How to Approach Security Testing of Web 3.0 Solutions: A Review of Existing Knowledge

Nika Jeršič[1,*], Muhamed Turkanović[1] and Tina Beranič[1]

[1]University of Maribor, The Faculty of Electrical Engineering and Computer Science, Koroška cesta 46, 2000 Maribor, Slovenia

Abstract

Despite its promising advancements, Web 3.0 solutions still encounter significant challenges, particularly in the realm of software testing and security testing. This paper introduces a literature review of existing research in the fields of Web 3.0 technologies and security testing. We found there is little research on the connection between these fields. Most security research in Web 3.0 is focused on smart contracts and network attacks, like Denial-of-Service (DoS). Our analysis identifies significant trends, attacks and mitigations in overlapping topics within the current body of research. Understanding these challenges is crucial for ensuring the reliability and security of Web 3.0 solutions as they become more integrated into mainstream technology.

Keywords

Security testing, Web 3.0, Blockchain, Quality assurance, Smart contracts

1. Introduction

Security testing in the quality assurance (QA) domain is one of the first steps in ensuring a robust and safe system. The systematic security approaches ensure that products meet or exceed customer expectations and that the implementation satisfies the requirements. It rigorously tests various processes to achieve the desired goals, including design, development, and evaluation of the system [1]. In the rapidly evolving landscape of technology, Web 3.0 stands out as the next major evolution, often referred to as the decentralized web. Unlike its predecessors, Web 1.0 (static web) and Web 2.0 (dynamic and social web), Web 3.0 aims to provide a more autonomous and secure digital environment by leveraging advanced technologies such as blockchain, semantic web, and artificial intelligence [2, 3]. This evolution promises enhanced user privacy, data ownership, and interoperability, making significant shifts towards a user-centric internet.

One of the key promises of Web 3.0 is improved security. Blockchain technology ensures that there is no single point of failure, reducing the risk of centralized attacks. However, not everything is so simple. Decentralized applications (dApps), smart contracts, and other Web 3.0 solutions have their unique security challenges. Smart contracts, for example, can be vulnerable to bugs in the code, which can be exploited by malicious actors. Similarly, decentralized applications can be targeted by attacks that exploit flaws in protocols or implementations.

Ensuring the robustness and security of Web 3.0 applications is crucial, given their complex and decentralized nature, which introduces new challenges in the area of security and reliability, requiring more advanced QA methodologies. Traditional security testing practices may not be sufficient to meet these challenges, and it is imperative to innovate and adapt these strategies to the unique requirements of decentralized systems. Therefore, incorporating rigorous security testing methodologies is essential to bridge the research gaps and achieve the desired goals of a secure and reliable decentralized web.

Despite promises of security and anonymity, security testing in the context of Web 3.0 is an under-researched area. This represents an important research gap, as many security challenges remain unrecognized and unaddressed.

SQAMIA 2024: Workshop on Software Quality, Analysis, Monitoring, Improvement, and Applications, September 9–11, 2024, Novi Sad, Serbia

*Corresponding author.

nika.jersic@um.si (N. Jeršič); muhamed.turkanovic@um.si (M. Turkanović); tina.beranic@um.si (T. Beranič)

0000-0002-5079-5468 (M. Turkanović); 0000-0001-6518-5876 (T. Beranič)

The goal of our research is to provide a comprehensive analysis of the current state of security testing in Web 3.0. Our research will answer the following research questions:
RQ1: What are the common security vulnerabilities in Web 3.0 technologies, and what types of security vulnerabilities are most frequently mentioned in existing research?
RQ2: What methods and techniques are currently used for security testing in Web 3.0 technologies?
RQ3: What are the specific QA strategies applied to Web 3.0 security testing?
RQ4: What are the challenges and gaps in current research on Web 3.0 security testing and QA?

With this paper, we want to contribute to the discussion about a potential research gap in the field of QA, focusing on security testing, within Web 3.0 technologies. For our research, we conducted a literature review to find existing research on this topic. We intend to identify and address these gaps to lay the groundwork for future research and development in creating more secure and reliable Web 3.0 applications for all users. With greater emphasis on security testing and closer cooperation between all involved, we can achieve significant improvements in the security of Web 3.0 technologies. With this article, we want to encourage the community to actively tackle these challenges and lay the foundations for a safer digital future.

