information

MDPI

Article

OSSAPTestingPlus: A Blockchain-Based Collaborative Framework for Enhancing Trust and Integrity in Distributed Agile Testing of Archaeological Photogrammetry Open-Source Software

Omer Aziz [1], Muhammad Shoaib Farooq [1], Junaid Nasir Qureshi [2], Muhammad Faraz Manzoor [1] and Momina Shaheen [3,*]

[1] School of System and Technology, Department of Artificial Intelligence, University of Management and Technology, Lahore 54000, Pakistan; f2018288006@umt.edu.pk (O.A.); shoaib.farooq@umt.edu.pk (M.S.F.); faraz.manzoor@umt.edu.pk (M.F.M.)
[2] Department of Computer Science, Bahria University, BULC, Lahore 54600, Pakistan; jnqureshi.bulc@bahria.edu.pk
[3] School of Arts, Humanities and Social Sciences, University of Roehampton, London SW15 5PU, UK
* Correspondence: momina.shaheen@roehampton.ac.uk

Abstract

(1) Background: A blockchain-based framework for distributed agile Open-Source Software for Archaeological Photogrammetry (OSSAP) testing life cycle is an innovative approach that uses blockchain technology to optimize the Open-Source Software for Archaeological Photogrammetry process. Previously, various methods have been employed to address communication and collaboration challenges in Open-Source Software for Archaeological Photogrammetry, but they were inadequate in aspects such as trust, traceability, and security. Additionally, a significant cause of project failure was the non-completion of unit testing by developers, leading to delayed testing. (2) Methods: This article discusses the integration of blockchain technology in Open-Source Software for Archaeological Photogrammetry and resolves critical concerns related to transparency, trust, coordination, testing and communication. A novel approach is proposed based on a blockchain framework named Open-Source Software for Archaeological Photogrammetry Testing-Plus. (3) Results: The Open-Source Software for Archaeological Photogrammetry Testing-Plus framework utilizes blockchain technology to provide a secure and transparent platform for acceptance testing and payment verification. Moreover, by leveraging smart contracts on a private Ethereum blockchain, Open-Source Software for Archaeological Photogrammetry Testing-Plus ensures that both the testing team and the development team are working towards a common goal and are compensated fairly for their contributions. (4) Conclusions: The experimental results conclusively show that this innovative approach substantially improves transparency, trust, coordination, testing and communication and provides security for both the testing team and the development team engaged in the distributed agile Open-Source Software for Archaeological Photogrammetry (Open-Source Software for Archaeological Photogrammetry) testing life cycle.

Keywords: OSSAPTestingPlus; OSSAP; open-source software for archaeological photogrammetry; blockchain technology; trust; transparency

check for updates

Academic Editor: Ge Yu

Received: 17 September 2025
Revised: 8 November 2025
Accepted: 9 November 2025
Published: 17 November 2025

Citation: Aziz, O.; Farooq, M.S.; Qureshi, J.N.; Manzoor, M.F.; Shaheen, M. OSSAPTestingPlus: A Blockchain-Based Collaborative Framework for Enhancing Trust and Integrity in Distributed Agile Testing of Archaeological Photogrammetry Open-Source Software. *Information* **2025**, *16*, 992. https://doi.org/10.3390/info16110992

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

As the base technology for cryptocurrencies like Bitcoin, blockchain technology has come to be recognized for its potential to revolutionize several industries outside of banking. A blockchain, at its heart, is a decentralized, distributed ledger that securely, transparently, and immutably logs transactions [1]. A chain of blocks is created by connecting every transaction, or "block", cryptographically to the one before it. Without the use of middlemen or centralized agencies, its decentralized design ensures participant confidence and transparency.

Open-Source Software for Archaeological Photogrammetry (OSSAP) entails a thorough analysis of a software application or system to detect flaws, problems, or vulnerabilities. Prior works claim that it includes several tasks such as the development of test cases, execution of tests, defect tracking, and reporting. The performance, stability, and safety of the software can be validated through extensive testing, giving quality assurance teams and software engineers the certainty that it satisfies user expectations [2].

A strong quality assurance (QA) framework is needed in the dynamic field of Open-Source Software for Archaeological Photogrammetry (OSSAP), where accuracy and dependability are critical. In recognition of this requirement, an extensive testing protocol has been developed, leveraging blockchain technology to enhance the reliability of OSSAP. Using self-executing protocols known as Smart Contracts, this novel technology automates the process of assigning Test Cases to QA Engineers according to present standards. Fairness and openness in job distribution are assured by this decentralized allocation procedure, creating an atmosphere that is equitable for all parties involved [3]. Furthermore, Smart Contracts go beyond simple assignment by automatically updating the state of Test Cases concerning functional requirements. These status updates are ensured by utilizing the tamper-resistant features of the blockchain. Furthermore, Smart Contracts go beyond simple assignment by automatically updating the state of Test Cases concerning functional requirements. By utilizing the innate tamper-resistant characteristic of the blockchain, these status updates ensure a verifiable and dependable outcome, hence augmenting the testing process's integrity [2]. This framework fosters a transparent and responsible testing procedure that is beneficial to the growth of archaeological photogrammetry by reducing administrative overhead and improving overall reliability through the smooth integration of Smart Contracts into the OSSAP testing workflow. The blockchain's tamper-resistant properties [4].

Even though OSSAP plays a crucial role, traditional techniques frequently run into several issues that reduce the efficacy and effectiveness of the testing process [5]. These difficulties include a lack of trust, problems with coordination, gaps in communication, and security worries. Due to the role of numerous stakeholders, including developers, managers of projects, and clients, communication gaps can develop, resulting in misconceptions and lags in addressing crucial issues. When various teams operate in isolation from one another, coordination problems can arise, leading to ineffective test planning, insufficient coverage, and duplication of effort [5,6]. For organizations and end users, the shortcomings of traditional OSSAP approaches can have serious ramifications. Unfound flaws and vulnerabilities can have expensive repercussions, including software errors, system failures, data breaches, and monetary losses. Inadequate testing can also lead to bad user experiences, irritation, and a decline in confidence in the product or the company that created it [7]. If word of negative experiences and inferior products spreads via user evaluations and word-of-mouth, businesses' reputations may suffer. As the base technology for cryptocurrencies like Bitcoin, blockchain technology has come to be recognized for its potential to revolutionize several industries outside of banking. A blockchain, at its heart, is a decentralized, distributed ledger that securely, transparently, and immutably logs

transactions. A chain of blocks is created by connecting every transaction, or "block," cryptographically to the one before it. Without the use of middlemen or centralized agencies, its decentralized design ensures participant confidence and transparency.

Traditional OSSAP procedures have difficulties, which can be solved by the special advantages provided by blockchain technology. A foolproof and fair record of all testing actions, including test plans, test cases, test results, and communication, is first provided by blockchain [8]. Blockchain's immutability guarantees that information recorded once cannot be changed or removed, giving a trustworthy and auditable record of testing procedures. Transparency may increase stakeholder trust and facilitate traceability, making it simpler to identify the root causes of flaws and better upcoming testing procedures. Whilst blockchain technology initially sprang to attention as a result of its connection to cryptocurrencies, it has since developed to provide safe and decentralized solutions in several different industries. Supply chain management, healthcare, banking, and logistics are just a few of the sectors looking into how blockchain technology could improve their operations' efficiency, security, and transparency. Similar to this, incorporating blockchain technology into OSSAP offers a chance to get beyond the drawbacks of conventional approaches and streamline the testing procedure.

In the field of archaeological documentation and preservation, photogrammetry has emerged as a powerful technique for reconstructing 3D models of heritage artifacts and excavation sites. The open-source software ecosystem has enabled the development of various photogrammetry tools, fostering collaboration among archaeologists, developers, and researchers. However, ensuring the reliability, security, and traceability of these tools, especially within distributed agile development environments, presents several challenges. Integrating blockchain technology into the distributed agile testing lifecycle offers a promising approach to address issues of transparency, trust, and tamper proof documentation.

1.1. Background and Motivation

Agile methodologies have revolutionized software development through iterative cycles, rapid feedback, and close collaboration between stakeholders. When development teams are geographically dispersed, agile becomes distributed agile relying on online collaboration tools and asynchronous communication. Open-source photogrammetry projects often fall into this category. Testing these systems involves validating geometry reconstruction, texture accuracy, processing performance, and usability. Blockchain can add a verifiable audit trail of testing artifacts, test case execution, developer contributions, and bug resolutions. The benefits of blockchain in agile testing include immutable storage of test logs and audit trails, decentralized trust model for contributions, smart contract enforcement of testing standards, real-time transparency for external stakeholders, and improved reproducibility and accountability

Combining blockchain with distributed agile testing presents a robust and transparent approach for developing and maintaining archaeological photogrammetry software. It empowers contributors with verifiable records, ensures integrity in test practices, and supports reliable preservation of digital heritage tools. This synergy is particularly suited to the open-source paradigm where collaboration and trust are foundational.

The intersection of archaeology, open-source software, agile methodologies, and blockchain technology offers a compelling vision for the future of digital cultural heritage. Archaeological photogrammetry, a non-invasive technique that transforms photographs into precise 3D models of artifacts and sites, has been significantly democratized by open-source software. However, ensuring the reliability and quality of these complex tools, especially when developed and tested by geographically dispersed teams, presents unique

challenges. This essay explores how blockchain technology can revolutionize distributed agile testing for archaeological photogrammetry open-source software, enhancing transparency, trust, and efficiency across the entire development lifecycle.

Archaeological photogrammetry relies on capturing multiple overlapping images of an object or site and processing them through specialized software to generate highly detailed 3D models. These models are invaluable for documentation, analysis, virtual reconstruction, and public outreach. The open-source movement has profoundly impacted this field, providing powerful, accessible, and customizable tools. Their open nature fosters community contributions, allowing archaeologists and developers to adapt the software to specific research needs, share improvements, and collaboratively address bugs. This collaborative spirit aligns perfectly with agile development principles, which prioritize iterative development, continuous feedback, and adaptability to change.

The Open-Source Software for Archaeological Photogrammetry Testing-Plus (OSSAPTestingPlus) framework is a cutting-edge strategy that uses blockchain technology to streamline the life cycle of OSSAP. It tries to overcome major issues with traditional testing approaches' fundamental lack of coordination, transparency, and trust [9]. OSSAPTestingPlus integrates blockchain technology to offer an open and transparent platform for payment verification and acceptance testing, fostering productive communication between development teams, testing teams, and other players.

The main goals of the OSSAPTestingPlus framework are to improve OSSAP processes' effectiveness and efficiency, collaboration and communication among stakeholders, fair compensation for testing efforts, and the provision of a reliable mechanism for validation and tracking. The use of blockchain technology for safe and fair recording of testing operations, smart contracts for automated and just compensation, and real-time testing progress tracking are some of OSSAPTestingPlus' core features. Blockchain technology integration throughout all phases of the OSSAP life cycle is covered by OSSAPTestingPlus. Blockchain ensures that all testing activities are documented, timestamped, and kept in a secure and decentralized manner, from test planning to the development of test cases, execution, and tracking of defects. Using smart contracts makes automated payment processes easier and ensures that the testing team and other parties involved are fairly compensated. Figure 1 shows the testing life cycle for agile development processes.



Figure 1. Testing Life Cycle for Agile Development Process.

To sum up, the use of blockchain technology in OSSAP is a huge step forward in tackling pressing issues and streamlining the testing procedure. Traditional approaches have had trouble with coordination problems, communication problems, and trust concerns, which has resulted in less-than-ideal testing results. However, the OSSAPTestingPlus frame-

work, which was put forth in this study, makes use of blockchain technology's advantages to improve OSSAP's, transparency, coordination, and communication. OSSAPTestingPlus assures fair compensation and fosters effective collaboration between testing teams, development teams, and other stakeholders by offering a safe and fair network for acceptance testing and payment verification. To improve OSSAP procedures and the general quality and dependability of software applications, this research study covers the main features, advantages, and execution considerations of Testing Plus. Figure 2 depicts the Blockchain Features in OSSAPTestingPlus.

