler and more concise referencing throughout the text of the paper.

Figure 5.1 shows the distribution of selected studies according to their publication year and source. The relative novelty of the research field focusing on the quality of blockchain-based applications is evident from the distribution of publications over the years. The earliest publication dates back to 2019, while the highest number of annual publications was recorded in 2024, with 11 papers across the included online academic libraries. This trend suggests that the field remains of ongoing interest to researchers. The Figure also shows that the maximum number of publications come from IEEE Xplore. Only one from ACM Digital Libraries meets our inclusion criteria.

## 5.2. Process of verification and validation - RQ1

The primary objective of the validation and verification process is to evaluate both the functional and non-functional aspects of blockchain application quality [8]. Among the 37 secondary studies analysed, 15 address the validation and verification process in techniques in the context of blockchain applications, representing 41% of the secondary studies included in our review. Studies S3, S9, S11, S13-17, S20-21, S24, S26, S29, S32, and S36 are particularly relevant to the response to the research question RV1. Verification and validation, as an overarching process comprehensively addressed within the development lifecycle of blockchain applications, is specifically discussed in three studies (S3, S9, and S26). Paper S14 addresses the security assurance of blockchain solutions within the development lifecycle. Based on the reviewed studies, verification and validation approaches can be grouped into two main branches: static analysis, which focuses on analysing the source code of smart contracts, and dynamic analysis, which examines the behavior of the program during execution [8]. Static code analysis is used for examining data and control flow, performing taint analysis, and identifying program code patterns that represent vulnerability. Pattern analysis is primarily applied in security assessments and in evaluating gas consumption efficiency.

**Table 3**
List of selected studies

| ID | Ref | Year | Publication Type | Online Library |
|---|---|---|---|---|
| S1 | [21] | 2024 | journal | ScienceDirect |
| S2 | [22] | 2024 | journal | ScienceDirect |
| S3 | [8] | 2022 | journal | ScienceDirect |
| S4 | [6] | 2025 | journal | ScienceDirect |
| S5 | [3] | 2025 | journal | ScienceDirect |
| S6 | [23] | 2024 | journal | ScienceDirect |
| S7 | [10] | 2024 | journal | ScienceDirect |
| S8 | [13] | 2023 | journal | ScienceDirect |
| S9 | [24] | 2020 | journal | ScienceDirect |
| S10 | [9] | 2019 | journal | IEEE |
| S11 | [25] | 2023 | conference | IEEE |
| S12 | [26] | 2020 | conference | IEEE |
| S13 | [27] | 2022 | conference | IEEE |
| S14 | [12] | 2019 | journal | IEEE |
| S15 | [28] | 2019 | conference | IEEE |
| S16 | [29] | 2020 | conference | IEEE |
| S17 | [30] | 2021 | conference | IEEE |
| S18 | [31] | 2022 | conference | IEEE |
| S19 | [32] | 2024 | conference | IEEE |
| S20 | [4] | 2020 | conference | IEEE |
| S21 | [33] | 2025 | conference | IEEE |
| S22 | [34] | 2022 | journal | IEEE |
| S23 | [35] | 2019 | conference | IEEE |
| S24 | [36] | 2022 | conference | IEEE |
| S25 | [1] | 2024 | conference | IEEE |
| S26 | [37] | 2020 | journal | IEEE |
| S27 | [38] | 2024 | journal | IEEE |
| S28 | [39] | 2024 | journal | IEEE |
| S29 | [40] | 2024 | journal | SpringerLink |
| S30 | [5] | 2020 | journal | SpringerLink |
| S31 | [41] | 2025 | journal | Scopus |
| S32 | [11] | 2025 | journal | Scopus |
| S33 | [42] | 2024 | journal | Scopus |
| S34 | [43] | 2023 | journal | Scopus |
| S35 | [2] | 2022 | journal | Scopus |
| S36 | [44] | 2020 | journal | Scopus |
| S37 | [45] | 2023 | journal | ACM |

Within dynamic analysis, we can further distinguish two major categories of techniques: testing and performance analysis. Performance analysis is less thoroughly discussed in the reviewed studies. Namely, only study S3 addresses this topic in detail, while sources S24 and S32 merely mention the techniques and offer only a brief description.

