Journal of Digital Media & Interaction
ISSN 2184-3120
Vol. 8, No. 19, (2025), pp. 9-27
DOI: 10.34624/jdmi.v8i19.40178

A Pragmatic Approach for Web3 Software Quality Assurance Based on International Guidelines

Rodrigo Antunes
Value for Health CoLAB,
Portugal
rodrigo.antunes@vohcolab.org
0009-0009-0083-5570

Liliana Freitas
Value for Health CoLAB; CEG-IST, Universidade de Lisboa,
Portugal
liliana.freitas@vohcolab.org
0000-0001-9513-2476

Pedro Dias
Value for Health CoLAB; CHRC, NMS|FCM, Universidade Nova de Lisboa,
Portugal
pedro.dias@vohcolab.org

Luís Silva
Exeedme, Portugal
luis.silva@exeedme.com

Federico Guede-Fernandez
Value for Health CoLAB; LIBPhys, NOVA School of Science and Technology,
Portugal
federico.guede@vohcolab.org
0000-0003-2762-0333

Salomé Azevedo
Value for Health CoLAB; CHRC, NMS|FCM, Universidade Nova de Lisboa; CEG-IST, Universidade de Lisboa, Portugal
salome.azevedo@vohcolab.org
0000-0003-1234-9464

Received: 4 June 2025
Accepted: 24 November 2025

Abstract
Ensuring software quality in the Web3 ecosystem presents unique challenges due to its decentralized architecture and evolving technical landscape. While international standards such as the SQuaRE (Systems and software Quality Requirements and Evaluation) framework offer structured approaches for quality assurance, they are often perceived as overly theoretical and not directly applicable to blockchain-based applications. This study aims to translate these standards into actionable practices suitable for Web3 environments, thereby supporting compliance and fostering stakeholder trust. Using the Design Science Research methodology, complemented by Lean Startup principles, a practical quality assurance guide was co-developed through collaboration between VOH.CoLAB researchers and the Exeedme project team and inspired by the practical experience in gaming and digital assets trading blockchain-based platforms. The resulting guide includes a structured framework comprising eight testing domains, 16 sub-domains and 108 targeted tests, with the domains addressing critical features of blockchain software, including, functional suitability, integration, security, performance, usability, portability, recoverability and resilience. This work contributes to the operationalization of international quality standards in decentralized technology, promoting more resilient and trustworthy blockchain applications.

Keywords *Web3 software quality, Blockchain applications, SQuaRE framework, Quality assurance, Design Science Research*

1. Introduction

The gaming industry has undergone transformative changes in recent years, driven by the growing integration of blockchain technologies. Games like Counter-Strike 2 (CS2) have fostered robust growth of the economy of virtual items, commonly known as “skins”, with marketplaces facilitating billions of dollars in transactions. The adoption of cryptocurrencies and non-fungible tokens (NFTs) has further expanded this ecosystem, enabling players to buy, sell, and trade digital assets with real-world value (CSGO & CS2 Item Economy, 2024). The rise of blockchain gaming is also evident in the sector's financial traction, with blockchain-based gaming companies attracting over $1.1 billion in investment in the second quarter of 2024 alone (NFT statistics in 2024, n.d.). Blockchain technology, which enables the creation of valuable digital assets such as NFTs, remains in a rising trend despite a slowdown in trading volume after the 2022 peak. The NFT market, while having consolidated in recent years, still shows strong growth, with a 50% increase in transaction volume in the first quarter of 2024 (NFT - Worldwide Statista Market Forecast, n.d.).

This rapid growth underscores the importance of ensuring software quality in Web3 applications, particularly in gaming, where user trust and platform stability are essential. Ensuring success and continuous improvement of blockchain-based software requires that its development be accompanied by the verification and enhancement of the solution's technological quality. However, guaranteeing software quality in the Web3 context presents unique challenges due to the decentralized nature of blockchain technology (Hossain Faruk et al., 2024). While the SQuaRE (Systems and software Quality Requirements and Evaluation) international standards provide structured frameworks for software quality assurance (Febrero et al., 2016; Shtefan & Zaporozhets, 2021), they are often considered too theoretical, failing to address the specificities of blockchain-based applications (Gordieiev et al., 2024; Tsuda et al., 2019). Ensuring compliance with these standards is crucial for gaining market advantages and establishing trust among stakeholders, namely developers and users (Mubarkoot et al., 2023), yet the existing frameworks lack a pragmatic implementation guidance tailored to Web3 specific requirements.

This study addresses this gap by proposing a structured and actionable quality assurance guide for blockchain-based applications, built inspired by trading gaming assets platforms. Developed through the collaboration between VOH.CoLAB researchers and the Exeedme project manager - a blockchain company specialized in gaming and skin trading for CS2 - this work draws from real-world experiences to tailor the international standards to the needs of Web3 software development and thus closing the gap between the theoretical ISO standards and the practical needs of the companies working in the gaming sector. The contributions of this study are twofold: it bridges the gap between software quality standards mostly based on Web2 environments and the practical realities of Web3 development and it supports the operationalization of international software quality standards in decentralized technologies, promoting higher trust, robustness, and adoption of Web3 applications.

This study was conducted as part of the Blockchain.PT Decentralizing Portugal with Blockchain agenda that brings together a nationwide coalition of 56 entities, including companies, research centers, associations, and public institutions to harness blockchain technology as a catalyst for innovation and international business development. This agenda aims to deliver 26 scalable, export-oriented products, with a strong focus on practical applications such as farm-to-fork traceability through the integration of IoT and blockchain, digital asset management in real estate and other industries, and solutions for seamless data exchange across different blockchain systems. Ultimately, the project seeks to position Portugal at the forefront of blockchain innovation in Europe and drive the country's digital transformation (BlockchainPT, 2025).

This paper is structured as follows: The next section outlines the methodology used to develop the proposed Web3 software quality testing guide, including the stages of the design process. The results section presents the identified test domains and subdomains, along with examples of specific tests that address both technical functionality and user experience. The discussion section interprets the key findings and highlights theoretical and practical implications. Finally, the conclusion summarizes the contributions of the study and outlines directions for future research.