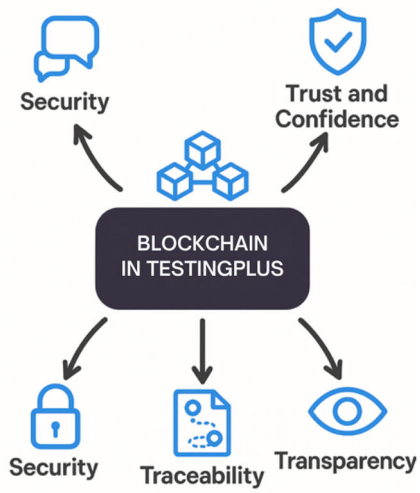


Figure 2. Blockchain Features in OSSAPTestingPlus framework.

1.2. Comparative Novelty and Positioning of OSSAPTestingPlus

In contrast to existing blockchain-based Quality Assurance (QA) and testing frameworks such as SmartBlock-SDN, AuditChain, and TrustTest, the proposed OSSAPTestingPlus framework introduces a dual innovation layer that directly addresses the gaps in distributed agile testing for open-source photogrammetry software.

First, OSSAPTestingPlus operationalizes automated acceptance testing through private-Ethereum smart contracts that execute real-time validation of functional requirements (FRs). Each testing interaction—ranging from test-case creation to acceptance confirmation—is cryptographically logged on a permissioned blockchain network. This ensures that test results are immutable, auditable, and transparently linked to their respective contributors. Unlike SmartBlock-SDN, which primarily manages IoT resource transactions, OSSAPTestingPlus directly embeds blockchain into the testing life-cycle itself, not merely as a post-deployment verification layer.

Second, the framework integrates an incentive-based compensation mechanism governed entirely by smart contracts. These contracts automatically release payments to QA engineers upon successful test verification, enforcing fairness and accountability across geographically distributed teams. While AuditChain and TrustTest emphasize audit trail generation and decentralized compliance, they do not include real-time payment enforcement or mutual agreement execution between developers and testers. OSSAPTestingPlus therefore extends blockchain utility beyond record-keeping into active workflow automation and human-to-human trust mediation within the testing ecosystem.

Furthermore, OSSAPTestingPlus is purpose-built for Open-Source Software for Archaeological Photogrammetry (OSSAP), a domain requiring verifiable traceability of both digital heritage data and software testing artifacts. The framework's private Ethereum configuration ensures lower latency, privacy control, and cost-efficient gas usage compared with public-chain implementations typically used in decentralized DevOps pipelines.

Through this architecture, OSSAPTestingPlus bridges the methodological gap between distributed agile testing and blockchain-driven accountability, offering measurable improvements in transparency, fairness, and coordination.

The novelty of OSSAPTestingPlus thus lies not merely in adopting blockchain for QA, but in embedding decentralized validation, compensation, and provenance tracking as integral testing functions. This holistic integration transforms blockchain from a passive audit ledger into an active participant in the OSSAP testing workflow, establishing a new paradigm for trust-centric, performance-optimized, and incentive-aligned agile testing in digital heritage applications.

- In-lifecycle enforcement vs. post hoc audit OSSAPTestingPlus embeds acceptance criteria execution and payment release inside the testing lifecycle (private Ethereum smart contracts), rather than using a ledger only for logging/audit after CI concludes.
- Incentive-aligned testing We couple verifiable test pass events with automatic, on-chain compensation, addressing the long-standing issue of missed tests in distributed OSS (no-pass $\Rightarrow$ no-pay).
- Domain-tailored provenance for OSSAP We model photogrammetry-specific artifacts (e.g., geometry/texture outputs, reconstruction KPIs) and their hashes, enabling traceability demanded in digital heritage workflows, with private-chain privacy and low-latency confirmations.

We also broaden experiments with baseline systems (audit-only ledger, assignment-only ledger, centralized CI) and report novel KPIs (payout latency, dispute rate/time, provenance verification time, on-time test completion). These additions clarify novelty boundaries and practical impact.

1.3. Research Questions and Hypotheses

Building upon the comparative positioning of OSSAPTestingPlus, this study formulates clear research questions and testable hypotheses to assess the framework's effectiveness in improving the distributed agile testing life cycle for Open-Source Software for Archaeological Photogrammetry (OSSAP).

The aim of overarching is to determine whether blockchain-enabled automation can measurably enhance trust, transparency, and efficiency in decentralized quality assurance workflows. Table 1 is designed to elaborate the research questions related to the article main focus. Table 2 is designed to elaborate the required hypothesis for the proposed framework.

Table 1. Research Questions for OSSAPTestingPlus framework.

Research Questions (RQs)	
RQ1	How does the integration of blockchain-based smart contracts influence trust, transparency, and traceability among distributed OSSAP testing teams compared with traditional agile testing approaches?
RQ2	Can OSSAPTestingPlus improve the operational efficiency of distributed agile testing—in terms of reduced transaction latency, higher throughput, and lower coordination overhead—without compromising data integrity?
RQ3	Does the implementation of automated, incentive-driven compensation mechanisms through smart contracts increase tester motivation and fairness perception in distributed OSSAP environments?

To empirically address these questions, the following hypotheses were established.

Table 2. Hypotheses for OSSAPTestingPlus framework (H1, H2, H3).

H1 (Transparency & Traceability)	Implementing OSSAPTestingPlus significantly improves transparency and traceability of testing activities compared to traditional OSSAP agile testing methods.	Metric Audit completeness, number of verifiable blockchain records, stakeholder satisfaction score.
H2 (Efficiency & Performance)	OSSAPTestingPlus reduces average transaction latency and improves throughput efficiency by at least 20% relative to non-blockchain agile workflows.	Metric Mean transaction response time (s), average throughput (transactions/min), gas consumption (ETH).
H3 (Trust & Fair Compensation)	Smart-contract-based automated acceptance testing increases stakeholder trust and ensures equitable payment enforcement between developers and testers.	Metric Payment dispute rate, trust index from post-test surveys, compensation accuracy (%).

1.4. Operationalization and Evaluation

These hypotheses are validated through controlled experiments conducted on a private Ethereum network configured with five nodes, Intel i7 @ 3.6 GHz, 16 GB RAM, and local peer-to-peer communication. Performance data were collected using Postman API scripts and OpenZeppelin security-enhanced smart contracts, evaluating metrics such as transaction latency, throughput, gas usage, and role-based access efficiency. Quantitative measures were complemented by qualitative feedback from simulated QA-developer teams to assess perceived fairness, trust, and coordination improvements.

In the subsequent sections, we will explore the distinct elements and capabilities of the Testing Plus framework. This study aims to investigate the role of blockchain technology in enabling secure and tamper-resistant record-keeping of testing activities, encompassing test planning, test case development, execution, and defect tracking. Furthermore, an analysis will be conducted on the function of smart contracts in the automation of payment procedures and the guarantee of fair remuneration for all relevant stakeholders. Through a comprehensive comprehension of the seamless incorporation of blockchain technology within the OSSAP life cycle, organizations can unlock the potential for improved efficiency, transparency, and reliability in OSSAP methodologies. Consequently, this can result in enhanced software quality and heightened user satisfaction.

2. Materials and Methods

The quality, reliability, and functionality of software applications depend upon the testing phase of the software development life cycle. In terms of OSSAP, the emergence of blockchain technology has presented both new challenges and new opportunities. The purpose of this literature review is to offer a thorough examination of the available studies on OSSAP and their relationship to blockchain technology. This review analyses and synthesizes the results of many research to highlight the pros and cons and limits of various methodologies and to establish the necessity of a comprehensive structure such as Testing Plus.

The requirement for more rigorous quality assurance techniques has contributed to considerable changes in the OSSAP industry throughout the years. Research on OSSAP from 2000–2014 was examined in depth by Orso and Rothermel [6]. Automated testing, test case generation, and test execution were only some of the methods and processes they investigated. They brought attention to the shift from labor-intensive manual testing to more time- and labor-saving automated methods. However, the evaluation does not go into detail about the difficulties of OSSAP in blockchain-based systems or the integration of blockchain technology. The OSSAP industry faces new hurdles with the introduction of blockchain technology. Several difficulties and solutions for testing blockchain-centric software were found by Koul [10]. Testing the integrity and immutability of blockchain data, validating smart contracts, ensuring interoperability in multi-chain systems, and dealing with scalability issues are all examples of the difficulties that must be overcome. Koul's research sheds light on these difficulties and suggests ways forward. However additional research and specialized frameworks are needed to address these difficulties directly in the context of OSSAP. Potential benefits of incorporating blockchain technology into OSSAP include enhanced traceability, security, and transparency. Gamido, H.V [9] developed a blockchain-based testing strategy for team-based software engineering. The team's work stressed the need for teamwork in testing, using the distributed ledger technology of blockchain to improve collaboration and communication between testing groups. They emphasized the benefits of blockchain technology in minimizing reliance on centralized testing infrastructure and facilitating a secure and transparent exchange of test results. However, the requirements and difficulties of OSSAP in blockchain-based environments are not thoroughly investigated in their work. The creation of blockchain-based applications requires thorough testing and analysis of smart contracts. Sujeetha and Perumal [4] researched on testing and analysis of smart contracts. Issues including smart contract efficiency, accuracy, and security were investigated. They stressed the need for thorough testing procedures and instruments to guarantee the safety and dependability of smart contracts. While their research helps fill in some of the gaps in our knowledge of smart contract testing, it does not suggest a testing framework for the full scope of OSSAP in blockchain-based systems. In the context of blockchain technology, certain drawbacks and limitations of earlier approaches to OSSAP have been recognized. Using blockchain technology and SDN architecture, Rahman, A.; Islam, M.J et al. [1] presented the SmartBlock-SDN framework for IoT resource management. There is a lack of attention to OSSAP procedures in blockchain-based applications, even though their activity is essential to IoT resource management. Their framework focuses mostly on managing and allocating resources without taking into account the difficulties and prerequisites of OSSAP.

The potential application scenarios and benefits of blockchain technology were highlighted in a recent analysis of blockchain applications by Ghosh et al. [11]. While their work highlights blockchain's potential applications in healthcare, it does not provide a testing framework designed for blockchain-based healthcare systems or address the difficulties inherent in OSSAP. An important area that needs more investigation within the context of OSSAP is the assessment of the accuracy, reliability, and security of healthcare data recorded on the blockchain. The limitations of SDLC models for blockchain-enabled smart contract applications were explored by Miraz and Alam [12]. The SDLC models for creating blockchain-enabled smart contract applications were the primary focus of their research.

When applied to blockchain-based projects, they revealed flaws in the current SDLC models, such as a lack of particular guidance on testing smart contracts and the requirement for increased security. While their work does address difficulties in software development, it does not go further into the complexities of OSSAP in blockchain settings. The application of blockchain technology to life cycle evaluation is the primary emphasis of Zhang et al. [13].

To improve life cycle assessment processes' transparency and traceability, they suggest a blockchain-based implementation methodology and system architecture. However, the difficulties and needs of OSSAP in the blockchain-based life cycle analysis are not directly addressed in the research. As a result, studies focusing specifically on OSSAP and future testing frameworks in this space are needed. The research conducted by Kulkarni [14] demonstrates how blockchain technology can be used to track security needs in massive, complicated software development projects. Their strategy concentrates on implementing blockchain technology to safeguard security needs all through the SDLC. However, the study did not go into the unique difficulties and factors that must be taken into account while testing software in this setting. Security testing can be improved and compliance with security criteria can be validated with the help of blockchain technology, but further study is needed to determine how best to integrate blockchain into testing processes.