Testing represents one of the most prominent groups of verification and validations techniques. The majority of the studies addressing the validation and verification process focus on testing. Our review identified 7 sources that discuss testing in the context of blockchain applications. Of these 7 studies, 6 studies (S13, S17, S20, S24, S29, and S32) examine testing in general and present a range of testing techniques used in primary studies. Among the secondary sources, the most frequently highlighted techniques are fuzz testing and mutation testing. Fuzz testing is an automated software testing technique based on injecting large amounts of random (including incorrect) input data into a program. Its primary purpose is to trigger crashes or unexpected behaviours, which makes it especially useful for security testing. Another important automated technique is mutation testing, which is not used for directly testing software products, but rather for evaluating the quality of test cases. The

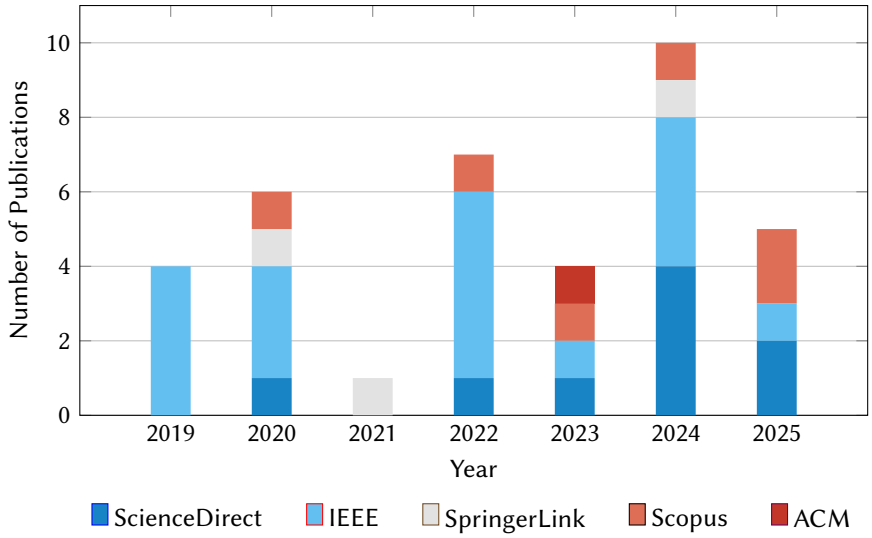


**Figure 1:** Distribution of selected studies by year and digital library

core idea is to introduce small intentional random changes into the program code, called mutants, and assess how effectively test suites detect those changes. Testing from the perspective of model-based development is addressed in more detail by paper S20.

Other testing techniques, mentioned in review studies, are unit testing, functional testing, integration testing, model-based testing, security testing, and performance testing, which further includes stress testing, load testing, and scalability testing. One of the seven identified sources, study S16 focuses specifically on acceptance testing.

Formal verification represents the next major group of techniques. Most formal techniques are based on static analysis. However, some also incorporate dynamic approaches, making it difficult to classify within a single category. A common characteristic of all formal verification techniques is their reliance on formal proof and mathematical modelling to demonstrate the functional or security compliance of a software design [10]. This approach fundamentally differs from testing, which derives quality assurance through empirical evaluation of the system's behavior.

The core techniques of formal verification include theorem proving, model checking, and abstract interpretation. Some of the reviewed studies also consider runtime verification and dynamic symbolic execution as part of the formal verification domain, although these methods combine elements of formalism and program execution. Among the reviewed studies, four focus on techniques of the formal verification, namely S11, S15, S21, and S36. The study S36 highlights verification techniques, with an exclusive focus on their application to security aspects. The overview and classification of verification and validation techniques is shown in Figure 2.