2. Methodology

The Design Science Research (DSR) methodology (Hevner & Park, 2004) was followed to co-create a structured quality assurance guide tailored to blockchain-based software systems. DSR is particularly suited for addressing complex, practice-oriented problems through the creation of innovative artifacts – such as models, frameworks, and processes – that are rigorously designed and evaluated in collaboration with stakeholders (Hevner & Park, 2004; Peffers et al., 2007). In this context, the artifact developed was a structured quality assurance guide to improve software robustness and compliance in Web3 applications.

Based on the methodological approach of (Londral et al., 2022), this study integrated elements of the Lean Startup method into the DSR methodology. While Londral et al. applied this approach in the health domain by structuring discussions around patient pathways, our study adapts it to the Web3 context by grounding the iterative cycles in the mapping of software quality assurance domains. This integration enabled the development of the guide through short iteration cycles, ensuring agility and responsiveness to real-world constraints and user needs. The methodology included six stages led by the VOH.CoLAB research team, in close collaboration with the Exeedme project manager, enabling a context-driven approach focused on the practical experience needs. Figure 1 presents the stages followed throughout the work.

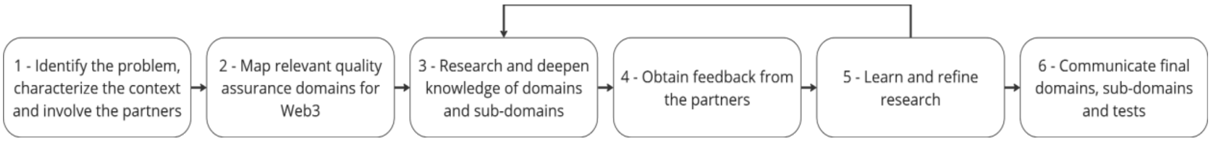

1 - Identify the problem, characterize the context and involve the partners
2 - Map relevant quality assurance domains for Web3
3 - Research and deepen knowledge of domains and sub-domains
4 - Obtain feedback from the partners
5 - Learn and refine research
6 - Communicate final domains, sub-domains and tests

Figure 1. Test roadmap development methodology (adapted from (Londral et al., 2022))

2.1. Stage 1 - Identifying the Problem, Characterizing the Context, and Involving the Partners

The first stage followed the principles of DSR methodology, emphasizing early collaboration between stakeholders to ensure the solution addressed real needs. A series of videoconference meetings were held between the VOH.CoLAB research team and the Exeedme project team to discuss challenges related to software quality in blockchain-based gaming platforms.

These discussions helped both teams bring their perspectives: Exeedme shared insights from their operational experience in developing and managing a blockchain-based marketplace for virtual gaming assets, while VOH.CoLAB contributed expertise in software quality and methodological frameworks. Together, they explored the limitations of existing standards, such as SQuaRE, for Web3 environments and identified the need for more practical, tailored guidance.

The main output of this stage was a shared understanding of the problem and its context.

2.2. Stage 2 - Map Relevant Quality Assurance Domains

This stage involved mapping test domains and subdomains to capture software development concerns from both teams and to provide the foundation for the following stages of solution development. Building on the challenges identified in Stage 1, an initial exploratory search in peer-reviewed databases (Scopus, Web of Science, and Google Scholar) using keywords such as software quality and blockchain quality assurance was conducted to scope the field and understand the key challenges. The review then concentrated on ISO/IEC documentation, which provided a preliminary set of domains to be used as the structured basis for supporting the definition of the relevant quality assurance domains.

2.3. Stages 3, 4, 5 - Iteration Cycles: Develop, Test and Learn

These stages were repeated for two iterative cycles of development, testing, and refinement, which led to the creation of the structure of the relevant domains and subdomains and to the discussion of potential tests for each domain and sub-domain. During these cycles, the VOH.CoLAB and Exeedme teams collaborated both through videoconference meetings and asynchronously by exchanging documents with the relevant information. The first cycle consisted of refining the preliminary set of test domains identified on Stage 2. With that objective in view, each domain was also divided into multiple sub-domains. The second cycle consisted of collaboratively defining a set of tests for each of the domains and subdomains. VOH.CoLAB proposed a set of tests for each domain and Exeedme provided valuable insights of the main functionalities, concerns and other technical details of Web3 applications, thus obtaining a set of tests that reflected the main areas of concern for platforms that use these technologies.

2.4. Stages 6 - Communicate final Domains, Sub-Domains and Tests

In the final stage, the full set of test domains, sub-domains, and tests was reviewed, finalized, and organized into a structured quality assurance guide. The final version was documented in a clear and practical format, making it easy to use for developers and teams working in Web3. The completed guide was then shared with all stakeholders, concluding the co-development process and preparing it for real-world testing.