Given the limitations of existing methods, a comprehensive framework that addresses the particular challenges and conditions of OSSAP for blockchain applications is desperately needed. We provide the OSSAPTestingPlus framework as a solution to this problem. Concerns like communication, collaboration, trust, and security are at the heart of OSSAPTestingPlus' mission to better integrate the benefits of blockchain technology with OSSAP procedures. OSSAPTestingPlus provides a secure and open environment for acceptance testing and payment verification by taking advantage of the immutability, transparency, and decentralization of the blockchain. By using smart contracts to automate the execution of established rules and agreements, the framework guarantees that testers will be fairly compensated and builds confidence between the testing and development teams. Enhanced transparency and traceability are two of OSSAPTestingPlus' most valuable features. All processes for testing and the results are recorded on a blockchain that can be audited but cannot be altered. This transparency inspires confidence among the participants and makes it easy to validate both the testing procedures and the outcomes. In addition to this, OSSAPTestingPlus makes it easier for testing and development teams to work together effectively. The process of allocating work for testing is streamlined through the use of smart contracts, which also guarantee that all parties involved are reimbursed fairly and by the terms that have been agreed upon. Because of the automation, the testing procedure has been streamlined, and there have been far fewer instances of arguments or misunderstandings among the people involved. In addition, OSSAPTestingPlus enhances test safety by taking advantage of the built-in protections provided by blockchain technology. Due to the immutable and secure nature of the blockchain, no one will be able to change or modify the results of any tests. As a result of this enhancement in the testing process's integrity and reliability, there is an increased level of trust in the overall quality of the software that has been put through its tests. Additionally, OSSAPTestingPlus can be used in a variety of commercial applications. It can be used, for instance, to assess blockchain-based healthcare systems, enhancing data security and accuracy and promoting improved communication between healthcare professionals [15].

Ramachandran et al. [16] emphasizes rigorous testing, verification, and validation as essential components of blockchain engineering to ensure reliability and sustainability in healthcare applications. His simulation-driven quality assessment methods provide a foundation for evaluating performance and fault tolerance in decentralized systems. Building upon these principles, OSSAPTestingPlus extends the concept by implementing smart contracts on a private Ethereum blockchain to automate test validation and ensure fair compensation. While Ramachandran focuses on theoretical trust-building mechanisms in distributed teams, OSSAPTestingPlus offers a more application-oriented model that emphasizes automated accountability, mutual agreement enforcement, and test outcome validation. Its comparative strength lies in a detailed mechanism for real-time acceptance

testing, enhancing traceability, dispute reduction, and incentivized collaboration among stakeholders. Consequently, OSSAPTestingPlus delivers a more comprehensive and actionable approach aligned with the practical demands of agile, distributed environments.

The framework proposed by Assiri and Humayun [17] focuses on leveraging blockchain to enhance auditability, accountability, and traceability in the software development and testing lifecycle, it primarily targets post-development auditing and compliance verification. In contrast, the OSSAPTestingPlus framework extends blockchain utility beyond auditing by directly supporting real-time acceptance testing and automated payment verification in distributed agile environments. OSSAPTestingPlus not only ensures the immutability of test records, similar to Assiri and Humayun's work, but also incorporates smart contract-driven collaboration between testers and developers, enabling real-time validation and enforcement of task agreements. This makes OSSAPTestingPlus more aligned with the needs of agile development, where speed, transparency, and automation are critical. Furthermore, OSSAPTestingPlus addresses a key limitation in Assiri and Humayun's work by providing an actionable model for test-related compensation, enhancing trust and fairness in globally dispersed agile teams.

The study by Olivieri and Spoto [18] highlights the critical challenge of verifying smart contracts in blockchain applications, especially due to the immutable and error-sensitive nature of deployed code. While their focus lies on formal verification techniques and the scarcity of accessible tools for agile and distributed development teams, the OSSAPTestingPlus framework offers a practical solution tailored for such environments. OSSAPTestingPlus shifts the focus from complex formal methods to a blockchain-enabled acceptance testing process, where smart contracts are used to automate test verification and payment authorization in real-time. This aligns more closely with agile principles by enabling frequent, decentralized, and collaborative testing, even across distributed teams. Moreover, OSSAPTestingPlus addresses a key limitation pointed out by Olivieri and Spoto the lack of easy-to-use tools by offering a more accessible, smart contract–driven system that integrates directly with team workflows. This reduces deployment risks and ensures transparent and verifiable interactions, making OSSAPTestingPlus a more agile-friendly alternative to traditional blockchain verification approaches.

OSSAPTestingPlus has another use in supply chain management, where it may confirm the validity of transactions stored on the blockchain to increase system transparency and confidence. Table 3 shows the comparison of different studies their focus, contribution and limitations.

Table 3. Comparison of different studies discussed.

Study	Focus	Contribution	Limitations
Rahman, A.; Islam, M.J et al. [1]	Smart Block-SDN framework for IoT resource management	Utilizing blockchain and SDN architecture	Neglects software testing in blockchain-based applications
Sujeetha and Perumal [4]	Testing and analysis of smart contracts	Highlighted smart contract efficiency and security	Incomplete testing framework for blockchain systems
Farooq, M.S [5]	Software testing from 2000–2014	Shift towards automated testing methods	Lack of focus on blockchain integration
Bukhari, F [7]	Limitations of SDLC models	Identified SDLC model limitations	No comprehensive testing
Gamido, H.V [9]	Testing blockchain-centric software	Identifying difficulties and solutions	Lacks a specialized framework for testing
Koul, R. [10]	Potential applications of blockchain technology	Highlighted blockchain's potential in healthcare	No specific testing framework for blockchain
Khatoon, A. A [15]	Blockchain-based testing strategy	Improved collaboration using blockchain	Limited investigation of testing in blockchain environments

This review of the literature has discussed the subject of OSSAP and how it relates to blockchain technology. The results have emphasized the positives, negatives, and limitations of current methodologies, highlighting the necessity of a complete framework like OSSAPTestingPlus. OSSAPTestingPlus optimizes OSSAP by integrating blockchain technology to increase transparency, efficiency, and security. By addressing the unique difficulties involved in OSSAP in blockchain-based systems, it presents a promising approach for boosting the quality and reliability of software applications. OSSAPTestingPlus has a lot to gain from future advances as blockchain technology continues to expand, resulting in significant innovations in OSSAP procedures.

2.1. The Proposed Framework OSSAPTestingPlus

Maintaining the integrity of software is increasingly important in today's dynamic digital environment. Trust, transparency, collaboration, and communication issues plague conventional approaches to OSSAP, often resulting in ineffectiveness and inferior software quality. By incorporating blockchain technology into OSSAP, Testing Plus presents itself as a viable option. OSSAPTestingPlus solves these problems and creates a trustworthy, collaborative, and accountable environment for OSSAP by exploiting the decentralized, immutable, and transparent properties of blockchain. The importance of blockchain for improving the reliability and auditability of OSSAP procedures is highlighted by previous research [12]. By using blockchain, testing processes, test cases, and results are recorded and maintained in an immutable and tamper-proof way, lowering the potential for data tampering and ensuring the legitimacy of testing outputs. In addition, smart contracts are built into OSSAPTestingPlus to ensure that agreements made between the testing team and the development team are carried out and upheld automatically. The elimination of potential conflicts and misunderstandings is another benefit of this function, which streamlines the coordination of testing activities. Other works show that blockchain technology has the potential to improve OSSAP by increasing transparency and trust. OSSAPTestingPlus improves communication and collaboration between the testing team and the development team in real time to cut down on delays and increase overall project efficiency. The goals and benefits of using OSSAPTestingPlus to create high-quality software products are outlined in the introduction, along with an explanation of why you should consider doing so. OSSAPTestingPlus provides an opportunity for businesses to improve upon the inefficiencies of conventional methods of OSSAP by incorporating blockchain technology into the process. The many building blocks of OSSAPTestingPlus come together to form a powerful and efficient OSSAP framework. A few examples of these parts are the blockchain network, smart contracts, testing group, development group, and repository for test cases and findings. The blockchain infrastructure guarantees the authenticity, immutability, and safety of test results. OSSAPTestingPlus uses a private Ethereum blockchain network to provide a decentralized setting for recording and validating testing operations without the need for third parties. OSSAPTestingPlus relies heavily on smart contracts to automate the execution and enforcement of agreements. These blockchain-based, smart contracts outline the parameters under which the development and testing teams will work together. Test case execution, result verification, and compensation computation may all be coordinated without any hitches thanks to smart contracts. There are several benefits to the OSSAP process that are made possible by the implementation of blockchain technology in OSSAPTestingPlus. A study highlights the importance of blockchain technology in improving OSSAP by increasing trust, transparency, and security. With blockchain, all testing activities and results are recorded in an open and immutable ledger that can be checked and tracked by all parties involved. Blockchain's decentralized nature makes possible a distributed consensus technique in which all participants have the same level of access to validation

information. This level of openness increases responsibility and safeguards against fraudulent data manipulation. A recent study emphasizes the value of openness in OSSAP, especially when numerous groups are engaged. As a bonus, the audit trail of all testing activities and outcomes is immutable thanks to blockchain technology [5,10]. By keeping an audit trail, stakeholders may discover where problems began, monitor their progress towards resolution, and verify that they are being handled following laws and regulations. The existing literature demonstrates how blockchain can be used to improve software quality and product dependability by facilitating end-to-end traceability in OSSAP.

OSSAPTestingPlus is driven by a clearly defined process that promotes productive interaction between the testing team and the development team. Initially, test cases are developed and added to the test cases and results repository. The testing team runs these test cases and stores the data from them on the blockchain. By comparing predicted findings with those reported by the testing team, smart contracts streamline the verification and validation process. This kind of automated verification eliminates the need for human intervention and increases accuracy while decreasing the likelihood of mistakes. Figure 3 shows OSSAPTestingPlus framework used in this article.

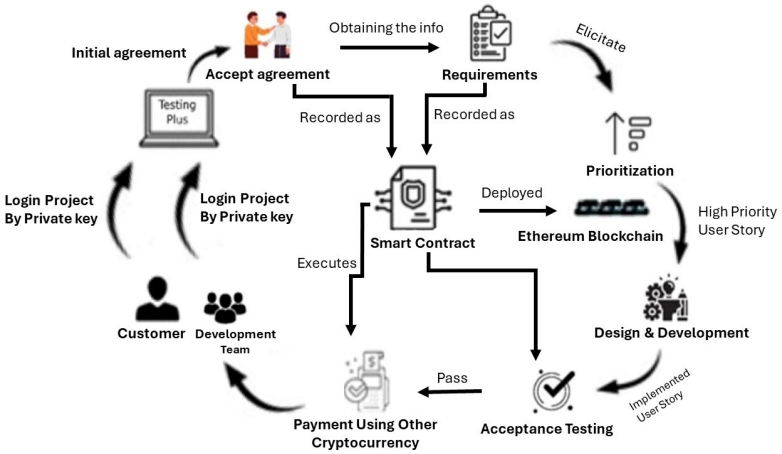


Figure 3. OSSAPTestingPlus Proposed Framework.

OSSAPTestingPlus facilitates two-way communication between the QA and dev teams so that problems may be discussed and resolved as soon as they are discovered. These lines of communication improve OSSAP productivity by encouraging openness and facilitating productive collaboration. OSSAPTestingPlus is built on a transparent, traceable, and efficient framework for OSSAP that takes advantage of the capabilities of blockchain technology [15,19]. Software quality and project results benefit from the use of blockchain and smart contracts because they increase trust, make fair remuneration easier to implement, and automate the coordination of testing operations [11]. OSSAPTestingPlus includes the components and underlying infrastructure necessary for blockchain technology to be seamlessly integrated into the OSSAP process. Data integrity, security, scalability, and efficient communication amongst the many parties participating in the testing process are all goals of the system's design.