The rest of the paper is structured as follows. In the second chapter we present the general background on Web 3.0, QA and security testing. In chapter three, we present the current state of existing articles in four digital libraries. The fourth chapter describes results from reviewing different articles on researched topic. In chapter five, we focus on answering our research question within a discussion and present key insights. And finally, in chapter six, we conclude our paper and discuss further research.

2. Web 3.0 and security testing

2.1. Web 3.0

As claimed in [2], a common synonym for Web 3.0 and the enthusiasm for Web 3.0 technologies is the decentralization. As an infrastructure, the Web 3.0 is not only decentralization, there are many more components, that make Web 3.0 the new future infrastructure of the Internet [4].

One such component is blockchain technology, which eliminates the need for trusted third parties through a consensus mechanism [5]. It is made of cryptographically secured and linked records, known as blocks, that are resilient to single points of failure, utilizing a peer-to-peer protocol. This mechanism ensures a shared sequence of transactions and blocks, while preserving the integrity and consistency across globally distributed nodes [6]. By design, main blockchain components are decentralization, integrity and auditability. Decentralization refers to the process of dispersing power, control and decision-making from one central place to numerous independent and equal entities. In the context of web technologies, this means a shift from the current situation, where most content and user data on the web is controlled by a few large technology companies, to a system where control is evenly distributed among all internet users. Successful decentralization of the web requires new technologies and protocols that allow secure use of the web without the need to trust individual centralized entities [7].

Additionally, smart contracts are another vital component of the Web 3.0 ecosystem. These are programs that automate and enforce agreements between parties without the need for intermediaries [6]. Their importance and potential vulnerability make them the subject of in-depth security scrutiny, while other components of the Web 3.0 ecosystem, such as user interfaces, APIs and infrastructure, are often neglected. A one-sided focus can also be risky, as potential weaknesses in these components can affect the overall security posture of the system [8].

2.2. Security testing within QA

Testing in the context of QA primarily evaluates, how software satisfies requirements, quality and expectations of customers.

It is divided into functional testing and non-functional testing. Functional testing is usually performed before releases and makes sure all tests pass before a new build release. Non-functional testing, performed during the development phase, evaluates aspects such as performance, usability, reliability, and scalability [9].

Security testing ensures that software and systems effectively protect sensitive information and data from security vulnerabilities. Security testing techniques, such as risk-based testing and model-based security testing, help in identifying vulnerabilities that could be exploited by malicious actors. Along with the high level of security required for their global functioning, thorough security testing is necessary to ensure robustness and reliability. This proactive approach allows for the mitigation of potential security risks before they are exploited [10].

3. The current state of existing research

To understand the current state of ensuring quality and security of Web 3.0, we conducted a thorough literature review. We searched across multiple digital libraries, including IEEE Xplore, ACM Digital Library, ScienceDirect, and SpringerLink, to ensure a comprehensive collection of relevant literature. Our goal was to explore various research papers connecting Web 3.0 with security testing within QA. To achieve this, we utilized different search strings to find research. These search strings are presented and discussed in table 1.

Table 1
Search Strings and Descriptions

No.	Search String	Description
1	("Quality Assurance") AND ("Web 3.0")	This basic search string was used to find general information on only QA in the context of Web 3.0 technologies.
2	("web 3.0" OR "Web 3.0" OR "Web3" OR "blockchain" OR "smart contracts") AND ("QA" OR "Quality Assurance")	This search string is a variant of search string 1, where we look to see if we can find more results if our keywords are written in different formats.
3	("security" OR "attacks" OR "vulnerabilities" OR "security analysis") AND ("web 3.0" OR "Web 3.0" OR "Web3" OR "blockchain" OR "smart contracts")	Our research paper also looks at security testing in the context of Web 3.0. Therefore, we searched with a more specific search string, adding keywords about security.
4	("security" OR "attacks" OR "vulnerabilities" OR "security analysis") AND ("web 3.0" OR "Web 3.0" OR "Web3" OR "blockchain" OR "smart contracts") AND ("QA" OR "Quality Assurance")	To include quality assurance, we used an even more specific search string that combines security aspects and quality assurance in the context of Web 3.0 technologies. This search string was used to obtain a more comprehensive overview of information on quality assurance and security analysis in Web 3.0 technologies.