3. Results

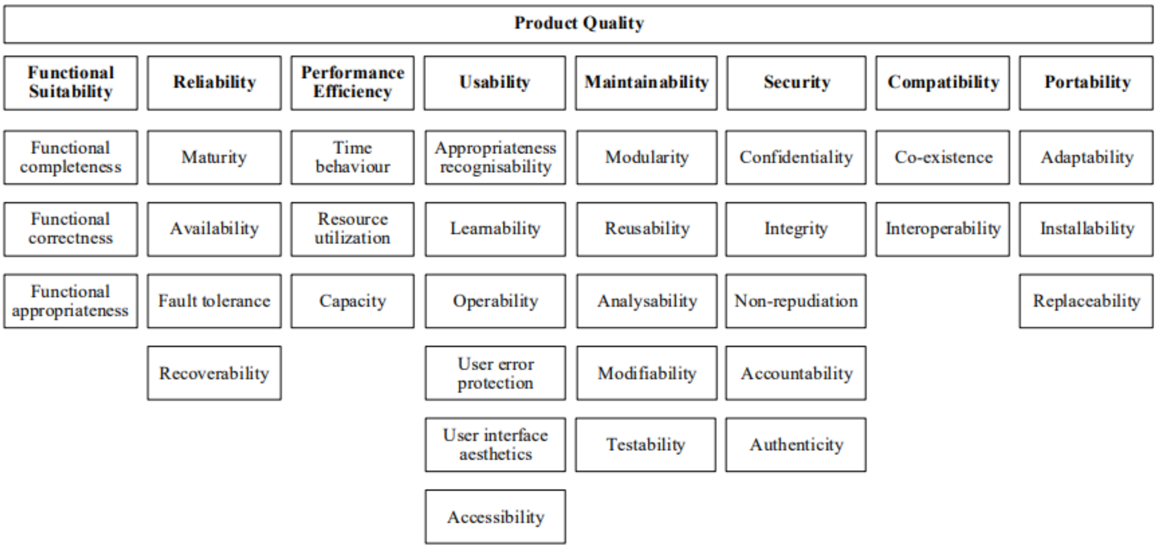
This section presents the results obtained by following the stages of the methodology. The process began with the partners' joint definition of the problem and characterization of the context. It then advanced to the mapping of internationally recognized software quality assurance domains, followed by iterative cycles of development, testing, and learning focused on the Web3 context. These steps culminated in the creation of the quality assurance guide designed to strengthen software robustness and ensure compliance in Web3 applications.

3.1. Stage 1 - Identifying the Problem, Characterizing the Context, and Involving the Partners

The discussions between the VOH.CoLAB and Exeedme teams led to a shared aim: to design a quality assurance guide tailored to the specific demands of blockchain gaming applications. Rather than offering general principles, the guide aims to present a structured framework organized into testing domains and subdomains. This structure serves as a practical bridge between the SQuaRE international guidelines and the realities of Web3 development, enabling teams to identify and define tests corresponding to each subdomain. By outlining actionable components, the guide helps developers and organizations implement targeted quality assurance practices that support both compliance with international standards and high-performance outcomes in blockchain-based gaming platforms. This mutual alignment provided a clear foundation for the execution of the remaining stages of this study.

3.2. Stage 2 - Map Relevant Quality Assurance Domains

The focused literature review of international standards highlighted the ISO/IEC 25000 SQuaRE series as the main reference for mapping quality assurance domains. SQuaRE is a series of standards developed by ISO/IEC (International Organization for Standardization/International Electrotechnical Commission) that focuses on defining and assessing the quality of systems and software (Febrero et al., 2016; Shtefan & Zaporozhets, 2021). The main objective of the SQuaRE standards is to provide a framework for evaluating software and system quality, ensuring that products meet quality requirements throughout their lifecycle. The international standards present two assessment models: product quality and quality-in-use, presented in Figure 2.



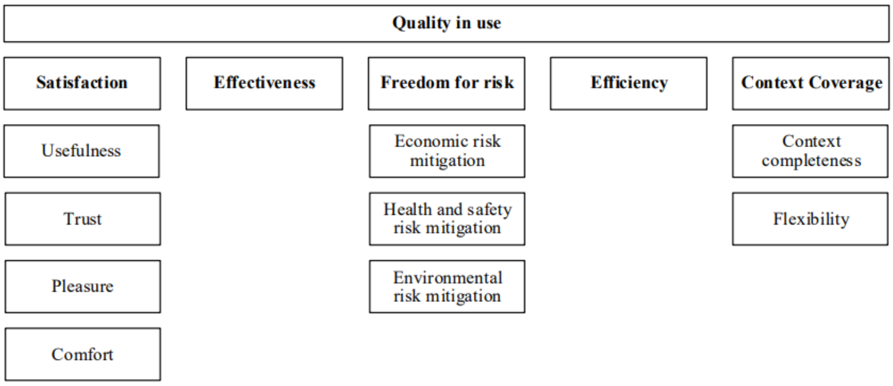


Figure 2. Characteristics and subcharacteristics of the Product Quality and Quality In Use Models Defined by the SQuaRE Standards (Febrero et al., 2016; Shtefan & Zaporozhets, 2021)

The product quality model defines eight characteristics: functional suitability, reliability, performance efficiency, usability, maintainability, security, compatibility, and portability, each subdivided into subcharacteristics that specify aspects of software behavior and value for end users. The quality-in-use model evaluates the impact of the product on stakeholders in real-use contexts, considering satisfaction, effectiveness, freedom from risk, efficiency, and context coverage. Together, these models provide a basis for defining requirements, generating evaluation measures, and supporting software quality assessment.

Based on this mapping, the teams reflected that, while comprehensive, the models were not always directly applicable to the context of decentralized, blockchain-based platforms. Several subcharacteristics required contextual adaptation to address the specificities of Web3 environments, such as smart contracts, decentralized architectures, and asset trading mechanisms. In addition, some re-structuring of the characteristics was deemed necessary so that the teams considered the structure was understandable. These adaptations were discussed in the subsequent iterative cycles of development, testing, and learning, so that teams agreed on a model that truly reflected the practical needs of blockchain-based gaming platforms.

3.3. Stages 3, 4, 5 - Iteration Cycles: Develop, Test and Learn

This first cycle of discussions and refinements led to the definition of the testing domains and subdomains most relevant for the Web3 environment, according to the involved teams. These domains and subdomains, presented in Table 1, aim to cover the widest possible range of functionalities to ensure that the system operates correctly, is secure, and meets business requirements. Thus, the objectives of each were carefully co-defined by the teams, drawing on the SQuaRE standards and on their expertise in the Web3 context. Following Table 1, the main aspects of software and infrastructure that need to be assessed and the main concerns raised in the collaborative discussions are presented, organised by domain.

Table 1. Test Domains and Subdomains and their Objectives.