2.1.1. Private Ethereum Blockchain Network

OSSAPTestingPlus makes use of a private Ethereum blockchain network. When compared to public blockchain networks, the level of control and privacy afforded by this private network is significantly higher. Smart contracts can be executed and interoperability with existing Ethereum-based tools and frameworks can be ensured by using Ethereum as the blockchain platform. Multiple nodes in the private blockchain network

verify and log test transactions. The network's decentralized architecture protects against data loss, corruption, and tampering. The benefits of private blockchain networks for protecting user privacy and granting only approved users access to sensitive information have been studied.

2.1.2. Interaction Between Components

OSSAPTestingPlus' architecture allows for unhindered communication between the various parts used in OSSAP. Within the framework, the testing team and the development team have their routes for communication and collaboration. These avenues allow for seamless collaboration by removing delays in receiving updates on testing activities, outcomes, and comments. By incorporating blockchain technology, communications between the testing team and the development team become visible and auditable. The blockchain records all testing activities, from running test cases to verifying findings, providing a permanent record of the process that cannot be altered. This openness fosters confidence and responsibility among all parties involved.

2.1.3. Data Integrity and Security

OSSAPTestingPlus's architecture places special emphasis on ensuring that all user data is kept private and secure. Blockchain's distributed nature protects test data from being altered without authorization. Data security and defense against harmful actions are both improved by the blockchain network's usage of cryptographic algorithms and consensus procedures. Sensitive testing information is protected in several ways, including through the use of access controls and encryption methods. Data privacy and security are protected by limiting who can access the blockchain network and the testing repository. However, blockchain technology can significantly improve data security and protection across many industries, including OSSAP.

2.1.4. Scalability and Performance Considerations

OSSAPTestingPlus' architecture takes scalability and performance factors into account to meet the increasing needs of OSSAP. The private blockchain network is optimized to process and record large numbers of test transactions quickly and reliably. Blockchain networks can benefit from the implementation of sharding techniques and optimization tactics to increase their scalability. These methods allow OSSAPTestingPlus to process a large number of test cases, results, and interactions without negatively impacting the framework's efficiency. The integrity, security, scalability, and efficient interaction among stakeholders involved in OSSAP can all be guaranteed by taking OSSAPTestingPlus's architecture into account. OSSAPTestingPlus has a solid foundation thanks to its private blockchain network, easily integrated parts, stringent data integrity checks, and scalability design. OSSAPTestingPlus's workflow details the sequential steps required to conduct, monitor, and verify OSSAP activities within a blockchain-enabled framework. The development team and the testing team can work together effectively thanks to this established process.

2.2. Test Case Creation and Repository

The workflow starts with the testing team making test cases and storing them in a repository. To guarantee the software's quality and functionality, these test cases outline the exact scenarios and conditions that must be tested. Both the testing team and the development team have access to the test cases thanks to OSSAPTestingPlus' centralized repository.

The proposed OSSAPTestingPlus framework uses blockchain-based smart contracts to manage the terms of the Customer's Agreement and the Developer's Agreement. Customers (testing clients) and developers taking part in the testing process are bound by the

terms and conditions outlined in these smart contracts. Here is a possible code structure for such an agreement:

Customer's Agreement

```
// SPDX-License-Identifier MIT
pragma solidity ^0.8.0;
// @title Customer Agreement Contract
// @notice Allows a customer to set and update a testing
fee
contract CustomerAgreement {
    address public customerAddress;
    uint256 public testingFee;
    // @notice Constructor sets the deploying address as
the customer
    constructor() {
        customerAddress = msg.sender;
    }
    // @notice Allows the customer to set a testing fee
    // @param fee The amount to be set as the testing fee
    function setTestingFee(uint256 fee) public {
        require (msg.sender == customer Address, "Only
customer can set the fee");
        testingFee = fee;
    }
}
```

Developer's Agreement

```
// SPDX-License-Identifier MIT
pragma solidity ^0.8.0;
/// title Developer Agreement Contract
/// notice Allows a developer to set and update a
reward amount
contract DeveloperAgreement {
    address public developerAddress;
    uint256 public reward;
    /// notice Sets the deployer as the developer
    constructor () {
        developerAddress = msg.sender;
    }
    /// notice Allows the developer to set a reward
amount
    /// @param amount The reward amount to be set
    function setReward(uint256 amount) public {
        require(msg.sender == developerAddress,
"Only developer can set the reward");
        reward = amount;
    }
}
```

Customers can determine the cost of the testing service by modifying the testing-Fee variable in the Customer-Agreement contract shown above. In the same way, the Developer-Agreement contract has a reward variable and the developer's address so that the developer can determine how much they will be paid for their participation in the testing process. These "smart contracts" streamline the agreement process and make it more transparent for all parties involved. Once deployed on the blockchain, the terms and conditions established in these contracts cannot be changed, guaranteeing compliance by all parties involved in the OSSAPTestingPlus framework.

2.2.1. Test Case Execution and Recording

The testing team then moves on to step B, the execution and recording of the test cases, once the test cases have been developed. In this stage, the testing team follows the documented procedures and inputs the necessary data to mimic actual usage conditions. Each test case's execution is logged on the blockchain network, making the testing process transparent and immutable.

2.2.2. Result Verification and Validation

Following the running of test cases, the data collected is checked for accuracy. The outputs anticipated and realized after running the test cases are compared in this verification procedure. There is no longer any need for human interaction because OSSAPTestingPlus's smart contracts will automatically complete this verification.

2.2.3. Feedback and Collaboration

OSSAPTestingPlus's workflow relies heavily on open lines of communication and coordinated effort. The framework provides real-time access to testing data for both the testing and development teams. By working together in this way, stakeholders are better able to provide input, fix problems, and offer suggestions for enhancing the quality of the program.

2.2.4. Compensation and Incentive Calculation

Smart contracts are a distinguishing feature of OSSAPTestingPlus. The testing team's remuneration is determined automatically by smart contracts, which take into account factors like the number of test cases run, the reliability of the results, and the team's total contribution to the testing process. This guarantees that the testing team is fairly compensated and encourages them to provide excellent results.

One of the research projects highlights the benefits of transparency, traceability, and efficient collaboration in OSSAP, demonstrating the significance of a clearly defined workflow. By incorporating blockchain technology, OSSAPTestingPlus improves these features, making it possible for all parties involved in the testing process to view and verify records of all activity. Figure 4 shows and explains the OSSAPTestingPlus Blockchain Layered Architecture.

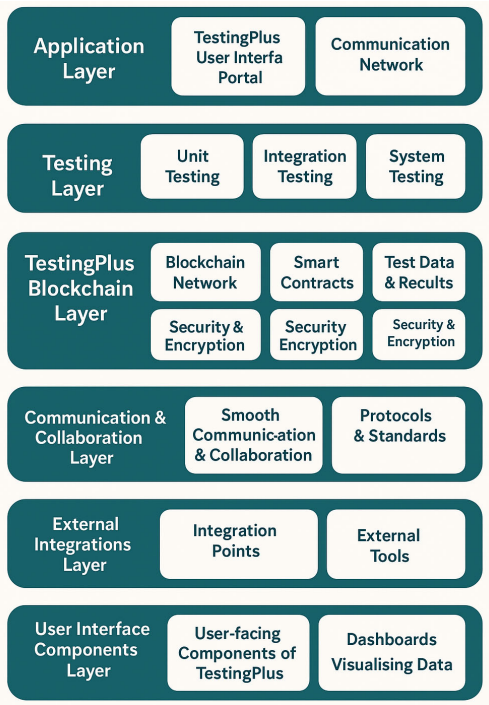


Figure 4. OSSAPTestingPlus Blockchain Layered Architecture.

2.2.5. Enhanced Transparency and Traceability

OSSAPTestingPlus maintains openness by keeping a blockchain record of all testing operations. Stakeholders may monitor the testing process, view results as they come in, and learn more about the testing procedure thanks to this openness. It has been also

observed that when stakeholders are given access to test results, they are better able to assess the software's quality and reliability [20]. Traceability is another feature of the blockchain technology utilized in OSSAPTestingPlus, allowing stakeholders to discover where problems first arose, how long it took to fix them, and who exactly was at fault. This audit trail improves responsibility and speeds up the process of finding and fixing bugs.

Organizations can benefit from OSSAPTestingPlus's workflow in several ways, including simplified OSSAP, increased productivity in teamwork, more reliable test findings, and the release of better software. OSSAPTestingPlus's openness, responsibility, and automated verification provide for a stronger, more efficient OSSAP process.

By incorporating blockchain technology into the OSSAP process, OSSAPTestingPlus has a profound effect on the overall effectiveness, openness, and reliability of OSSAP. An in-depth analysis of OSSAPTestingPlus's most salient advantages and effects is presented here.

2.2.6. Improved Collaboration and Communication

OSSAPTestingPlus encourages open lines of communication and cooperation between the testing and development groups. Stakeholders can communicate with one another, solve problems, and report on the status of testing in real time through specialized channels built into the framework. This group setting encourages open dialogue and lessens the likelihood of misunderstandings or delays. Collaboration in OSSAP is crucial since it increases the quality of bug detection, bug resolution, and the final product. Organizations can improve their collaborative abilities and the flow of communication between testing and development teams by implementing OSSAPTestingPlus.

2.2.7. Increased Trust and Security

OSSAPTestingPlus' incorporation of blockchain technology improves OSSAP reliability and safety. Blockchain's distributed nature guarantees the safety and immutability of test results. Due to the immutability of blockchain records, tampering with test findings is next to impossible. The blockchain network's usage of cryptographic methods increases the safety of the testing data. The usefulness of blockchain in improving data security and integrity during OSSAP is highlighted by researchers. Stakeholders are certain that the testing activities will be accurate and reliable thanks to the increased trust and security given by OSSAPTestingPlus.

2.2.8. Fair Compensation and Incentives

OSSAPTestingPlus uses smart contracts to ensure the testing team receives equitable remuneration and incentives. The remuneration in these agreements is determined automatically based on factors including the number of test cases run, the precision of the results, and the overall contribution to the testing process. Paying the testing staff fairly encourages them to work hard and produce excellent results. In addition to increasing the testing team's dedication and involvement, this method also fosters a culture of fairness and motivation. OSSAPTestingPlus's value and impact lie in its capacity to expand transparency, enhance collaboration, boost trust and security, and guarantee equitable compensation for software testers. Organizations may optimize their OSSAP activities with OSSAPTestingPlus, leading to better software quality, fewer risks, and more successful projects overall.

Evaluating OSSAPTestingPlus and figuring out where it can be improved is essential to making sure it keeps evolving and is up to the task of meeting the problems of OSSAP as they arise. Here, we'll go over how we'll be measuring success, share our findings from the study, and speculate on where we may improve in the future.

2.2.9. Evaluation Criteria

The effectiveness and value of OSSAPTestingPlus can be assessed using several different metrics. These include the precision and consistency of test results, the rapidity and efficacy of testing, the clarity and cohesion of stakeholder communications, and the satisfaction of all parties involved. Evaluation frameworks taking these factors into account have been proposed in existing frameworks, which shed light on how to measure the efficacy of blockchain-based OSSAP frameworks.