To ensure that the papers included in the research show the actual current state of the existing research on this topic, we introduced inclusion and exclusion criteria to the literature review. Studies were included if they:

- are written in English language,
- are academic journals or conference papers,
- are applied to our research string keywords,
- fall under Computer science, Engineering or Informatics, in the subject areas

Studies were not included in the research, if they:

- are not accessible through the university network,
- are books (except proceeding books),
- are not applied to our keywords in search strings.

We decided to include only English literature, because English is the most widely used language in scientific and academic literature, ensuring wide accessibility and visibility of research. We avoid translation and interpretation problems that could affect the accuracy of the analysis. Academic journals and conference papers are peer-reviewed to ensure high quality information. These sources are credible and properly checked, which is crucial for scientific research. The use of keywords specific to our research ensures that the sources chosen are directly related. This increases the relevance and accuracy of our findings and allows for a deeper understanding of specific issues. We decided not to review books for several reasons. Firstly, books often contain extensive and general information that is not always directly relevant to our specific research topic. Finding relevant information in books would require significantly more time and effort compared to reviewing articles, which are usually more focused on specific research questions. Secondly, articles in scientific journals are often more up-to-date and include the latest research and data, which is crucial for our analysis. In addition, the articles are peer-reviewed and thus provide a higher level of scientific credibility. Our analysis focuses on the latest trends and data, so we have decided to limit our search to scientific articles in digital libraries.

4. Results

In this section, we present the results of the literature review, focusing on the intersection of Web 3.0 technologies and security testing. Table 2 below presents the results of defined query searches, illustrating the number of articles and papers available in each database for the specified queries, before and after applying inclusion and exclusion criteria. We performed a thorough analysis of the articles, on the basis of which we identified several specific trends. These trends are presented in more detail in the tables 3 and 4 that highlight the frequency and overlap of different themes in the reviewed literature.

Table 2
Search results for various queries across different digital libraries before and after applying criteria.

Search string number	Library	Number of results before including criteria	Number of results after including criteria
1	IEEE	0	0
	ScienceDirect	2	2
	SpringerLink	7	1
	ACM	9	0
2	IEEE	80	79
	ScienceDirect	936	85
	SpringerLink	1,202	236
	ACM	372	95
3	IEEE	17,185	15,951
	ScienceDirect	16,558	7,747
	SpringerLink	26,428	17,224
	ACM	5,887	98
4	IEEE	39	38
	ScienceDirect	667	270
	SpringerLink	970	350
	ACM	300	76

As can be seen in table 2, for search string 3, we found that the number of articles would be too large for a detailed analysis, so we had to apply additional criteria to limit the number of results and examined the only first 100 results of search results. In addition, for all search strings, we found that most of the articles are related to other fields where the data or records were in a blockchain system,

which was not relevant for our study. We also noticed that many articles were repetitive, especially within the same digital libraries, which further reduced the number of relevant articles for our analysis. The total number of articles used in this analysis is 9. It is also important to note that the criteria for inclusion were very specific and strict. These strict criteria resulted in a reduction in the number of final articles, which in turn means that these articles are more targeted and relevant to our research. In this way, we have ensured that our analysis is based on the most relevant and high-quality sources, which is essential for the credibility and usefulness of our findings.

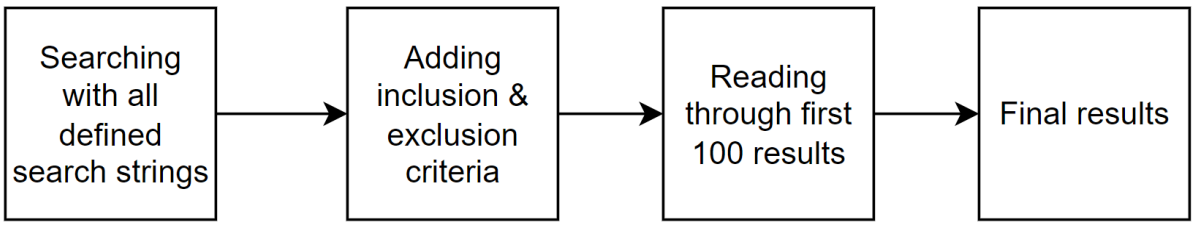

Figure 1: Flow diagram of our analysis of articles.