Test Domain	Objective	Subdomains	Objectives
Functionality	Evaluate whether the platform meets the defined functional requirements.	Functional Completeness	Ensure the existence of the implementation of the expected features.
		Functional Correctness	Validate whether the results obtained within each implemented feature are as expected.
		Functional Appropriateness	Assess whether the provided functions are appropriate for the intended task.
Integration	Verify connectivity between the web platform and the blockchain network, ensuring seamless integration with external APIs (e.g., payment providers, authentication services) and proper data exchange.	Blockchain Integration	Verify that the communication between the Web3 software frontend and the blockchain is consistent, the execution of the programmed functions in the smart contracts produces the expected results and that these results are correctly reflected to the user.
		Integration with third-party APIs	Ensure seamless communication between the software and other APIs, namely from game platforms and other skin marketplaces.
		Payment API Integration	Ensure that the software handles payment responses correctly by only updating the blockchain when payments succeed, preventing changes if they fail and clearly informing the user throughout the process.
Security	Ensure the software's resilience against external attacks.	Smart contracts	Test smart contract code resilience to external attacks.
		Decentralized Apps	Test website resilience to external attacks.
Performance	Confirm that the software utilizes system resources efficiently.	Time Behaviour	Evaluate the speed of actions performed with a special focus on skin transactions.
		Resource Utilization	Assess the system's ability to perform its normal activities without consuming excessive resources.
		Capacity	Evaluate how the platform reacts under increasing loads of users and transactions, in order to identify potential bottlenecks.

Usability	Evaluate how easy and efficient it is for users to interact with the software.	Operability	Evaluate how easy it is to operate in the Web3 application and how this ease encourages users to stay on the platform.
		Navigation	Evaluate how easily users can navigate between different pages of the website, to determine whether their daily interactions are intuitive and free from major difficulties in understanding the steps needed to reach their goals.
		User responsiveness	Ensure a smooth experience where users feel that their activities on the platform are seamless and uninterrupted.
Portability	Assess the software's ability to operate across different environments and devices.	Adaptability	Test the software's compatibility across different browsers and devices.
		Substitutability	Evaluate how easily certain components can be replaced with components with other technologies or providers.
Recoverability	Measure the system's ability to restore data and recover from failures.	-	-
Resilience	Analyze the platform's capability to maintain operations during partial service disruptions.	-	-

3.3.1. Functionality domain

Functionality domain assesses whether the Web3 software defined functions provide everything required to complete the user's tasks and whether they do so accurately and correctly. Finally, it determines whether the offered functionalities are relevant to the software's intended use objectives. *Functionality* domain is divided into three subdomains. The *functional completeness* subdomain aims to ensure the implementation of the expected functionalities in a skin marketplace, such as user registration, login, purchasing, selling and listing skins, whereas the primary purpose of *functional correctness* subdomain is to validate whether the outcomes obtained within each implemented functionality are as expected. The *functional appropriateness* subdomain aims to assess whether the provided functions are suitable for the intended task.

3.3.2. Integration domain

Integration domain verifies the connectivity between the web platform and the blockchain network, ensuring seamless integration with external APIs (e.g., payment providers, authentication services) and proper data exchange. This verification allows the Web3 software to communicate with the blockchain to record and maintain the transaction history and ownership of skins as they are traded, ensuring greater transparency. The Integration domain is divided into three subdomains. The *Blockchain Integration* subdomain verifies that the communication between the marketplace API and the blockchain is consistent, that smart contract functions execute as intended, and that their results are accurately reflected in the frontend for the user. The *Integration with third-party APIs* subdomain focuses on validating the integration with third-party authentication providers, such as Steam in the case of gaming platforms, to streamline the login process and enhance security through features like two-factor authentication (2FA). Finally, the Payment API Integration subdomain assesses the integration with trusted payment services, such as PayPal, which supports smoother new user onboarding and promotes greater platform usage, since these providers already have established credibility and transparency within the community.

3.3.3. Security domain

Security testing on the platform is of great importance as it ensures the Web3 software resilience to external attacks that could compromise its operation. Responsibility for security is shared between the Web3 application and the underlying blockchain infrastructure, with each requiring targeted tests. For the Web3 software, it is necessary to safeguard all personal data of registered users, ensuring secure registration and login processes. For the blockchain, the primary focus of testing will be on smart contracts, given their central role in recording transactions conducted in the Web3 software. *Security* domain is divided into two subdomains: *Smart Contracts* and *Decentralized Apps*. One of the most critical components in the marketplace's security is testing the *Smart Contracts*, which are responsible for recording key information, such as skin ownership and market value. If these contracts are not regularly audited to identify potential bugs and areas for improvement, the marketplace could be exposed to a wide range of attacks. Potential security vulnerabilities in the marketplace's smart contracts include reentrancy attacks, integer overflow and underflow, return values, access control, front-running, denial of service attacks, uninitialized storage pointers, and timestamp dependency (Jiao et al., 2024). Additionally, as a *Decentralized App,* the Web3 software is susceptible to common web application attacks such as Cross-Site Scripting or SQL injection attacks. Together, these subdomains ensure components are secure, supporting trustworthy and robust Web3 software.

3.3.4. Performance domain

Performance domain refers to how effectively the software utilizes system resources. It is divided into three subdomains. *Time behaviour* subdomain tests should measure how fast the Web3 software performs actions (for e.g., skin transactions in the case of gaming platforms). Faster response times improve the user experience. However, due to the nature of blockchain, instant responses aren't always possible. It's important to also check the delays between the Web3 software, the blockchain, and other systems to find areas that can be optimized. When it comes to the *Resource Utilization* subdomain, blockchain systems can require significant resources to complete transactions or run smart contracts. Higher resource use leads to more energy consumption and higher gas fees. By evaluating gas fees in terms of energy use, we can consider how sustainable the platform is over time. Inefficient blockchains may cause unnecessary financial and environmental costs. The *Capacity* subdomain evaluates how the platform handles more users and transactions. It helps find performance bottlenecks so improvements can be made without disrupting normal use. Like time behavior tests, capacity testing should assess both the Web3 software and the blockchain, as each can face different challenges that may need unique solutions.