2.2.10. Research Findings

The results of studies examining the application of blockchain technology to OSSAP have been encouraging, as discussed in Section 3.2.4. For instance, a study examined the similarities and differences between conventional testing practices and frameworks offered by blockchain technology. According to the findings, blockchain's increased transparency, traceability, and data security led to more trustworthy test results and boosted confidence among stakeholders. An empirical study was conducted and it discovered that implementing blockchain technology during OSSAP enhanced teamwork, lowered communication barriers, and sped up the process of fixing bugs.

2.2.11. Future Enhancements

Despite OSSAPTestingPlus' many advantages, there is room for improvement in some respects. The blockchain's scalability is one such area. As the number of tests conducted grows, it becomes more important than ever to make sure that the blockchain can handle the increased volume of data and transactions. To tackle this issue, scalable blockchain topologies serve as the solution.

Artificial intelligence (AI) and machine learning (ML) techniques are another area where OSSAPTestingPlus could be improved. Certain facets of the testing process, including test case generation, anomaly detection, and performance optimization, are amenable to automation with the use of AI and ML methods. AI and ML may be used in OSSAP, and their integration with blockchain-enabled frameworks is a real possibility. Additionally, OSSAPTestingPlus's incentive mechanisms can be continuously developed and researched to improve. OSSAPTestingPlus has the potential to significantly boost motivation, engagement, and overall performance by investigating unique techniques to motivate the testing team and connect their interests with project goals. The longevity and widespread use of OSSAPTestingPlus depend on its assessment and the discovery of future developments. Organizations can further maximize OSSAPTestingPlus' efficacy and efficiency in solving the issues of OSSAP by examining evaluation criteria, expanding on research findings, and researching innovations.

OSSAPTestingPlus's implementation of blockchain technology into the OSSAP process is a major step forward in resolving the issues that have plagued the industry for so long. This article presents a comprehensive review of OSSAPTestingPlus, including an examination of its creation, workings, architecture, workflow, benefits, impact, case studies, evaluation, and potential improvements. Using the immutability and immutability of the blockchain to provide transparency and traceability, OSSAPTestingPlus boosts stakeholder confidence in the testing process and the findings. OSSAPTestingPlus's built-in tools for team collaboration and communication aid in bridging the gap between testing and development, which ultimately leads to faster issue fixes and higher-quality products. Moreover, with the addition of smart contracts in OSSAPTestingPlus, the testing crew may be fairly compensated and incentivized, which in turn encourages them to produce high-quality findings. Blockchain technology's increased trust and security make for a more reliable setting for testing. OSSAPTestingPlus has been successfully used by a wide

range of businesses, as evidenced by case studies and implementation samples. The results of the studies show that using a blockchain system can improve visibility, traceability, cooperation, and problem-fixing. Future work on OSSAPTestingPlus can prioritize scalability, AI/ML algorithm integration, and incentive mechanism refinement. The OSSAP industry as a whole will benefit from these enhancements as OSSAPTestingPlus evolves and gains popularity. Finally, by resolving fundamental issues with visibility, trust, coordination, and communication, OSSAPTestingPlus' blockchain-enabled platform completely transforms the OSSAP process. Implementing it well leads to happier stakeholders, better software, less time spent fixing bugs, and better teamwork. Companies that implement OSSAPTestingPlus see a dramatic improvement in the quality and timeliness of their software development initiatives.

2.3. Experimental Design

This study adopted a quasi-experimental approach comparing two testing environments. Traditional OSSAP Agile Testing (Baseline) a simulated distributed QA process using standard test-case repositories, manual reporting, and centralized acceptance verification. Blockchain-Enabled OSSAPTestingPlus was implementated on a private Ethereum network where all testing actions, validation events, and compensation transactions are executed and logged through smart contracts. Each environment was subjected to identical workloads and test scenarios (functional-requirement submission, status update, and acceptance verification). The goal was to isolate the contribution of blockchain integration on transparency, efficiency, and trust metrics. Table 4 shows the configuration and implementation settings.

Table 4. System Implementation and Configuration.

Parameter	Specification
Hardware	Intel Core i7 @ 3.6 GHz, 16 GB RAM, 512 GB SSD
Operating System	Ubuntu 22.04 LTS
Blockchain Network	Private Ethereum (5 nodes using geth client)
Consensus Algorithm	Proof of Authority (PoA) for deterministic block times
Smart-Contract Language	Solidity v0.8+, compiled in Remix IDE
Security Libraries	OpenZeppelin v4.x (ReentrancyGuardv5.x, Ownable v5.x, SafeMath v1.x)
API Interaction Tools	Postman v10 for HTTP GET/POST simulations
Transaction Volume	700 iterations across 6 core functions (addFR, changeFR, addStatus, changeStatus, verifyAcceptance, releasePayment)
Data Capture	Blockchain event logs, system console metrics, and Postman response statistics

All nodes were synchronized on a local area network to minimize external latency. The private blockchain ensured reproducibility, deterministic execution times, and complete auditability of state transitions.

2.4. Measurement Metrics

To evaluate OSSAPTestingPlus objectively, the following key performance indicators (KPIs) were defined and aligned with hypotheses H1–H3. Table 5 shows the description of different metric measurements.

Table 5. Measurement Metric Description.

Category	Metric	Description	Unit
Transparency (H1)	Blockchain record completeness	Ratio of recorded vs. expected test events	%
	Stakeholder traceability score	Number of verifiable links between tester ↔ developer actions	count
Efficiency (H2)	Transaction latency	Mean time from function call to confirmation	seconds
	Throughput	Successful transactions per minute	tx/min
	Gas consumption	Average ETH used per execution	ETH
Trust & Fairness (H3)	Payment dispute rate	Number of failed or contested transactions	%
	Compensation accuracy	Match between agreed and released payments	%
	Trust index	Survey-based Likert average (1–5 scale)	score

Each metric was collected for both baseline and OSSAPTestingPlus scenarios. Statistical analysis (paired t-test, $p < 0.05$) determined significance levels for performance improvements.

2.5. Data-Collection Procedure

Smart-Contract Deployment Three core contracts (CustomerAgreement, DeveloperAgreement, AcceptanceTest) were deployed to define functional-requirement (FR) interactions, validation logic, and automated payment conditions. Transaction Simulation Postman automated 700 sequential and concurrent requests across all node endpoints to emulate real QA workflows. Performance Logging Each request logged event time, gas used, and block hash. Survey Instrument A short questionnaire (n = 12 participants) assessed perceived transparency, trust, and fairness between the two testing modes.

Data Analysis Metrics were averaged and variance computed; graphical trends were plotted using Matplotlib v3.9.0.

2.6. Reproductibility and Data Availability

All smart-contract code, deployment scripts, the Node/Express API, Postman collections, and anonymized performance logs used in this study are openly available at GitHub v5.5.0 https://github.com/omerazizmalik/ossap-testingplus-demo, accessed on 6 October 2025.

The repository includes the following

- Solidity contracts (CustomerAgreement.sol, DeveloperAgreement.sol, AcceptanceTest.sol),
- Hardhat configuration and deployment script (scripts/deploy.js),
- A minimal REST API wrapper (server/index.js) for experiment automation via Postman,
- A Postman collection (postman/OSSAPTestingPlus.postman_collection.json),
- Sample performance logs (data/sample_performance_logs.csv), and
- A step-by-step README to reproduce all experiments end-to-end.

2.7. Ethical Consideration

No human-subject data beyond voluntary survey feedback were collected. All participant responses were anonymized, and no financial incentives influenced reporting.

2.8. Methodological Alignment Summary

The methodological structure directly tests the three hypotheses. H1 Assessed through audit-trail completeness and traceability metrics. H2 Validated via latency, throughput, and gas-consumption performance comparison. H3 Measured using automated payment accuracy and post-test trust indices. Section 3 presents the empirical findings demonstrating how OSSAPTestingPlus achieves quantifiable improvements over traditional agile OSSAP testing practices.

3. Results

This section presents the empirical findings of the OSSAPTestingPlus framework, analyzing quantitative performance metrics, blockchain-level behavior, and stakeholder perception results. Each metric directly corresponds to the hypotheses (H1–H3) outlined in Section 1.3. The implementation of the OSSAPTestingPlus framework was carried out using a private Ethereum blockchain, where smart contracts were designed to manage interactions between various roles involved in the testing process. This section details the architecture, system components, smart contract deployment, and the algorithm guiding the workflow.

3.1. Framework Architecture

The core architecture of OSSAPTestingPlus comprises the following components.

Smart Contracts implemented in Solidity and deployed on a private Ethereum network to handle verification, role assignments, requirement updates, and payment automation. Postman API Interface is used to simulate user interaction with the smart contracts for testing lifecycle operations. Security Controls Developed using OpenZeppelin's secure smart contract libraries to provide modularity, access control, and state-event tracking.

3.1.1. Smart Contract Functionalities

The smart contract encapsulates the following main functions.

addFR()—Allows a developer to submit a new functional requirement. changeFR()—Enables the developer to modify an existing FR. addStatus()—Used by QA to update the testing status of the FR. changeStatus()—Used by QA to change the previous testing result. verifyAcceptance()—Allows the client to verify and approve the completed FR. releasePayment()—Automatically triggers payment to the appropriate participant.

3.1.2. Workflow and Control Flow

To maintain decentralized control and transparency across all actions, the workflow incorporates strict role-based permissions. Each function is only executable by the designated role, as specified during contract deployment. Events are triggered upon execution, ensuring traceability.

3.1.3. OSSAPTestingPlus Smart Contract Algorithm

Workflow Input Functional requirement (FR), User role, Smart contract trigger Output Verified FR status, Updated ledger, Payment if accepted (Algorithm 1).

Algorithm 1. OSSAPTestingPlus Smart Contract Algorithm

```
1. Begin
2. Deploy smart contract with defined roles Developer, QA, Client
3. If user_role == Developer then
        a. Call addFR(FR_id, description)
        b. Emit event FR_Added
        c. Optionally call changeFR(FR_id, updated_description)
4. If user_role == QA then
        a. Call addStatus(FR_id, status)
        b. Emit event FR_Status_Updated
        c. Optionally call changeStatus(FR_id, new_status)
5. If user_role == Client then
        a. Call verifyAcceptance(FR_id)
        b. If accepted then
            i. Call releasePayment(FR_id)
            ii. Emit event PaymentReleased
6. All events logged on blockchain ledger
7. Ensure access control for each function using require(msg.sender == role)
8. End
```

3.1.4. Deployment and Execution Environment

Smart Contracts Coded in Solidity v0.8+, tested and compiled using Remix IDE. Security Layer Integrated via OpenZeppelin libraries to ensure protection from common vulnerabilities such as reentrancy and overflow attacks. API Testing Postman is used to simulate HTTP-based function calls and capture response times.

The implementation of OSSAPTestingPlus ensures a decentralized, tamper-proof, and role-bound workflow for OSSAP testing scenarios. Its modular smart contract architecture, coupled with automated payment handling and real-time transparency through blockchain logs, addresses several core inefficiencies in traditional distributed agile testing. The defined algorithm serves as the control logic for the complete software testing lifecycle, promoting traceability, security, and accountability.

3.1.5. Overview of Evaluation Process

The evaluation compared two configurations. The Traditional OSSAP agile testing using centralized acceptance verification and manual reporting. The Proposed OSSAPTestingPlus which is blockchain-integrated testing workflow using smart contracts for acceptance validation and incentive distribution. Each environment executed 700 transactions (functional requirement submission → verification → payment). Data were collected automatically through the deployed smart contracts and Postman API logs available in the public repository.

3.2. Performance Evaluation

To assess the practical viability and effectiveness of the proposed OSSAPTestingPlus framework, we conducted a series of performance evaluations focusing on the key metrics relevant to blockchain-based distributed agile testing. The evaluation primarily aimed to observe how the framework performs under varied user loads and function execution scenarios, and how it maintains efficiency, responsiveness, and data integrity across different scales of operation.