## 5.3. Quality challenges of blockchain applications - RQ2

The analysis of the collected studies revealed that the majority of contributions included in our review adopt a bottom-up approach. Among the 37 secondary studies analysed, 22 focus on the quality-related aspects and attributes of blockchain applications, accounting for 59% of the studies considered in our review. These studies do not target the overall validation end verification process but rather aim to provide solutions to concrete quality-related issues. Among the identified secondary studies that follow the bottom-up approach, some studies that address specific challenges (e.g. vulnerability), the other focus on individual aspects or quality attributes of blockchain applications (e.g. security). For research question RV2, studies S1-2, S4-8, S10, S12, S18-19, S22-23, S25, S27-28, S30-31, S33-35, and S37 are particularly relevant.

The analysis of challenges addressed by the reviewed studies reveals a certain monotony in the field. The primary quality related challenge in blockchain applications is related to vulnerabilities in smart

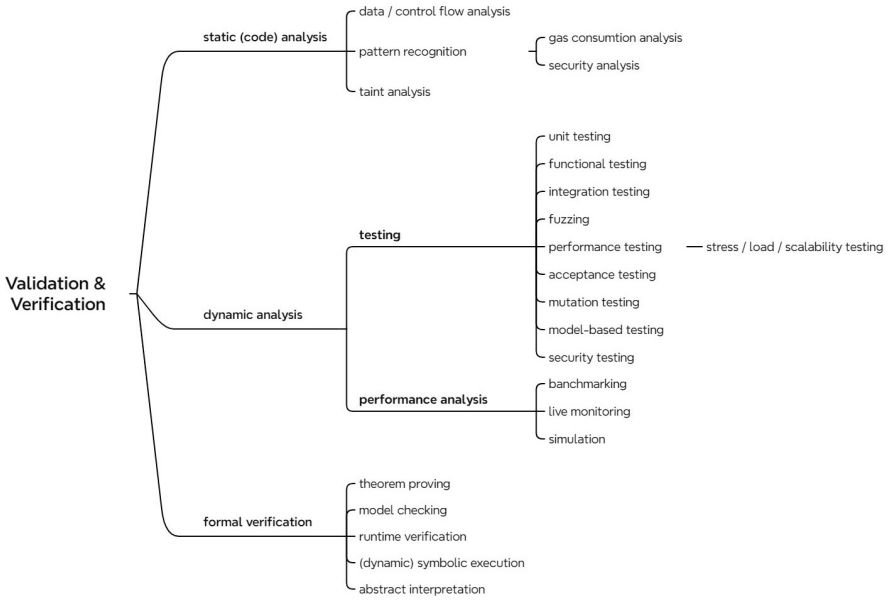


**Figure 2:** Classification of major verification and validation techniques

contracts. Analysis reveals that 15 identified studies (S2, S4, S6-8, S12, S18, S23, S25, S27-28, S33-35, and S37) focus primarily on vulnerability detection and prevention in smart contracts as the central quality assurance issue. The main objective of these secondary studies is to compile a set of known vulnerabilities and provide a list of tools that can be used to detect and mitigate them. Vulnerability detection techniques and tools are mainly based on static code analysis and increasingly leverage machine learning and deep learning approaches (e.g., S2, S33). Some studies also focus on new static analysis techniques, the optimization of established formal verification methods, and the advancement of testing strategies.

The review also identifies 7 studies (S1, S5, S10, S19, S22, and S30-31) that address the broader topic of security in blockchain applications. The primary focus of the papers revolves around security analysis and evaluation. However, it is characteristic of these studies that security is typically framed in terms of vulnerabilities and their mitigation.

## 6. Discussion

The immutability of smart contracts is that limitation of blockchain applications that needs to be adequately addressed during the development process of such applications. Techniques and methods for evaluating the quality of blockchain applications adhere to the principle of deploying smart contracts to networks only when they do not contain bugs, vulnerabilities, and other shortcomings. This would eliminate the need for later corrections to smart contracts.

The literature review shows a wide spectrum of available techniques. The first important group is formal verification. Formal verification techniques employ mathematical models and formal proofs to rigorously demonstrate the functional correctness and security robustness of smart contract designs. The techniques can therefore be used to mathematically prove that smart contracts function according to specifications. Formal methods do, however, have limitations. Their use can be limited if the systems are too complex or it does not have a limited number of states. In such cases, it is sensible to use testing.