3.3.5. Usability domain

Usability domain aims to assess the overall user experience in the Web3 software and the ease of use of the platform. The clarity of the interfaces will be a crucial factor in the platform's impact on users, contributing to increased usage and user retention. User feedback will play a vital role in evaluating the Web3 software usability and will provide valuable insights for refining the visual design and available functionalities, with the goal of continuously enhancing the user experience. *Usability* domain is divided into three subdomains. *Operability* subdomain aims to evaluate how easy it is to operate in the Web3 software and how this ease encourages users to stay on the platform. *Navigation* subdomain aims to evaluate how easily users can move between different pages of the Web3 software, allowing for an analysis of whether everyday operations are intuitive and can be performed without significant interpretation difficulties. Finally, for a positive user experience, the platform must respond to user actions as promptly as possible, contributing to a seamless experience where users feel that activities on the platform are uninterrupted. The *User Responsiveness* subdomain assesses these aspects of the software utilization.

3.3.6. Portability domain

Portability domain refers to the Web3 software's ability to operate across different environments and devices. It is divided into two subdomains. *Adaptability* subdomain aims to evaluate the platform's compatibility across various browsers (Chrome, Firefox, Safari, etc.) and devices (desktop, mobile, tablet). *Substitutability* subdomain refers to the ease with which specific Web3 software components can be replaced by alternative solutions, to ensure these transitions can occur swiftly, cost-effectively, and without service interruptions, maintaining continuity for both users and systems. The ability to easily replace software components enables organizations to respond rapidly to technological changes or emerging market needs, ensuring that they are using the most suitable or up-to-date solutions.

3.3.7. Recoverability domain

Recoverability domain refers to the Web3 software ability to restore data and recover its state after failures. It is essential to ensure that the marketplace has effective backup and recovery procedures so that if a severe failure occurs that renders the system unavailable, it will be possible to restore the system to its original state without affecting users. It has no defined subdomains.

3.3.8. Resilience domain

Resilience domain aims to assess the Web3 software ability to continue operating during partial service interruptions. Similar to the recoverability testing domain, this domain has no defined subdomains.

3.3.9. The use of domains and subdomains in practice

Following the definition of testing domains and subdomains tailored to the Web3 context, the proposed structure was evaluated through its application in a second iteration cycle. In this phase, a set of 108 tests were systematically defined with Exeedme, aligned with the domains and subdomains and tailored to the requirements of their gaming platform.

The tests were obtained by following the methodology described in section 2. They were initially defined by VOH.CoLAB research team on the basis of research. These were subsequently reviewed by Exeedme, who provided feedback and valuable insights. This collaborative exchange not only validated the methodology but also guided further refinement, ensuring a more comprehensive and rigorous approach of the tests that can be applied in a Web3 platform. This process went through several iterative cycles, ultimately resulting in the tests presented in Table 2, which presents the distribution of these tests across the defined domains and subdomains.

These individual tests were not intended as standalone results, but rather as a demonstration of how the structured model can effectively guide the design of relevant and context-specific quality assurance activities.

Table 2. Distribution of tests by domain (and subdomain when applicable)

Domain	Subdomain (number of tests by subdomain)	Total number of tests
Functionality	Functional Completeness (11) Functional Correctness (13) Functional Appropriateness (10)	34
Integration	Blockchain Integration (9) Payment API Integration (4) Third-party API Integration (2)	15
Security	Smart Contracts (12) Decentralized Apps (8)	20
Performance	Time Behaviour(5) Capacity (4) Resource Utilization (2)	11
Usability	Navigation (5) Operability (5) User Responsiveness (2)	12
Portability	Adaptability (2) Substitutability (4)	6
Recoverability tests	-	3
Resilience tests	-	7

During the iteration cycles, both teams agreed that a critical component in safeguarding the security of a blockchain-based platform lies in the systematic testing of smart contracts, which are responsible for recording highly sensitive information, such as the ownership of skins and their market value. Without frequent audits aimed at identifying potential improvements and vulnerabilities in the code, these contracts could become a major source of risk, leaving the marketplace exposed to a wide spectrum of possible attacks. This spectrum has been analyzed and several tests were defined to cope with potential threats in the platform's smart contracts.

The set of tests were defined in the *Security* domain, namely in the *Smart Contracts* subdomain. They can prevent several attack types, such as the reentrancy attack, where a smart contract makes an external call to another contract before updating its own state. If the external contract calls back into the original contract during this process, the original contract's logic can be exploited, potentially allowing multiple withdrawals of funds before the state is correctly updated. Table 3 shows the tests defined for the *Smart Contracts* subdomain.

Table 3. Smart contracts tests

Test type	Description	Objective
Reentrancy	Simulate consecutive fund withdrawals through smart contract execution	Test the smart contracts' resilience to reentrancy attacks
	Simulate consecutive skin purchases through smart contract execution	
	Simulate consecutive skin sales through smart contract execution	
	Simulate consecutive skin trades through smart contract execution	
Overflow	Trigger overflow in possible integer variables of the smart contract	Ensure that the smart contract has no variables that can cause overflow
Underflow	Trigger underflow in possible integer variables of the smart contract	Ensure that the smart contract has no variables that can cause underflow
Return Values	Check that in smart contracts relying on external contracts, the return values are correct and being validated	Ensure the smart contract has no validation flaws that could cause unexpected results
Execution Authorization	Simulate unauthorized access by users trying to execute smart contracts	Ensure the smart contract can only be executed by authorized users
Front-running	Simulate execution of multiple transactions and verify if they are vulnerable to front-running	Ensure the smart contract has mechanisms to prevent attackers from front-running sensitive transactions
Denial of Service	Simulate execution of multiple transactions and monitor gas costs	Ensure the smart contract has mechanisms to prevent reaching the allowed gas limits

Uninitialized Pointers	Simulate smart contract execution and verify that variable initialization is well-defined	Ensure the smart contract has no pointer vulnerable to data replacement
Timestamp Dependency	Check that there are no variables dependent on timestamps	Avoid unfair transaction mining

Other tests were defined for evaluating the user experience, which is a relevant aspect of the success of a skin marketplace. For example, the *Navigation* subdomain tests defined in Table 4 aim to assess the software capabilities in this area by evaluating some of the generic functionalities of a skin marketplace and how the users respond to the developed frontend.