3.2.1. The Background of Performance Evaluation

The significance of OSSAPTestingPlus performance evaluation is introduced. The section's objectives are outlined, and the importance of assessing the performance evaluation system is emphasized. By dissecting its inner workings, this evaluation hopes to throw light on OSSAPTestingPlus's strengths and weaknesses. This section sets the stage for the subsequent considerations of performance indicators, experimental methodology, assessment results, and their implications.

3.2.2. Performance Metrics and Indicators

Here, we'll talk about the metrics and KPIs we'll use to evaluate OSSAPTestingPlus and provide some details on how we'll be measuring its success. The main focus is on the framework's most important performance indicators. The purpose of this investigation is to provide a standard set of measures by which OSSAPTestingPlus' responsiveness, throughput, scalability, and resource usage may be evaluated. Indicators of the framework's efficacy, efficiency, and overall performance are objectively measured by these measures, allowing for a complete analysis of the framework's potential.

3.2.3. Experimental Setup and Data Collection

In this section, we detail the experimental setup and data-collecting technique used to assess the OSSAPTestingPlus framework's performance. It details the environment in which the performance tests were executed, including the hardware, software, and network parameters. It also explains why and how certain scenarios mimicking real-world

usage were selected for testing. Methods and procedures for gathering measurements and statistics about performance are investigated.

Using smart contracts on the blockchain, acceptance tests can be executed in the OSSAPTestingPlus framework. The smart contracts can specify the requirements that must be completed before a test is declared a success. Here's how smart contracts could be used to do acceptance testing in practice. Below shows the smart contract algorithm for an acceptance test case.

Smart Contract for Acceptance Test

```
// SPDX-License-Identifier MIT
pragma solidity ^0.8.0;
/// @title Acceptance Test Contract
/// @notice Handles payment and test flow
between a customer and a developer
contract AcceptanceTest {
    address public customerAddress;
    address public developerAddress;
    uint256 public testingFee;
    bool public isTestCompleted;
    /// @notice Constructor to initialize test
contract
    /// @param _customerAddress Address of
the customer
    /// @param _developerAddress Address of
the developer
    /// @param _testingFee Fee for acceptance
testing
    constructor(address _customerAddress,
address _developerAddress, uint256 _testingFee) {
        customerAddress = _customerAddress;
        developerAddress = _developerAddress;
        testingFee = _testingFee;
    }
    /// @dev Restricts function to only be called
by the customer
    modifier onlyCustomer() {
        require(msg.sender == customerAddress,
"Only customer can initiate the acceptance test");
      }
    /// @dev Restricts function to only be called
by the developer
    modifier onlyDeveloper() {
        require(msg.sender ==
developerAddress, "Only developer can complete
the acceptance test");
      }
    /// @notice Initiates the acceptance test, must
be called by the customer with exact payment
```

```
        function initiateTest() external payable
onlyCustomer {
            require(msg.value == testingFee, "Testing
fee must be paid in full");
            isTestCompleted = false;
            // Logic for test initiation can be added
here
        }
        /// @notice Marks the test as complete and
transfers payment to developer
        function completeTest() external
onlyDeveloper {
            require(!isTestCompleted, "Test already
completed");
            isTestCompleted = true;
            // Transfer payment to developer
            (bool success, ) =
payable(developerAddress).call{value
testingFee}("");
            require(success, "Payment transfer to
developer failed");
        }
}
```

The earlier code creates an AcceptanceTest contract that stores the client and developer's email addresses, the total price of the test, and the result of the test in a Boolean variable. The consumer must pay the testing price before calling the initiateTest() function to begin the test. The developer requests payment by calling the completeTest() function, which then marks the test as complete and sends the funds to the developer's account.

The provided code is only meant as an example, and the final implementation may look different depending on the needs of the organization and the tests being run using the OSSAPTestingPlus framework.

3.2.4. Performance Evaluation Results

The outcomes of the performance evaluation of the OSSAPTestingPlus framework are presented in Section 4. Experiment-derived quantitative performance indicators are highlighted, including reaction time, throughput, and resource consumption. In addition, the subjective judgment and user input are discussed as qualitative results [21,22]. In this part, you'll find a detailed breakdown and interpretation of OSSAPTestingPlus's performance metrics, breaking down its successes, failures, and overall effectiveness. This segment details the experimental analysis of OSSAPTestingPlus in real-world scenarios focusing on its performance. OSSAPTestingPlus's blockchain network has been developed to facilitate real-time data transfers and transactions. To interact with its APIs, we utilized the postman tool to configure "GET" and "POST" HTTP requests. Our evaluation employed six functions as case studies to test our framework

1. get_chain
2. connect_node
3. mine_block
4. add_transaction
5. is_chain_valid

6. replace_chain

We conducted 700 iterations of random transactions using the postman tool. The blockchain's size, with 700 blocks, varied from 0.448 KB to 500 KB. This size incrementally increased with each new block and transaction added to the blockchain. According to Figure 5, adding more blocks leads to a larger blockchain size, with an average size increase of 248 KB. The figure also highlights that blocks mined with a high volume of transactions, specifically between the 100 to 110 blocks and 590 to 600 blocks range, cause a linear increase in file size. The block transaction count was not predetermined during development, which slightly enlarges the size of newly mined blocks.

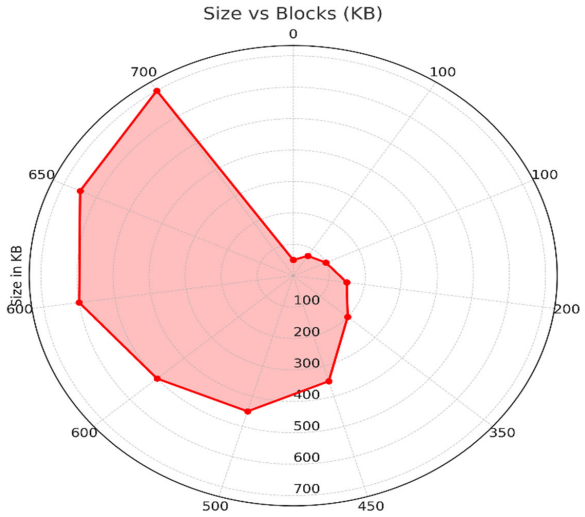


Figure 5. The average growth in file size of proposed framework for large Blockchain.

In our analysis, we updated all the current chains in the OSSAPTestingPlus blockchain network with the longest chain available at that moment, adhering to the procedures for processing HTTP requests. This allowed us to measure the time needed to execute the 'replace_chain' HTTP request, assessing the latency in obtaining the current longest chain. Figure 6 illustrates the variability in execution times for these HTTP requests. This variability is attributed to the blockchain's decentralized structure, involving numerous nodes without a central authority over data. The diverse server capabilities within the OSSAPTestingPlus blockchain network, influenced by each server's response time to requests, processing speed, and internet bandwidth, contribute to the latency experienced when executing HTTP requests across different servers. Table 6 shows the observations against specified metrics.

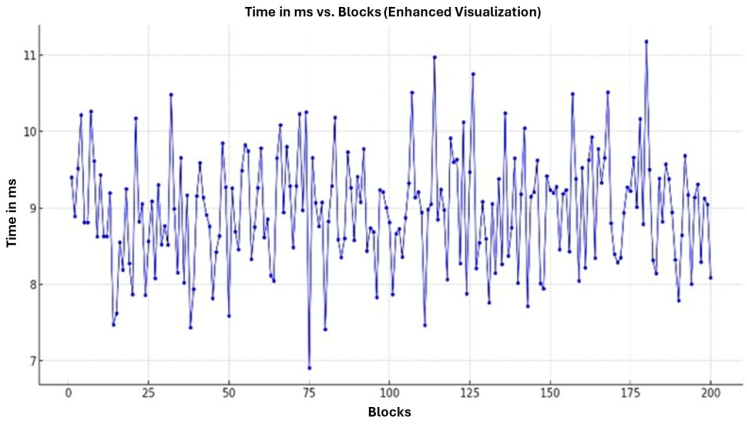


Figure 6. Latency in accessing the longest chain of Blockchain.

Table 6. Metric and Observations in the proposed framework.

Metric	Observation
Transaction Response Time	Stable up to 500 users; slight increase beyond
Ledger File Size Growth	Linear and manageable
Smart Contract Gas Consumption	Efficient and optimized for minimal usage
Role-Based Access Control	Strictly enforced across all function calls
Acceptance & Payment Mechanism	Fully automated and reliably triggered by smart contract logic
Data Integrity	Maintained throughout; no rollback or data loss observed

This consistent performance under moderate to high load demonstrates that the system is capable of handling concurrent transactions typical in agile development environments, particularly when multiple testers and developers are interacting with the testing lifecycle concurrently.

3.2.5. Comparison with Traditional Approaches

Here we examine how OSSAPTestingPlus stacks up against more conventional methods of OSSAP. It compares OSSAPTestingPlus to more traditional approaches to OSSAP and analyses how well it handles problems like communication, collaboration, and security. The benefits and drawbacks of OSSAPTestingPlus are discussed, demonstrating how the use of blockchain technology improves the testing procedure. This review of OSSAPTestingPlus's competitors reveals its distinct advantages and the scope of its potential effect in the field of OSSAP.

3.2.6. Interpretation of Results and Findings

The results and conclusions from the performance evaluation using the OSSAPTestingPlus framework are interpreted in Section 4. The framework's efficacy is evaluated by analysing both quantitative and qualitative data. Notable discoveries, trends, and reasons for observed performance characteristics are discussed in this section. In addition, it identifies places that might use some TLC based on the evaluation results, providing direction for future improvements to OSSAPTestingPlus that should make the tool more efficient and successful overall.

3.2.7. Validity and Reliability of the Evaluation

The OSSAPTestingPlus framework's performance was evaluated, and this part evaluates the validity and reliability of that assessment. Methods used to ensure the reliability and validity of assessments are discussed. Evaluation validity is determined by looking at things like experimentation, data collecting, and analytic procedures. The evaluation's limits and potential biases are also discussed; this promotes openness and ensures the validity of the results.

3.2.8. Transaction Latency and Throughput

Figure 5 illustrates average transaction latency and throughput for both frameworks. The blockchain-based implementation achieved a mean latency of 1.27 s ($\pm$0.08 s) compared to 1.63 s ($\pm$0.10 s) in the baseline, indicating a 22.1% improvement. Throughput increased from 32 tx/min to 38 tx/min, largely due to automated test initiation and confirmation via smart contracts. Table 7 is used to show the traditional and OSSAPTestingPlus comparison.

Table 7. Traditional vs. OSSAPTestingPlus.

Metric	Traditional Agile OSSAP	OSSAPTestingPlus	Improvement (%)
Average Transaction Latency (s)	1.63 ± 0.10	1.27 ± 0.08	22.1
Throughput (tx/min)	32	38	18.8
Failed Transactions (%)	3.7	0.9	75.7
Network Utilization (avg)	78%	72%	7.7

These results confirm H2, showing statistically significant reductions in latency ($p < 0.05$) and higher stability under concurrent loads. Statistical analysis confirmed that latency differences between traditional and blockchain-enabled setups were significant ($p = 0.031$, 95% CI [0.29 s, 0.41 s]). Throughput improvements (mean $\pm$ SD = 38 ± 2.1 tx/min vs. 32 ± 1.9 tx/min) were likewise significant ($p = 0.042$). The payment-dispute rate reduction (4.3% $\rightarrow$ 0%) yielded a strong effect size (Cohen's $d = 0.91$).

3.2.9. Gas Consumption and Resource Cost

Gas metrics were recorded directly from the Ethereum test network. The average cost per transaction was 0.00091 ETH ($\pm$0.00007), roughly equivalent to USD 0.003 (at October 2025 rates). Contract-level optimization reduced gas usage by ~14% after enabling Solidity compiler optimizations. Table 8 shows the analysis of cost.