Figure 1 presents our flow of analyzing the results. This process was repeated for every search string within each library. The first block was inserting our defined search strings into the digital libraries search bars. This step got us the results from table 2, column Number of results before including criteria, which was in total 70,641 results. When adding inclusion and exclusion criteria in the second block in figure 1 we included already defined criteria. We decided to read only the first hundred results, sorted by "relevance" in each digital library, which was at the end the total of 1,074 results. Our paper analysis of the results was primarily based on reading the title. If the title did not give us a clear indication of what this paper is about, we continued to read the abstract of that paper. And if the abstract did not give us a clear indication of the content, we read the full text to determine if the paper will be included in our final analysis.

4.1. The analysis

When analyzing the articles from the right side of table 2, in the column Number of results after including criteria, we noticed specific trends when connecting Web 3.0 technologies, QA and security testing. These trends are presented in table 3, where we look at where specific topics appear in several papers and table 4, where we look at overlapping topics, and where they appear.

Table 3 shows the frequency of individual topics related to security testing in Web 3.0 technologies. In particular, the topic of smart contracts appears namely five times in different sources. Where the topic of security vulnerabilities appears six times in different sources. Blockchain technology is also prominent, appearing in four articles. These highlight the strong emphasis on vulnerabilities in smart contracts and blockchain technology within Web 3.0 security testing.

Table 3
Occurrences of topics in security testing in Web 3.0 technologies.

Topic	Occurrences	Sources
blockchain	4	[11], [12], [13], [14]
smart contracts	5	[15], [16], [2], [11], [12]
security vulnerabilities	6	[15], [16], [11], [12], [17], [18]
security checks	2	[2], [5]
security frameworks	3	[5], [13], [18]
information security goals	1	[14]

In addition, in table 4 overlapping areas are presented, where certain topics are discussed simultaneously in several articles. For example, the overlap between smart contracts and security vulnerabilities

is noticeable in two sources [15], [16], indicating a critical area of concern for researchers. Another significant overlap is observed between smart contracts, security vulnerabilities, and blockchain, which is covered in three sources. This suggests that while smart contracts are a pivotal component of Web 3.0, their security remains a significant challenge, necessitating focused research efforts to identify and mitigate vulnerabilities.

Table 4
Topics covered by each article in security testing and Web 3.0 technologies.

Overlapping areas	Sources
smart contracts, security vulnerabilities	[15], [16]
smart contracts, Security Checks	[2]
smart contracts, security vulnerabilities, Blockchain	[11]
smart contracts, security vulnerabilities, Blockchain, QA	[12]
security checks, security frameworks	[5]
blockchain, security frameworks	[13]
blockchain, information security goals	[14]

To further elaborate, we analyzed the security vulnerabilities and attacks in Web 3.0 technologies. Table 5 shows each vulnerability and attack and the sources in which they are discussed. The studies propose suggestions to mitigate the risks, such as regular software updates, the use of static and dynamic code analysis tools, the implementation of software development security cycles, stronger security configurations and the use of Security Information and Event Management (SIEM) systems.

4.2. Analyzing individual papers

From our analysis, we can see when security testing is carried out in Web 3.0 technologies, it is often focused on smart contracts. Despite their benefits, smart contracts are vulnerable to several security risks, such as Reentrancy, Denial-of-Service, Timestamp Dependence and Mishandled Exception [15]. A study conducted by Huang et al. [16] presents a deeper analysis where they categorize the most common smart contract security vulnerabilities into three different causes, Solidity language, blockchain platform and misunderstanding of common practices.