In practice, the subjects can be asked to complete the most important tasks while their performance (e.g. completion rate, time taken, number of errors) and subjective feedback were recorded. These tests aim to capture both the efficiency of the developed frontend and the intuitiveness of its design. They also assess if the platform matches the users expectations and introduces new features that improve user experience.

Although *Navigation* is used here as an example, similar procedures were applied to other functional areas to ensure a comprehensive evaluation of the overall user experience, such as *Operability* which focuses on measuring simplicity, clarity and ease-of-use, whereas *User Responsiveness* tests measure the users' perception of waiting time for different actions in the platform. The objective is to assess not only the actual system response time but also how this duration is subjectively perceived by users. To this end, the effectiveness of progress indicators and waiting messages is tested, examining whether they help reduce perceived frustration and increase the sense of transparency during the process, again by gathering the subjective feedback of users.

Table 4. *Navigation* subdomain tests

Type	Description	Objective
Navigation	Ask users to list an item for sale	Evaluate whether users can navigate the marketplace easily and quickly find the features.
	Ask users to perform tasks such as finding a specific skin	
	Ask users to access the transaction history	
	Ask users to access their profile	
	Ask users to access notifications	

4. Discussion

This study proposed a structured quality assurance guide for blockchain-based applications. Departing from the ISO/IEC SQuaRE model, the guide defines eight testing domains and 16 associated subdomains tailored to the unique requirements of Web3 systems. This structure was tested in practice for the case of a blockchain-based marketplace for virtual gaming assets, for which 108 tests were defined within the domains and subdomains. The key contribution of this study lies not in the 108 individual tests defined, but in the underlying structure that supports systematic, context-specific quality assurance practices (Precht et al., n.d.; J. Xu et al., 2020). This model enables both developers and quality assurance professionals to bridge the gap between high-level international standards and the operational realities of decentralized application development.

4.1 Defined Domains and Subdomains

The defined domains reflect a balance between user-oriented quality concerns and technical development needs. Domains and subdomains such as *Usability* and *Functional Appropriateness* (in *Functionality*) focus on the end-user experience, ensuring the software provides a satisfactory experience, conveys trust, and meets the main needs of the users. These concerns align with findings that user satisfaction and perceived quality are central to adoption and sustained use of digital platforms (Bevan, 2009; Kitchenham & Pfleeger, 1996). Although user experience is a concern for every platform, and not exclusive to Web3 platforms, there is still a gap in expectations between users that typically use Web2 and Web3 platforms (Hou, 2024). Therefore, the tests designed for evaluating a Web3 platform user experience should be tailored to the specific requirements of its users, such as transaction transparency and system control.

The work of (Vacca et al., 2024) highlights that having functional evaluation framework may contribute to ensuring contract quality prior to deployment, one of the main components tested *Functional Correctness* (in *Functionality* domain). As (Vacca et al., 2024) suggests these tests may be conducted by using tools that assess the smart contracts code. Additionally, the proposed tests for *Functional Appropriateness* subdomain include frontend validation to determine whether the outcomes of marketplace transactions display all relevant information.

Despite not considering user feedback, implementing tests that aim to improve the marketplace's performance will contribute to a faster, more efficient, and smoother marketplace experience. The *Performance* domain aims to evaluate the latency of the system, from the smart contracts time response to the overall time the application takes to complete a transaction and register it in the blockchain. As referred in (Zhang et al., 2021) the gas price is a key parameter controlled by users, so *Resource Utilization* subdomain tests have an important role in maximizing the marketplace's user adoption.

The *Security* domain emerged as particularly vital, given the trust-sensitive nature of blockchain-based marketplaces and the high prevalence of attacks targeting smart contracts (Atzei et al., 2017). Quality assurance for smart contracts is not trivial as errors are immutable post-deployment and can have irreversible financial consequences.

In addition to verifying contract behavior, the model includes tests for common web vulnerabilities and GDPR compliance, acknowledging the increasing importance of privacy, users' trust and regulatory alignment in decentralized systems (Finck, 2018). To assess GDPR compliance in a blockchain-based marketplace, the tests may examine how personal data is collected, stored, and processed, and if it ensures that user rights such as access and erasure are respected despite the immutability of the ledger (Belen-Saglam et al., 2023). Another important aspect would be the Governance quality assessment. This can be evaluated by reviewing the clarity of decision-making structures, transparency of policies, and the presence of accountability mechanisms such as audits or dispute resolution processes (Ibrahimy et al., 2024). The tests raised by both teams focus on document analysis (whether they exist and clarity of the documentation), technical audits of smart contracts and off-chain infrastructure, compliance checklists against legal and governance standards, and user studies to evaluate the accessibility and effectiveness of rights and decision-making processes.

Similar to the *Functionality* domain, the *Portability* domain has aspects that can be evaluated from the user's perspective, such as the *Adaptability* subdomain. Additionally, the *Replaceability* subdomain may also impact development teams. The ability to develop code that is adaptable to various technologies and reusable contributes to greater flexibility in blockchain ecosystems, where rapid technological evolution is the norm (X. Xu et al., 2019).

At last, although domains like *Resilience* and *Recovery* are not directly user-facing, they improve maintainability, uptime, and failure response – factors that significantly contribute to user trust and satisfaction in the long term (Wagner & Deissenboeck, 2007). The tests suggest the simulation of blockchain network failures or timeouts to assess if the marketplace does not compromise important data nor the subsequent pending transactions.