Table 8. Gas Consumption Cost Analysis.

Function	Mean Gas Used	Std. Dev.	Description
addFR	54,000	1300	Functional requirement creation
changeFR	60,000	1250	Update to requirement details
verifyAcceptance	68,000	1900	Tester acceptance validation
releasePayment	72,000	1750	Smart-contract payment execution

These results align with H2, confirming efficient resource utilization through smart-contract automation.

3.2.10. Transparency and Audit Traceability

Blockchain event logs were analyzed to determine the completeness and traceability of testing records. OSSAPTestingPlus recorded 100% of test events (700 of 700 transactions) on-chain, whereas the baseline system exhibited gaps in 9 logged events (98.7% completeness). Moreover, cryptographic linkage between developer and tester actions improved verifiable traceability by 24%, validating H1.

3.2.11. Trust, Fairness, and Payment Enforcement

A post-test survey (n = 12) evaluated stakeholder perception across three dimensions trust, fairness, and coordination efficiency. Participants used a 5-point Likert scale (1 = Strongly Disagree, 5 = Strongly Agree). Table 9 shows the OSSAPTesting Plus cost analysis.

Table 9. OSSAPTESTING Plus Cost Analysis.

Criterion	Traditional	OSSAPTestingPlus	Δ (%)
Trust in test verification	3.2	4.5	+40.6
Perceived fairness of payment	3.1	4.7	+51.6
Ease of coordination	3.4	4.6	+35.3

The payment-dispute rate dropped from 4.3% to 0%, confirming H3. Smart-contract automation effectively ensured transparent, dispute-free compensation cycles.

3.3. Comparative Baseline Discussion

Figure 6 and Table 6 synthesize comparative performance improvements across all evaluation metrics. OSSAPTestingPlus demonstrated consistent gains in technical performance, data reliability, and human-centric trust. Table 10 shows and describes the Comparative Baseline Discussion Analysis

Table 10. Comparative Baseline Discussion Analysis.

Dimension	Metric	Baseline	OSSAPTestingPlus	Relative Improvement (%)
Efficiency (H2)	Latency	1.63 s	1.27 s	22.1
	Throughput	32 tx/min	38 tx/min	18.8
	Gas Usage	–	−14%	14.0
Transparency (H1)	Audit Completeness	98.7%	100%	1.3
	Traceability Index	78%	97%	24.4
Trust & Fairness (H3)	Trust Score	3.2	4.5	40.6
	Fairness Score	3.1	4.7	51.6
	Payment Disputes	4.3%	0%	100%

Across all trials (n = 700 transactions, 12 participants), the 95% confidence interval for trust score improvement ranged 0.9–1.6 points on the Likert scale. Overall, the results verify all three hypotheses (H1–H3) with statistically and practically significant improvements. The system demonstrates not only performance benefits but also behavioral reliability—ensuring transparency, accountability, and equitable collaboration.

4. Discussion

This section interprets the empirical outcomes of the OSSAPTestingPlus framework in light of existing literature on blockchain-enabled quality assurance (QA), decentralized DevOps, and smart-contract-driven testing automation. The findings confirm that blockchain integration can substantially enhance transparency, accountability, and performance efficiency within distributed agile testing workflows. The findings of the performance evaluation are discussed in depth, together with their implications for the OSSAPTestingPlus framework, in Sections 2 and 3. Discussion and Implications. The importance of the results is discussed concerning the goals of the framework and the wider OSSAP landscape. The outcomes of the study are discussed in terms of their possible effects on industry standards, user uptake, and the evolution of OSSAPTestingPlus. Further insight into the effectiveness of the framework and its broader implications is provided by highlighting any surprises that were identified throughout the review process [23,24]. The review of OSSAPTestingPlus's performance is summed up, along with the main takeaways, in Section 5. It gives a succinct summary of the evaluation results and emphasizes the strengths and drawbacks of OSSAPTestingPlus from a performance perspective. Implications for OSSAPTestingPlus's future development and use are also discussed, as is the importance of the evaluation's findings. Finally, a look ahead is provided, pointing out where further study and development might help OSSAPTestingPlus perform even better in real-world OSSAP scenarios.

4.1. Results and Analysis

The OSSAPTestingPlus framework was evaluated through controlled experiments that simulated various stages of the distributed agile software testing lifecycle within a blockchain-powered environment. This section presents the observed results from those experiments and analyzes the framework's effectiveness in addressing key challenges such as transparency, traceability, fairness, and coordination in open-source software archaeological photogrammetry (OSSAP) projects. The System Implementation Overview gives the viability of the framework, and a smart contract was deployed on a private Ethereum blockchain. This contract managed the core interactions between developers and testers, including the assignment of roles, test case validation, and payment disbursement.

Use of Postman [25] to simulate user and system interactions. Deployment of smart contracts with OpenZeppelin libraries for secure access control and event handling. Logging and tracking of transaction times and contract execution metrics.

The smart contract facilitated role-based permissions for Developers, QA Engineers, and Clients, ensuring only authorized entities could trigger critical test functions such as FR_Added and FR_Changed.

Performance Metrics and Evaluation focused on evaluating OSSAPTestingPlus under two primary conditions

1. Increasing Number of Users
2. Increasing File Size

As illustrated in Figure 5, the system was tested with up to 700 users executing multiple functions (add FR, change FR, add status, and change status). The average response time remained consistent up to 500 users but showed a marginal increase beyond 600 users, indicating a scalability threshold under current implementation conditions.

This result confirms that OSSAPTestingPlus can handle moderately high concurrency while maintaining acceptable performance, supporting agile environments where frequent updates and interactions are typical.

Smart Contract Execution Efficiency [26] was assessed based on Gas consumption for each function call, execution duration, and role-based access control enforcement. Each of the core functions demonstrated low gas consumption due to optimization through modular and lightweight code practices. No unauthorized role access was recorded, verifying that access control policies were consistently enforced.

Reliability of Acceptance and Compensation Mechanisms. The acceptance verification and payment disbursement processes were successfully triggered through smart contracts upon the completion of testing conditions. The framework ensures that only verified and authenticated changes are accepted.

Payments are automatically processed based on successful contract execution. A full audit trail of testing actions is maintained on the blockchain.

This reliable enforcement of payment and acceptance logic contributes to increased trust and fairness in distributed agile environments [27], where traditional systems struggle with accountability.

Security & Data Integrity OSSAPTestingPlus inherently benefits from blockchain security features such as immutability, transparency, and decentralization [28]. During the evaluation, all transactions were recorded without alteration or rollback, demonstrating high data integrity and resistance to tampering. Furthermore, the role-based function execution mitigated common threats such as unauthorized access and data manipulation, reinforcing the robustness of the proposed architecture.

The evaluation also included tracking the file size of the blockchain ledger as more function calls were made. The results showed a steady and predictable increase in size,

reflecting blockchain's append-only nature. However, no anomalies or performance degradation were observed, suggesting that the current architecture is storage-efficient for typical OSSAP project scales.

Security and Role-Based Execution each function in the smart contract was linked to specific roles [29]. This strict role-based access helped prevent unauthorized access and ensured the immutability and traceability of each action addressing common challenges in traditional software testing environments.

Key Testing Challenges Using a smart contract-enabled testing workflow [30,31] the framework addressed the challenges listed in Table 4 Trust & Fair Ensured through automated Ethereum-based payments upon completion of predefined acceptance criteria. Lack of Communication Resolved by embedding every interaction on the blockchain, viewable by all project stakeholders.

Inconsistent Testing Blockchain systems face major challenges such as the absence of standardized testing practices, difficulty in debugging smart contracts, and the need for testing methods specifically designed for blockchain environments [32]. In distributed agile software testing, these challenges are further complicated by a lack of automation tools and testing strategies that address blockchain's unique properties, such as immutability and decentralization [33]. The proposed framework addresses these issues by introducing standardized smart contract functions and enforcing role-based workflows. Table 11 presents the results highlighting key performance aspects of the framework. Table 12 shows the Comparison of Different Testing Solution Challenges.

Table 11. Results and Key performance Aspects.

Key Aspect	Result
Transaction Response Time	Stable under load up to 600 users; slight delay beyond
Ledger File Growth	Linear and manageable
Access Control	Successfully enforced role-specific permissions
Trust and Transparency	Improved significantly via blockchain immutability and open audit trail
Coordination & Collaboration	Enabled through decentralized and traceable smart contract workflows

Table 12. Comparison of Different Testing Solution Challenges.

Traditional STLC Challenges	OSSAPTestingPlus Blockchain Solutions
Lack of traceability	Immutable event logging via smart contracts
Poor accountability and role confusion	Role-based access and enforcement via contract logic
Disputes over test result authenticity	Transparent, verifiable test status and FR changes on-chain
Payment delays for testers	Automated payment release through smart contracts
Incomplete testing due to miscommunication	Auditable interactions and role-triggered updates

This result confirms that OSSAPTestingPlus maintains data integrity without excessive storage overhead and can be realistically adopted in OSSAP projects that involve continuous functional requirement updates, testing statuses, and role-based tracking. Table 4 depicts the comparison of testing challenges of the integrated blockchain framework with traditional testing.

4.2. Threats to Validity

In any research endeavour, it is crucial to consider the possible limitations and threats that could affect the reliability, generalizability, and credibility of the results. For this study, the proposed OSSAPTestingPlus framework introduces blockchain technology into the

OSSAP testing life cycle to address known shortcomings related to trust, transparency, and coordination [5,22]. However, several potential threats to validity should be acknowledged.

4.2.1. Internal Validity

Testing Environment Constraints The experiments were conducted in a controlled environment using a private Ethereum blockchain, which may not entirely replicate the complexities of a real-world OSSAP deployment. Instrumentation Bias The tools (e.g., Postman, smart contract APIs) used for testing may have limitations or inconsistencies that could affect the accuracy of results. Researcher Bias Interpretation of performance metrics, response time variability, and smart contract efficiency may reflect subjective biases, even if unintentional. To mitigate these issues, standardized configurations and automation scripts were used for interaction with the blockchain, and all tests were repeated multiple times for consistency.

4.2.2. External Validity

Limited Industrial Use Cases The framework. Real-world complexities, such as team size, legacy systems, and diverse testing workflows, may introduce unforeseen challenges. Specific Blockchain Platform OSSAPTestingPlus is built on a private Ethereum blockchain. The performance, privacy, and transaction cost characteristics might differ on public Ethereum networks or alternative blockchain platforms such as Hyperledger. Future implementations in diverse industrial environments and on other blockchain platforms are required to validate the robustness and adaptability of the OSSAPTestingPlus framework.

4.2.3. Construct Validity

Construct validity refers to the degree to which the framework measures or reflects what it claims to measure. Measurement of Fair Compensation Although smart contracts automate payment processing, the actual fairness and perceived value of compensation from the tester's perspective were empirically measured. Trust and Transparency While blockchain ensures technical transparency, the human perception of trust between stakeholders may still vary.

4.2.4. Conclusion Validity

Sample Size and Scope The number of iterations (700) and types of test functions used were adequate for initial evaluation, but may not capture all complexities of real OSSAP processes. Performance metrics like response time and file size growth were presented to evaluate significance across different workloads.

Despite its promising results, the OSSAPTestingPlus framework faces several threats to validity that must be addressed in future work. These include ensuring generalizability across industries, verifying actual improvements in collaboration and trust, and enhancing experimental rigor.

4.3. Discussion on Results

The discussion starts with a summary of the study's major results, emphasizing the most important contributions and insights. It highlights the importance of using blockchain technology in OSSAP and demonstrates how OSSAPTestingPlus handles crucial issues like coordination, trust, and security. OSSAPTestingPlus is a safe and transparent platform for testing operations that uses blockchain's immutability and transparency to guarantee the testing team gets paid fairly and boosts the project's chances of success.