The second most popular topic when connecting security within Web 3.0 technologies is blockchain. Alfaw et al. [11] discuss several critical aspects of blockchain technology and its security challenges, mostly focusing on the lack of central authority and the role of cryptography in ensuring security and privacy. It highlights the necessary security attributes for blockchain systems and classifies various security threats and vulnerabilities into categories such as client vulnerabilities, consensus mechanism vulnerabilities, mining pool vulnerabilities, network vulnerabilities, and smart contract vulnerabilities, with examples including digital signature vulnerabilities, mining malware, and smart contract vulnerabilities. Additionally, the paper addresses the need for secure design and deployment of smart contracts to prevent security breaches and reviews security solutions such as digital signatures, zero-knowledge proofs, and attribute-based encryption, while also highlighting sector-specific solutions for healthcare and transaction sectors.

Shen et al. [21] proposed a QA framework, based on blockchain. This paper can illustrate that `blockchain` in relation to `QA` is usually a tool or technology to perform QA, rather than as a research area where QA would be performed on the blockchain or in general, Web 3.0 technologies. Similar to this paper, Li et. al [12] focuses on QA and data security in smart grids with usage of blockchain, before adding data on a decentralized storage. The paper also analyzes security threats and proposes mechanisms to handle them, such as validating transactions and using smart contracts.

Also focusing more on the human factor, the study by Xiang et al. [22] presented how developers working on DeFi (Decentralized Finance) projects respond to attacks during the development process. They found that the state of the developers' response to attacks is barely reaching minimal standards. Furthermore, they found that a lot of attacks could be prevented with the proper configuration. Additionally, Liu et al. [2] report that developers frequently enhance transaction-reverting statements

Table 5
Common vulnerabilities and attacks found in sources with mitigation strategies.

Vulnerability/Attack	Mitigation Strategy	Sources
digital signature vulnerability	stronger randomization techniques	[11], [5]
user addresses vulnerability	identity verification and secure protocols	[11], [5]
delegate call vulnerability	avoid using delegate call with untrusted contracts	[15], [5], [16]
block-hash vulnerability	avoid using block hashes as randomness	[15], [5], [16]
hash function vulnerability	use resistant hash functions	[11], [5]
timestamp dependency vulnerability	avoid using block timestamps	[15], [5], [16], [19]
transaction ordering dependency (TOD) vulnerability	fair transaction ordering mechanisms	[15], [5], [16]
arithmetic underflow/overflow vulnerability	use safe math libraries	[15], [5], [16]
freezing ether vulnerability	implement escape hatches	[15], [5], [16]
reentrancy attack	checks-effects-interactions pattern, Oyente tool	[15], [5], [16], [19]
alternative history attack	stronger consensus mechanisms	[11], [5]
finney attack	implement time delays	[11], [5]
block withholding attack	ensure prompt block broadcasting	[11], [5]
bribery attack	strict anti-bribery rules	[11], [5]
selfish mining attack	increase attack threshold, random nodes	[11], [5], [15]
pool hopping attack	design reward systems to discourage hopping	[11], [5]
fork after withholding attack	detect and penalize block withholding	[11], [5]
delay attack	ensure rapid transaction propagation	[11], [5]
replay attack	unique transaction signatures	[11], [5], [18]
sybil attack	robust identity verification	[11], [5], [17]
DDoS attack	fee-based and age-based mempool management	[11], [5], [15], [20], [17]
transaction malleability attack	unique transaction identifiers	[11], [5]
timejacking attack	synchronize network time	[11], [5]
cryptojacking attack	detection and prevention mechanisms	[11], [5]

by adding clauses, variables, or new transaction-reverting statements, which are primarily used for range and logic checks. These statements are typically used for seven types of authority verifications or validity checks and are often customized in template contracts.

As a countermeasure to attacks, the study presented by Kushwaha et al. [5] presents security checks, which are crucial to ensure the integrity and security of these technologies. Furthermore, Cha et al. [13] introduced a holistic approach to managing security risks in blockchain applications. The proposed framework for permissioned blockchain applications maps its controls to existing security guidelines and standards, such as ISO/IEC 27001. This paper also emphasizes the growing need for security risks management framework for permissioned blockchains. In his paper, Monev [14] aims to fill gaps in existing literature by establishing foundational information security goals vital for the secure operation and development of blockchain solutions. It proposes a baseline of nine information security goals for blockchain solutions:

- Accountability, which establishes an unambiguous link between people's responsibility and

electronic identity and ensures that participants are reasonably accountable for their actions.
- Auditability, which ensures that data, systems, software or other elements are reasonably auditable; Nonrepudiation, which ensures that the entity that caused the event cannot deny that the event occurred.
- Authenticity, which confirms that the entity is who it claims to be.
- Availability, which ensures that data is accessible at a predetermined time and speed.
- Confidentiality, which protects data so that it is available only to those for whom it is intended.
- Integrity, which ensures that data is modified as intended and designed.
- Privacy, which allows individuals to choose what private information to share and with whom, which is done with reasonable control.
- Reliability, which ensures that the system consistently delivers the intended behavior and results.