4.2 Practical Implications of the findings

The practical testing of the proposed structure was achieved through its application to a real blockchain gaming platform. The 108 defined tests were not meant to be exhaustive or universal; rather, they illustrate how the structure guides meaningful quality assurance activity within a specific context. This operationalization addresses concerns raised in the literature that international standards like ISO/IEC SQuaRE, while theoretically robust, often lack practical guidance and are difficult to adopt in agile or domain-specific environments (Gordieiev et al., 2024; Suryn et al., 2003).

Theoretically, this study contributes to the broader effort of adapting classical quality assurance models to emerging technologies. It shows how high-level constructs such as *Functional Suitability* or *Compatibility* can be translated into blockchain-based environments. This supports the idea that models like SQuaRE can retain relevance if implemented through context-aware, modular, and testable structures ("ISO/IEC 25010," 2023; Kitchenham & Pfleeger, 1996).

In practice, the model supports integration into modern software engineering pipelines, such as continuous deployment. Many of the tests, particularly those in the *Functionality*, *Security*, and

Resilience domains, can be implemented as automated unit tests to be used, for instance, in the context of releasing new features, where they would be automatically executed whenever a release occurs. This not only enables early error detection and control but also fosters modular architecture and maintainable code, a recognized benefit of test-driven development practices (Janzen & Saiedian, 2005). The *Modularity* test presented under *Resilience* domain, for example, is directly aligned with these software engineering principles.

Beyond the technical implementation, it is important to situate quality assurance within the broader systemic debates surrounding Web3. As highlighted in recent work (Balduf et al., 2023; Esposito et al., 2025; Hawes, 2023), the promise of decentralization is often undermined by infrastructural centralization, for example reliance on cloud providers or concentration of nodes in limited geographies, and by governance asymmetries that concentrate decision-making power in the hands of a few actors. These dynamics directly influence resilience and security, while also shaping how quality and trust are understood within decentralized ecosystems. Ensuring a shared understanding of these dimensions, alongside transparent governance mechanisms and awareness of infrastructural dependencies, is essential to connect software quality assurance with sustainability.

The objective of this framework is that it can be generalized into other Web3 softwares. While the methodology provides a domain-agnostic process for co-developing quality assurance approaches in Web3, the framework organizes Web3 domains and subdomains that could be applicable to other cases. For instance, if one would apply the framework to DeFi, the domain and subdomain structure could be maintained, but the subdomain objectives and examples shift to DeFi risks and features (Ma et al., 2023; Zhou et al., 2023). *Functionality* would test the correctness and appropriateness of smart contract logic for swaps or loans, Integration would validate oracles, wallets, and payment rails. *Security* would focus on resistance to exploits like flash-loan attacks. *Performance* and *Usability* would then assess transaction speed, cost efficiency, and user experience across different wallets and networks, similarly to what is already tested in this framework.

4.3 Strengths and limitations of the study

A key strength of this study lies in its co-design methodology, involving collaboration between researchers and industry actors. This ensured the resulting guide is grounded in both theory and the practical constraints of software teams working in Web3 environments. However, the study is not without limitations. The framework was only applied to one use case, a virtual asset trading platform, so broader applicability remains to be evaluated. Moreover, while the defined tests were developed and refined in collaboration with stakeholders, the defined tests have not yet been deployed and assessed in a live production setting.

5. Conclusions

This study provides a structured and actionable quality assurance guide for blockchain-based applications, that supports teams in identifying and defining tests systematically. We demonstrate that it is possible to use international standards to evaluate and enhance blockchain applications systematically. This research contributes to bridging the gap between theoretical frameworks and practical implementation, fostering the development of high-quality Web3 applications and assisting the certifying path of Web3 software.

Future research should focus on expanding and refining the proposed domains and subdomains based on the practical insights of Web3 software development teams. Applying the framework to different contexts, such as decentralized finance (DeFi) or NFT platforms, could reveal additional quality dimensions or require new adaptations. Furthermore, estimating the effort required to implement tests in each subdomain would enhance the model's applicability by supporting test prioritization and strategic planning, particularly for teams with limited resources.

References

Atzei, N., Bartoletti, M., & Cimoli, T. (2017). *A survey of attacks on Ethereum smart contracts*.

Balduf, L., Korczyński, M., Ascigil, O., Keizer, N. V., Pavlou, G., Scheuermann, B., & Król, M. (2023). The Cloud Strikes Back: Investigating the Decentralization of IPFS. *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC*, *1*, 391–405. https://doi.org/10.1145/3618257.3624797

Belen-Saglam, R., Altuncu, E., Lu, Y., & Li, S. (2023). A systematic literature review of the tension between the GDPR and public blockchain systems. *Blockchain: Research and Applications*, *4*(2), 100129. https://doi.org/10.1016/J.BCRA.2023.100129

Bevan, N. (2009). Extending Quality in Use to Provide a Framework for Usability Measurement. In M. Kurosu (Ed.), *Human Centered Design* (pp. 13–22). Springer. https://doi.org/10.1007/978-3-642-02806-9_2

BlockchainPT. (2025). https://blockchain.void.pt/

CSGO & CS2 Item Economy: 2023 Overview and 2024 Thoughts. (2024). In *Swap.gg*. https://swap.gg/blog/csgo-item-economy-2023-overview

Esposito, M., Tse, T., & Goh, D. (2025). Decentralizing governance: exploring the dynamics and challenges of digital commons and DAOs. *Frontiers in Blockchain*, *8*, 1538227. https://doi.org/10.3389/FBLOC.2025.1538227/BIBTEX

Febrero, F., Calero, C., & Ángeles Moraga, M. (2016). Software reliability modeling based on ISO/IEC SQuaRE. *Information and Software Technology*, *70*, 18–29. https://doi.org/10.1016/j.infsof.2015.09.006

Finck, M. (2018). Blockchains and Data Protection in the European Union. *European Data Protection Law Review*, *4*(1), 17–35. https://doi.org/10.21552/edpl/2018/1/6

Gordieiev, O., Rainer, A., Kharchenko, V., Pishchukhina, O., & Gordieieva, D. (2024). A Unified Approach to the Development of Technology-Based Software Quality Models on the Example of Blockchain Systems. *IEEE Access*, *12*, 118875–118889. https://doi.org/10.1109/ACCESS.2024.3448271

Hawes, B. (2023). *Web3: The Promise & the Reality*.