In area of testing blockchain applications, fuzz testing and mutation testing are the most frequently highlighted techniques. The fuzz testing automates the generation of a large quantity of random input data, including incorrect inputs, which are passed into smart contracts inputs. This approach creates circumstances in which edge cases of operation are also tested, which, in addition to functional incorrectness, also reveals potential vulnerabilities in the program code. The mutation testing technique

represents a validation of applied testing that evaluates its coverage. Small mutations in the program code must be detected by the test suite, for the testing process to be considered effective. Relevant approaches for the field include techniques based on static analysis of source code. The purpose of these techniques is to detect and eliminate problematic code that could lead to vulnerabilities in smart contracts.

The literature review suggests that the removal of defects and vulnerabilities in blockchain applications is a multi-stage process. No single technique can fully address the quality challenges of blockchain applications. Therefore, they should be used in a complementary manner whenever possible.

Our analysis of the main quality challenges in developing blockchain applications shows that secondary studies focus predominantly on security. As thoroughly documented by Wei et al. [45], security incidents have historically undermined trust in safety of blockchain solutions. The immutability of smart contracts once deployed, combined with the need to ensure security, significantly increases the engineering effort required to develop reliable blockchain applications. The reviewed secondary sources provide a detailed and comprehensive solution of security concerns, offering numerous vulnerability detection tools and methods that can be directly applied in practice by developers of blockchain applications. However, security is only one of several critical quality attributes, and a comprehensive quality assurance approach must also address other aspects beyond security. In the future, it is expected that the academic community will devote similar attention to other aspects of the internal quality of blockchain applications as it currently does to security.

## 7. Conclusions

The development of blockchain applications and their integration into business environments presents a vital engineering challenge for two main reasons. First, compared to conventional web applications, blockchain-based applications are architecturally more complex. They contain smart contracts as a key component. Second, once smart contracts are deployed on a blockchain network, they cannot be easily replaced with upgraded or fixed versions.

The primary objective of this systematic review of secondary sources is to examine aggregated knowledge resources and to identify key specifics in the quality assurance of blockchain applications. Recognizing these specifics is essential for constructing a comprehensive quality assurance process tailored to blockchain-based applications.

A top-down analysis of the secondary literature reveals a wide range of techniques, highlighting formal verification and static analysis as key approaches that complement traditional testing techniques for blockchain applications. In contrast, a bottom-up perspective uncovers a relatively modest coverage of internal quality aspects. Most identified studies focus on the security aspect of smart contracts, while other quality attributes remain unaddressed in the secondary literature.

To develop a comprehensive quality assurance process model for blockchain applications, evaluating the suitability of individual techniques within specific contexts will be necessary. Individual techniques do not represent universal solutions for quality assessment and should be used complementarily. Furthermore, quality assurance techniques for other internal quality attributes of blockchain applications must be explored. Based on the findings of this review, the area of internal quality attributes presents numerous opportunities for future research.

Based on identified secondary sources in the validation and verification of blockchain applications, the article lays the foundation for further research to conduct an in-depth analysis of validation and verification techniques. Future work will focus on identifying novel techniques and adapting existing ones in the domain, as well as exploring the application of composite approaches for validating and verifying blockchain applications. A significant challenge for future studies will be including sources beyond the academic community, such as industry reports and white papers.

## Acknowledgments

The authors acknowledge financial support from the Slovenian Research and Innovation Agency (Research Core Funding No. P2-0057).

## Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT, Gemini, and Grammarly for grammar and spelling checks, paraphrasing, and rewording. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

## References

[1] A. Gurjar, B. R. Chandavarkar, Smart contract vulnerabilities and detection methods: A survey, 2024 15th International Conference on Computing Communication and Networking Technologies, ICCCNT 2024 (2024). doi:10.1109/ICCCNT61001.2024.10724246.

[2] H. Rameder, M. di Angelo, G. Salzer, Review of automated vulnerability analysis of smart contracts on ethereum, Frontiers in Blockchain 5 (2022) 814977. doi:10.3389/FBLOC.2022.814977/PDF.