These goals are intended to improve the security framework for blockchain solutions and to provide a comprehensive approach to protecting data and maintaining the reliability of blockchain systems.

In several articles, as the tables 3 and 5 show, we notice that there is a lot of discussion about vulnerabilities in Web 3.0 technologies and how important it is to take care of security. In table 3 we see that topics such as "security vulnerabilities" and "smart contracts" are frequently discussed. This shows the great interest of researchers in identifying security risks. However, only a few articles focus on strategies to prevent these vulnerabilities and attacks. For example, vulnerabilities such as "digital signature vulnerability" and "delegate call vulnerability" are mentioned, but the proposed strategies to prevent them are less often discussed.

In addition, the table 4 shows articles, such as [11] and [12], cover multiple areas that include both vulnerabilities and security strategies. This highlights the need for more in-depth research that not only identifies vulnerabilities, but also suggests concrete measures to prevent them. Thus, it would be useful for future research to also include concrete measures and methods to protect against known security threats, thereby contributing to a more comprehensive understanding and management of security risks in Web 3.0.

Table 6
Gaps in security testing and QA in Web 3.0 technologies.

Area	Detected weaknesses	Research gaps
QA	lack of research addressing QA as a research area in Web 3.0	a search for `("Quality Assurance" AND "Web 3.0")` in table 2 indicates a research gap or lack of distinction between traditional QA approaches and those in Web 3.0
security testing	lack of concrete measures to prevent known vulnerabilities such as digital signatures and delegate call vulnerabilities	detailed strategies to prevent security vulnerabilities are needed
QA tools and techniques	lack of new tools and techniques for automated testing of smart contracts and blockchain applications	develop new tools and techniques and use advanced technologies such as artificial intelligence and machine learning
specific application areas	lack of tailored QA strategies for different sectors such as healthcare and DeFi	develop tailored QA strategies that address specific security and functional challenges in different sectors

Table 6 presents key findings of research gaps between QA and Web 3.0. The most important finding was a lack of research addressing QA as a research area in its own right in the context of Web 3.0 technologies. Searching for the keywords `("Quality Assurance" AND "Web 3.0")` in the table 2 showed that there is either a research gap in this area or no clear distinction between traditional QA

approaches and those specific to Web 3.0 technologies. At the moment, it is not clear whether the lack of differences between QA in traditional and Web 3.0 environments is due to the lack of hits in digital libraries, or whether there is in fact a research gap that has not been addressed yet. This suggests the need for further research to clarify whether existing QA methods need to be adapted for use in Web 3.0 or whether this is an undiscovered research area.

5. Discussion

The analysis, referencing our RQ1, revealed several common security vulnerabilities of Web 3.0 technologies. Among the most frequently mentioned are vulnerabilities in smart contracts and blockchain technology. Specific vulnerabilities, that are also presented in 5 include digital signature vulnerability, delegate call vulnerability, block hash vulnerability, hash function vulnerability, replay vulnerability, and timestamp dependency vulnerability. These vulnerabilities are critical because of the potential to compromise the security and functionality of Web 3.0 applications. The frequent mention of these vulnerabilities in various sources indicates their prevalence and the need for reliable security measures to eliminate them. For example, replay vulnerability appears in four different sources, indicating an important area of concern for researchers.

Various methods and techniques, as addressed in RQ2, are used in the security testing of Web 3.0 technologies. Static and dynamic code analysis tools are widely used to detect vulnerabilities in smart contracts and blockchain applications. In addition, methods such as the checks-effects-interactions pattern and the use of the Oyente tool are used to mitigate specific attacks such as reentrancy. The studies also highlight the importance of using secure development practices, such as stronger randomization techniques, identity verification protocols, and secure math libraries, to increase security. Deploying security information and event management (SIEM) systems is another key technique for monitoring and managing security events in real time.