Hevner, A., & Park, J. (2004). *Design Science in Information Systems Research*. https://www.researchgate.net/publication/201168946

Hossain Faruk, M. J., Raya, P., Siam, M. K., Cheng, J. Q., Shahriar, H., Cuzzocrea, A., & Bringas, P. G. (2024). A Systematic Literature Review of Decentralized Applications in Web3: Identifying Challenges and Opportunities for Blockchain Developers. *Proceedings - 2024 IEEE International Conference on Big Data, BigData 2024*, 6240–6249.

https://doi.org/10.1109/BIGDATA62323.2024.10826066

Hou, C.-C. (2024). *Optimizing User Experience of Decentralized Applications for Web2 and Web3 Users - Case Anonymous Decentralized Social Platform*. https://aaltodoc.aalto.fi/handle/123456789/133628

Ibrahimy, M. M., Norta, A., & Normak, P. (2024). Blockchain-based governance models supporting corruption-transparency: A systematic literature review. *Blockchain: Research and Applications*, *5*(2), 100186. https://doi.org/10.1016/J.BCRA.2023.100186

ISO/IEC 25010:2023. (2023). In *ISO*. https://www.iso.org/standard/78176.html

Janzen, D., & Saiedian, H. (2005). Test-driven development concepts, taxonomy, and future direction. *Computer*, *38*(9), 43–50. https://doi.org/10.1109/MC.2005.314

Jiao, T., Xu, Z., Qi, M., Wen, S., Xiang, Y., & Nan, G. (2024). A Survey of Ethereum Smart Contract Security: Attacks and Detection. *Distributed Ledger Technologies: Research and Practice*, *3*(3). https://doi.org/10.1145/3643895

Kitchenham, B., & Pfleeger, S. L. (1996). Software quality: the elusive target [special issues section]. *IEEE Software*, *13*(1), 12–21. https://doi.org/10.1109/52.476281

Londral, A., Azevedo, S., Dias, P., Ramos, C., Santos, J., Martins, F., Silva, R., Semedo, H., Vital, C., Gualdino, A., Falcão, J., Lapão, L. V., Coelho, P., & Fragata, J. G. (2022). Developing and validating high-value patient digital follow-up services: a pilot study in cardiac surgery. *BMC Health Services Research*, *22*(1). https://doi.org/10.1186/s12913-022-08073-4

Ma, W., Zhu, C., Liu, Y., Xie, X., & Li, Y. (2023). A Comprehensive Study of Governance Issues in Decentralized Finance Applications. *ACM Transactions on Software Engineering and Methodology*, *1*. https://doi.org/10.1145/3717062

Mubarkoot, M., Altmann, J., Rasti-Barzoki, M., Egger, B., & Lee, H. (2023). Software Compliance Requirements, Factors, and Policies: A Systematic Literature Review. *Computers & Security*, *124*, 102985. https://doi.org/10.1016/J.COSE.2022.102985

NFT - Worldwide Statista Market Forecast. (n.d.). In *Statista*. https://www.statista.com/outlook/fmo/digital-assets/nft/worldwide

NFT statistics in 2024: Growth trends and outlook Kraken. (n.d.). https://www.kraken.com/pt-br/learn/nft-statistics

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Precht, H., Wunderlich, S., & Marx Gómez, J. (n.d.). *Applying Software Quality Criteria to Blockchain Applications: A Criteria Catalog*. https://hdl.handle.net/10125/64511

Shtefan, N., & Zaporozhets, O. (2021). Software quality model based on SQuaRE standards. *Radiotekhnika*, *207*, 159–165. https://doi.org/10.30837/rt.2021.4.207.17

Suryn, W., Abran, A., & April, A. (2003). *ISO/IEC SQuaRE. The second generation of standards for software product quality*.

Tsuda, N., Washizaki, H., Honda, K., Nakai, H., Fukazawa, Y., Azuma, M., Komiyama, T., Nakano, T., Suzuki, H., Morita, S., Kojima, K., & Hando, A. (2019). WSQF: Comprehensive Software Quality Evaluation Framework and Benchmark Based on SQuaRE. *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, 312–321. https://doi.org/10.1109/ICSE-SEIP.2019.00045

Vacca, A., Fredella, M., Di Sorbo, A., Visaggio, C. A., & Piattini, M. (2024). Functional suitability assessment of smart contracts: A survey and first proposal. *Journal of Software: Evolution and Process*, *36*(7), e2636. https://doi.org/10.1002/SMR.2636

Wagner, S., & Deissenboeck, F. (2007). An Integrated Approach to Quality Modelling. *Fifth International Workshop on Software Quality (WoSQ'07: ICSE Workshops 2007)*, 1. https://doi.org/10.1109/WOSQ.2007.3

Xu, J., Zhang, H., & Gong, J. (2020). *Analysis for Blockchain Application Quality*.

Xu, X., Weber, I., & Staples, M. (2019). Architecture for Blockchain Applications. *Architecture for Blockchain Applications*. https://doi.org/10.1007/978-3-030-03035-3/COVER

Zhang, L., Lee, B., Ye, Y., & Qiao, Y. (2021, April 19). Evaluation of Ethereum End-To-end Transaction Latency. *2021 11th IFIP International Conference on New Technologies, Mobility and Security, NTMS 2021*. https://doi.org/10.1109/NTMS49979.2021.9432676

Zhou, L., Xiong, X., Ernstberger, J., Chaliasos, S., Wang, Z., Wang, Y., Qin, K., Wattenhofer, R., Song, D., & Gervais, A. (2023). *SoK: Decentralized Finance (DeFi) Attacks*. https://doi.org/10.1109/SP46215.2023.00180