[3] T. Hu, B. Li, Dynamic information utilization for securing ethereum smart contracts: A literature review, Information and Software Technology 182 (2025) 107719. URL: https://www.sciencedirect.com/science/article/pii/S0950584925000588. doi:10.1016/J.INFSOF.2025.107719.

[4] N. Sánchez-Gómez, J. Torres-Valderrama, J. A. García-García, J. J. Gutiérrez, M. J. Escalona, Model-based software design and testing in blockchain smart contracts: A systematic literature review, IEEE Access 8 (2020) 164556–164569. doi:10.1109/ACCESS.2020.3021502.

[5] Z. Wang, H. Jin, W. Dai, K. K. R. Choo, D. Zou, Ethereum smart contract security research: survey and future research opportunities, Frontiers of Computer Science 15 (2021) 1–18. URL: https://link.springer.com/article/10.1007/s11704-020-9284-9. doi:10.1007/S11704-020-9284-9/METRICS.

[6] N. Hejazi, A. H. Lashkari, A comprehensive survey of smart contracts vulnerability detection tools: Techniques and methodologies, Journal of Network and Computer Applications 237 (2025) 104142. URL: https://www.sciencedirect.com/science/article/pii/S1084804525000396. doi:10.1016/J.JNCA.2025.104142.

[7] M. Krichen, M. Lahami, Q. A. Al-Haija, Formal methods for the verification of smart contracts: A review, Proceedings of the 2022 15th IEEE International Conference on Security of Information and Networks, SIN 2022 (2022). doi:10.1109/SIN56466.2022.9970534.

[8] D. Marijan, C. Lal, Blockchain verification and validation: Techniques, challenges, and research directions, Computer Science Review 45 (2022) 100492. URL: https://www.sciencedirect.com/science/article/pii/S1574013722000314. doi:10.1016/J.COSREV.2022.100492.

[9] J. Liu, Z. Liu, A survey on security verification of blockchain smart contracts, IEEE Access 7 (2019) 77894–77904. doi:10.1109/ACCESS.2019.2921624.

[10] F. R. Vidal, N. Ivaki, N. Laranjeiro, Vulnerability detection techniques for smart contracts: A systematic literature review, Journal of Systems and Software 217 (2024) 112160. URL: https://www.sciencedirect.com/science/article/pii/S016412122400205X. doi:10.1016/J.JSS.2024.112160.

[11] M. Lahami, A. J. Maalej, M. Krichen, A systematic literature review on dynamic testing of blockchain oriented software, Science of Computer Programming 240 (2025) 103211. doi:10.1016/j.scico.2024.103211.

[12] Y. Huang, Y. Bian, R. Li, J. L. Zhao, P. Shi, Smart contract security: A software lifecycle perspective, IEEE Access 7 (2019) 150184–150202. doi:10.1109/ACCESS.2019.2946988.

[13] H. Chu, P. Zhang, H. Dong, Y. Xiao, S. Ji, W. Li, A survey on smart contract vulnerabilities: Data sources, detection and repair, Information and Software Technology 159 (2023) 107221. URL: https://www.sciencedirect.com/science/article/pii/S0950584923000757. doi:10.1016/J.INFSOF.2023.107221.

[14] International Organization for Standardization, International Electrotechnical Commission, ISO/IEC/IEEE International Standard - Software and systems engineering –Software testing –Part 1:General concepts, Technical Report, ISO, 2022. doi:10.1109/IEEESTD.2022.9698145.

[15] IEEE, IEEE Standard for System, Software, and Hardware Verification and Validation, Technical Report, Institute of Electrical and Electronics Engineers, 2017. doi:10.1109/IEEESTD.2017.8055462.

[16] Project Management Institute (PMI), IEEE Draft Guide: Adoption of the Project Management Institute (PMI) Standard: A Guide to the Project Management Body of Knowledge (PMBOK Guide)-2008 (4th edition), Technical Report, Project Management Institute, 2011. doi:10.1109/IEEESTD.2011.5937011.