As explored with RQ3, quality assurance strategies for Web 3.0 security testing are evolving and focus on preventative and corrective actions. Specific quality assurance strategies include the implementation of security frameworks adapted to Web 3.0 technologies, the use of robust security checks during the development process, and the use of audit and non-responsibility principles to ensure accountability and traceability. However, there is a gap in the literature regarding comprehensive quality assurance frameworks specifically designed for Web 3.0, as most existing strategies are adapted from traditional quality assurance approaches without substantial modifications to address the unique challenges of Web 3.0 environments.

The analysis, in response to RQ4, revealed several challenges and gaps in current research on Web 3.0 security testing and quality assurance, which is also presented in table 6. A significant challenge is the lack of comprehensive quality assurance frameworks specifically designed for Web 3.0 technologies. Although traditional quality assurance strategies are often used, they may not fully address the unique security challenges presented by Web 3.0 applications. In addition, there is a gap in the development of new tools and techniques for automated testing of smart contracts and blockchain applications. The lack of detailed strategies to prevent known vulnerabilities, such as digital signature and delegate call vulnerabilities, highlights the need for more focused research on mitigation measures. In addition, there is a lack of tailored quality assurance strategies for different application areas such as healthcare and DeFi, which have specific security and functional requirements.

6. Conclusion

Web 3.0 technologies, specifically, blockchain in the context of security testing, are usually focused on smart contracts and other Web 3.0 solutions. During this study, no articles included security testing of Web 3.0 as a concept. Blockchain is usually seen as an infrastructure or platform with which QA is performed, rather than as an object on which to focus specific QA procedures or research. Despite the fact that terms such as `Web 3.0` and `QA` are well represented in the literature, QA does not seem to be

a major area of research in Web 3.0 context. The role of blockchain in security testing is primarily as a means to improve QA processes, providing tools and frameworks that improve security and reliability. However, the human factor remains a critical component, with a need for better training and awareness among developers to respond effectively to security threats.

The integration of security testing within Web 3.0 presents both opportunities and challenges. The analysis reveals that smart contracts and blockchain technologies are at the forefront of security issues. While smart contracts offer significant advantages, they are prone to specific vulnerabilities that require targeted security measures. Blockchain technology, despite its inherent advantages, faces a number of security threats that require robust cryptographic solutions and secure deployment practices.

Finally, comprehensive security risk management frameworks and fundamental information security objectives are essential for the safe development and operation of Web 3.0 technologies. By addressing these multi-faceted security challenges through technical and human-centered approaches, the potential of Web 3.0 technologies can be more safely and effectively exploited. Future research should continue to explore these areas, focusing on developing innovative security solutions and improving existing frameworks to ensure the resilience and integrity of Web 3.0 systems.

Acknowledgments

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

References

[1] A. Khanjani, R. Sulaiman, The Process of Quality Assurance under Open Source Software Development, 2011.

[2] Z. Liu, Y. Xiang, J. Shi, P. Gao, H. Wang, X. Xiao, B. Wen, Q. Li, Y. C. Hu, Make web3.0 connected, IEEE Transactions on Dependable and Secure Computing 19 (2022) 2965–2981. doi:10.1109/TDSC.2021.3079315.

[3] U. W. Chohan, Web 3.0: The future architecture of the internet? (2022). URL: https://ssrn.com/abstract=4037693.

[4] C. Guan, D. Ding, J. Guo, Web3.0: A review and research agenda, Proceedings - 2022 RIVF International Conference on Computing and Communication Technologies, RIVF 2022 (2022) 653–658. doi:10.1109/RIVF55975.2022.10013794.

[5] S. S. Kushwaha, S. Joshi, D. Singh, M. Kaur, H. N. Lee, Ethereum smart contract analysis tools: A systematic review, IEEE Access 10 (2022) 57037–57062. doi:10.1109/ACCESS.2022.3169902.