[17] V. Garousi, M. V. Mäntylä, A systematic literature review of literature reviews in software testing, Information and Software Technology 80 (2016) 195–216. URL: https://www.sciencedirect.com/science/article/pii/S0950584916301446. doi:10.1016/J.INFSOF.2016.09.002.

[18] H. K. V. Tran, M. Unterkalmsteiner, J. Börstler, N. bin Ali, Assessing test artifact quality—a tertiary study, Information and Software Technology 139 (2021) 106620. URL: https://www.sciencedirect.com/science/article/pii/S0950584921000938. doi:10.1016/J.INFSOF.2021.106620.

[19] International Organization for Standardization, International Electrotechnical Commission, Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models, Technical Report ISO/IEC 25010:2011, ISO, Geneva, Switzerland, 2011. URL: https://www.iso.org/standard/35733.html.

[20] B. Kitchenham, S. Charters, Guidelines for performing Systematic Literature Reviews in Software Engineering, Technical Report EBSE-2007-01, Keele University and Durham University, 2007.

[21] G. Wu, H. P. Wang, X. Lai, M. Wang, D. He, S. Chan, A comprehensive survey of smart contract security: State of the art and research directions, Journal of Network and Computer Applications 226 (2024) 103882. URL: https://www.sciencedirect.com/science/article/pii/S1084804524000596. doi:10.1016/J.JNCA.2024.103882.

[22] H. Wu, Y. Peng, Y. He, J. Fan, A review of deep learning-based vulnerability detection tools for ethernet smart contracts, CMES - Computer Modeling in Engineering and Sciences 140 (2024) 77–108. URL: https://www.sciencedirect.com/org/science/article/pii/S1526149224001735. doi:10.32604/CMES.2024.046758.

[23] Y. He, J. Fan, H. Wu, A systematic review and performance evaluation of open-source tools for smart contract vulnerability detection, Computers, Materials and Continua 80 (2024) 995–1032. URL: https://www.sciencedirect.com/org/science/article/pii/S1546221824004892. doi:10.32604/CMC.2024.052887.

[24] M. Almakhour, L. Sliman, A. E. Samhat, A. Mellouk, Verification of smart contracts: A survey, Pervasive and Mobile Computing 67 (2020) 101227. URL: https://www.sciencedirect.com/science/article/pii/S1574119220300821. doi:10.1016/J.PMCJ.2020.101227.

[25] R. B. Fekih, M. Lahami, M. Jmaiel, S. Bradai, Formal verification of smart contracts based on model checking: An overview, Proceedings of the Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE (2023). doi:10.1109/WETICE57085.2023.10477834.

[26] J. Xu, F. Dang, X. Ding, M. Zhou, A survey on vulnerability detection tools of smart contract bytecode, Proceedings of 2020 IEEE 3rd International Conference on Information Systems and Computer Aided Education, ICISCAE 2020 (2020) 94–98. doi:10.1109/ICISCAE51034.2020.9236931.

[27] N. P. Imperius, A. D. Alahmar, Systematic mapping of testing smart contracts for blockchain applications, IEEE Access 10 (2022) 112845–112857. doi:10.1109/ACCESS.2022.3216874.

[28] Y. Murray, D. A. Anisi, Survey of formal verification methods for smart contracts on blockchain, 2019 10th IFIP International Conference on New Technologies, Mobility and Security, NTMS 2019 - Proceedings and Workshop (2019). doi:10.1109/NTMS.2019.8763832.

[29] P. Vilain, J. Mylopoulos, H. A. Jacobsen, A preliminary study on using acceptance tests for representing business requirements of smart contracts, IEEE International Conference on Blockchain and Cryptocurrency, ICBC 2020 (2020). doi:10.1109/ICBC48266.2020.9169480.

[30] R. Sujeetha, C. A. D. Preetha, A literature survey on smart contract testing and analysis for smart contract based blockchain application development, Proceedings - 2nd International Conference on

Smart Electronics and Communication, ICOSEC 2021 (2021) 378–385. doi:10.1109/ICOSEC51865.2021.9591750.

[31] S. S. Kushwaha, S. Joshi, D. Singh, M. Kaur, H. N. Lee, Systematic review of security vulnerabilities in ethereum blockchain smart contract, IEEE Access 10 (2022) 6605–6621. doi:10.1109/ACCESS.2021.3140091.

[32] H. Zhu, L. Yang, L. Wang, V. S. Sheng, A survey on security analysis methods of smart contracts, IEEE Transactions on Services Computing 17 (2024) 4522–4539. URL: https://ieeexplore.ieee.org/document/10683998/. doi:10.1109/TSC.2024.3463394.

[33] R. B. Fekih, M. Lahami, S. Bradai, M. Jmaiel, Formal verification of erc-based smart contracts: A systematic literature review, IEEE Access (2025). doi:10.1109/ACCESS.2025.3527158.

[34] J. Su, J. Liu, Y. Nan, Y. Li, Security evaluation of smart contracts based on code and transaction - a survey, Proceedings of International Conference on Service Science, ICSS 2022-May (2022) 41–48. doi:10.1109/ICSS55994.2022.00016.

[35] M. D. Angelo, G. Salzer, A survey of tools for analyzing ethereum smart contracts, Proceedings - 2019 IEEE International Conference on Decentralized Applications and Infrastructures, DAPPCON 2019 (2019) 69–78. doi:10.1109/DAPPCON.2019.00018.

[36] N. Arsat, N. S. A. A. Bakar, N. Yahya, Testing in blockchain-based systems: A systematic review, 2022 10th International Conference on Cyber and IT Service Management, CITSM 2022 (2022). doi:10.1109/CITSM56380.2022.9935846.

[37] S. Kim, S. Ryu, Analysis of blockchain smart contracts: Techniques and insights, Proceedings - 2020 IEEE Secure Development, SecDev 2020 (2020) 65–73. doi:10.1109/SECDEV45635.2020.00026.

[38] W. Zhao, W. Mi, X. Zhang, The security paradox of smart contracts: Blind spots and prospects of current detection strategies, Proceedings of the 2024 27th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2024 (2024) 1546–1551. doi:10.1109/CSCWD61410.2024.10580546.

[39] Z. A. Khan, A. S. Namin, A survey of vulnerability detection techniques by smart contract tools, IEEE Access 12 (2024) 70870–70910. doi:10.1109/ACCESS.2024.3401623.

[40] A. Elakaş, H. Sözer, I. Şafak, K. Kalkan, A systematic mapping on software testing for blockchains, Cluster Computing 27 (2024) 7111–7126. URL: https://link.springer.com/article/10.1007/s10586-024-04421-7. doi:10.1007/S10586-024-04421-7/FIGURES/6.

[41] U. U. Ibekwe, U. M. Mbanaso, N. A. Nnanna, U. A. Ibrahim, Navigating the smart contract threat landscape: a systematic review, Indonesian Journal of Electrical Engineering and Computer Science 37 (2025) 1209 – 1224. doi:10.11591/ijeecs.v37.i2.pp1209-1224.

[42] R. Kiani, V. S. Sheng, Ethereum smart contract vulnerability detection and machine learning-driven solutions: A systematic literature review, Electronics (Switzerland) 13 (2024) 2295. doi:10.3390/electronics13122295.

[43] F. Jiang, K. Chao, J. Xiao, Q. Liu, K. Gu, J. Wu, Y. Cao, Enhancing smart-contract security through machine learning: A survey of approaches and techniques, Electronics (Switzerland) 12 (2023) 2046. doi:10.3390/electronics12092046.

[44] A. Singh, R. M. Parizi, Q. Zhang, K. K. R. Choo, A. Dehghantanha, Blockchain smart contracts formalization: Approaches and challenges to address vulnerabilities, Computers and Security 88 (2020) 101654. doi:10.1016/j.cose.2019.101654.

[45] Z. Wei, J. Sun, Z. Zhang, X. Zhang, X. Yang, L. Zhu, Survey on quality assurance of smart contracts, ACM Comput. Surv 37 (2023) 43. URL: https://arxiv.org/pdf/2311.00270.