[6] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, F. Y. Wang, Blockchain-enabled smart contracts: Architecture, applications, and future trends, IEEE Transactions on Systems, Man, and Cybernetics: Systems 49 (2019) 2266–2277. doi:10.1109/TSMC.2019.2895123.

[7] G. Korpal, D. Scott, Decentralization and web3 technologies (2022). URL: https://doi.org/10.36227/techrxiv.19727734.v1. doi:10.36227/techrxiv.19727734.v1.

[8] A. H. Pranav, M. Latha, S. Ashwin, R. Chinnaiyan, Blockchainas a service (baas) framework for government funded projects e-tendering process administration and quality assurance using smart contracts, Institute of Electrical and Electronics Engineers Inc., 2021. doi:10.1109/ICCCI50826.2021.9402348.

[9] J. Metsä, M. Katara, T. Mikkonen, Testing non-functional requirements with aspects: An industrial case study, Proceedings - International Conference on Quality Software (2007) 5–14. doi:10.1109/QSIC.2007.4385475.

[10] M. Felderer, M. Büchler, M. Johns, A. D. Brucker, R. Breu, A. Pretschner, Security testing: A survey, Advances in Computers 101 (2016) 1–51. doi:10.1016/BS.ADCOM.2015.11.003.

[11] A. Alfaw, W. Elmedany, M. S. Sharif, Blockchain vulnerabilities and recent security challenges: A

review paper, 2022 International Conference on Data Analytics for Business and Industry, ICDABI 2022 (2022) 780–786. doi:10.1109/ICDABI56818.2022.10041611.

[12] F. Li, X. Li, Y. Fu, P. Zhao, S. Liu, A secure and privacy preserving incentive mechainism for vehicular crowdsensing with data quality assurance, IEEE Vehicular Technology Conference 2021-September (2021). doi:10.1109/VTC2021-FALL52928.2021.9625317.

[13] S. C. Cha, C. M. Shiung, G. Y. Lin, Y. H. Hung, A security risk management framework for permissioned blockchain applications, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 301–310. doi:10.1109/SmartIoT52359.2021.00055.

[14] V. Monev, Defining and applying information security goals for blockchain technology, 2020 34th International Conference on Information Technologies, InfoTech 2020 - Proceedings (2020). doi:10.1109/INFOTECH49733.2020.9211073.

[15] E. M. Sifra, Security vulnerabilities and countermeasures of smart contracts: A survey, Proceedings - 2022 IEEE International Conference on Blockchain, Blockchain 2022 (2022) 512–515. doi:10.1109/BLOCKCHAIN55522.2022.00080.

[16] Smart contract security: A software lifecycle perspective, IEEE Access 7 (2019) 150184–150202. doi:10.1109/ACCESS.2019.2946988.

[17] S. Shukla, I. Gupta, K. Naresh, Addressing security issues and future prospects of web 3.0, 2022 2nd Asian Conference on Innovation in Technology, ASIANCON 2022 (2022). doi:10.1109/ASIANCON55314.2022.9908800.

[18] R. Bruwer, H. Jacobus, Web 3.0: Governance, risks and safeguards, 2016.

[19] R. Pise, S. Patil, A deep dive into blockchain-based smart contract-specific security vulnerabilities, 2022 IEEE International Conference on Blockchain and Distributed Systems Security, ICBDS 2022 (2022). doi:10.1109/ICBDS53701.2022.9935949.

[20] E. M. Poleshchuk, I. A. Shcherbinina, S. E. Putilova, Security analysis of smart contracts in blockchain networks, Proceedings - 2022 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology, USBEREIT 2022 (2022) 252–254. doi:10.1109/USBEREIT56278.2022.9923336.

[21] X. Shen, C. Xu, L. Zhu, R. Lu, Y. Guan, X. Zhang, Blockchain-based lightweight and privacy-preserving quality assurance framework in crowdsensing systems, IEEE Internet of Things Journal 11 (2024) 974–986. doi:10.1109/JIOT.2023.3288349.

[22] D. Xiang, Y. Lin, L. Nie, Y. Zheng, Z. Xu, Z. Ding, Y. Liu, An empirical study of attack-related events in defi projects development, Empirical Software Engineering 29 (2024). doi:10.1007/s10664-024-10447-7.