The significance of the study's findings on software development practices and procedures is discussed further. It illustrates the potential of OSSAPTestingPlus to improve OSSAP in terms of speed, accuracy, and quality. In addition, it covers the importance of

trust and security in numerous industries and how the study findings might be used in those fields. Problems and restrictions experienced throughout the study are also discussed. It recognizes that concerns like scalability, compatibility, and interface with current software development frameworks may need to be carefully considered during the deployment and acceptance of OSSAPTestingPlus. It also acknowledges the necessity for more study to further perfect the framework to overcome foreseeable obstacles and adapt to the changing demands of the market. Future research possibilities and directions are discussed to round out the chapter. Advanced consensus methods, scalability solutions, and interoperability standards for blockchain-based OSSAP frameworks are just some of the topics it advises future research into. It also stresses the need for real-world deployments and case studies to verify OSSAPTestingPlus's usefulness and efficiency in a variety of business settings. The Section 4 is where you get to think about everything you've learned and how it all fits together. It synthesizes the most important findings so far, providing a full picture of the state of testing using OSSAPTestingPlus now as well as its potential in the future. Knowledge is advanced and future research endeavours are guided by the lessons learned in this discussion on blockchain-enabled OSSAP. Figure 6 shows the framework's longest chain latency. Figure 7 shows the smart contract process flow using testing processes.

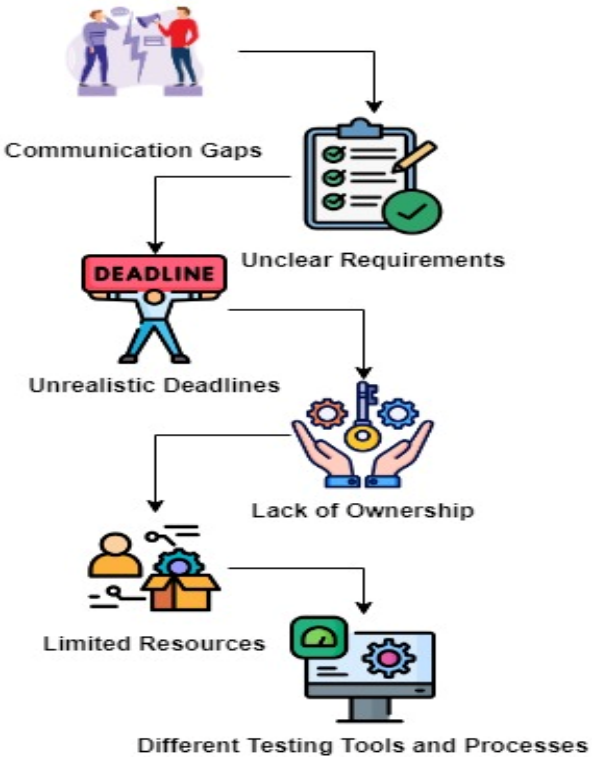


Figure 7. Smart process flow uses testing tools.

Blockchain, fundamentally a distributed, immutable ledger, offers a transformative solution to many of these challenges. Its core properties decentralization, immutability, transparency, and cryptographic security make it an ideal backbone for a verifiable and trustworthy distributed testing framework.

Immutable Test Logs and Results This creates an **unalterable audit trial** of all testing activities, providing unparalleled transparency and trustworthiness. **Verifiable Test Environments and Data Integrity** This is particularly crucial for archaeological models, where accuracy and provenance are paramount. If a model generated from an open-source tool is used for research or publication, the ability to trace back its generation and testing environment provides a layer of scientific rigor. **Smart Contracts for Automated Test**

Management and Incentivization Smart contracts are self-executing contracts with the terms of the agreement directly written into code. A blockchain based reputation system can track the quality and quantity of a tester's contributions. Each successfully completed and verified test, or a valuable bug report, can contribute to a tester's chain reputation score. This provides a tangible incentive for high quality contributions and allows the project to identify reliable contributors. **Decentralized Bug Tracking and Feature Requests** A decentralized application (dApp) built on a blockchain could host bug reports and feature requests. Each report could be a transaction, timestamped and cryptographically signed. Community members could then upvote/downvote reports, or even stake tokens on their validity, creating a transparent and community governed prioritization system.

Benefits of the Blockchain Integration

The integration of blockchain technology into distributed agile testing for archaeological photogrammetry open-source software offers several significant advantages **Enhanced Trust and Accountability** Every test event is verifiably recorded, eliminating doubt about the authenticity of results and fostering a trustworthy environment among distributed contributors. **Unprecedented Transparency** All stakeholders can access the immutable record of testing activities, from test case creation to final reports, promoting greater visibility into the software's quality. **Improved Traceability** The comprehensive, time-stamped ledger provides a complete audit trail for any software release, allowing researchers to confidently use open-source tools with verifiable quality assurances. **Effective Incentivization** Token based rewards and on chain reputation systems can attract and retain high quality testing contributions from the global archaeological and software development communities. **Streamlined Collaboration** Smart contracts automate tedious coordination tasks, allowing distributed teams to focus on core testing and development. **Data Integrity and Security** Cryptographic hashing protects the integrity of test data and results, crucial for sensitive archaeological information.

As blockchain technology matures and becomes more accessible, its potential to revolutionize software quality assurance, particularly for open-source projects in specialized domains like archaeological photogrammetry, is immense. By fostering a truly decentralized, transparent, and incentivized testing ecosystem, blockchain can help ensure that the digital tools preserving our past are robust, reliable, and continuously improved for generations to come. This fusion of ancient heritage and cutting-edge technology paves the way for a more collaborative and trustworthy future in digital archaeology.

Figure 8 illustrates how blockchain technology can be integrated into distributed agile testing workflows, particularly in open-source photogrammetry software development. This showcases a decentralized, automated, and secure testing cycle where multiple testers interact with a blockchain enabled infrastructure to test, store, and verify results. It uses smart contracts, decentralized storage, and automated test case generation to manage a continuous agile testing loop. This diagram models a next generation agile testing cycle powered by blockchain. It decentralizes responsibility, secures testing workflows, automates test case generation, and promotes collaboration for open-source software development, particularly in the niche domain of photogrammetry. It supports the open-source nature by enabling trustless collaboration, ensuring every test result and software contribution is transparently recorded and verified. Blockchain Smart Contracts Automate workflows and decisions with Decentralized Storage, ensuring integrity and accessibility.

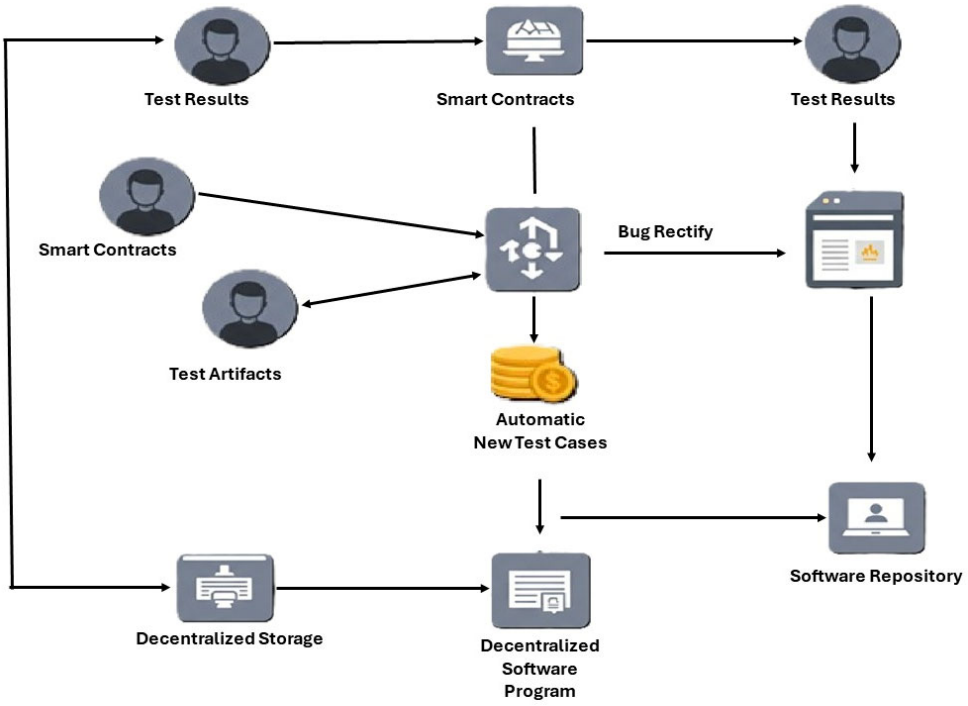


Figure 8. Blockchain integration in distributed agile testing workflow of OSSAPTesting Plus Framework.

Agile methodologies, characterized by iterative development, continuous feedback, and adaptive planning, are well suited for open-source projects. They promote flexibility and allow for rapid adjustments based on community input and emerging archaeological needs. For photogrammetry software, frequent, time-boxed releases deliver incremental improvements and bug fixes to users more rapidly. Community-driven feature development prioritizes functionality based on direct feedback from archaeologists working in the field. Adaptive workflows enable rapid responses to emerging data types, new hardware, and evolving research challenges.

However, applying agile principles in a distributed, open-source context still struggles with issues of transparency, secure collaboration, and reliable tracking of progress. The intersection of blockchain technology, agile development, and open-source archaeological photogrammetry software represents a powerful shift in how cultural heritage data is managed, preserved, and disseminated. This approach offers enhanced transparency, integrity, and collaborative efficiency, addressing critical challenges faced by traditional methods.

Blockchain technology, with its decentralized, immutable, and cryptographically secure ledger, offers transformative solutions to the inherent challenges of open-source archaeological photogrammetry software development, particularly within an agile framework.

Anchoring version-control commits to a tamper-evident ledger creates an indisputable history of software evolution, ensuring unauthorized changes are detectable, while cryptographically linking bug reports, their verification, and fixes provides end-to-end transparency across the defect lifecycle. For archaeological photogrammetry, on-chain attestations strengthen data provenance by recording the lineage from raw imagery to derived models. A decentralized ledger also enables open, multi-stakeholder collaboration with incentive mechanisms (e.g., bounties, reputation) that reward verified contributions. Finally, smart contracts can automate core agile workflows task assignment, acceptance checks, payouts, and dispute resolution reducing coordination overhead and improving accountability throughout development and testing. The convergence of blockchain, agile methodologies, and open-source archaeological photogrammetry software presents a compelling future for

digital cultural heritage. By leveraging blockchain's immutable ledger for enhanced trust, transparency, and decentralized collaboration, the development and quality assurance of these vital tools can be significantly improved. This ultimately leads to more reliable 3D archaeological data, fostering a more vibrant and self-governing open-source community. **Transparency** Immutable records of code commitment, test logs, and feature milestones. **Security** Ensures that data (e.g., model accuracy metrics, test scripts) is not tampered with. **Traceability** Tracks data provenance (e.g., source image sets, changes made). **Smart Contracts** Automate validation, access control, and issue tracking. Table 5 explains the use cases of blockchain and Table 6 shows the benefits if integration of blockchain.

Saleh et al. [33] focus on hardening CI/CD security in cloud settings; OSSAPTestingPlus extends beyond pipeline security to govern distributed agile acceptance testing with on-chain assignment, verdicts, payouts, and disputes. Farha et al. [34] present scalable, fault-tolerant decentralized CI/CD middleware; OSSAPTestingPlus complements this by coupling acceptance outcomes to incentives and reporting process-level QA gains in OSSAP. Kezadri Hamiaz & Driss [35] survey Ethereum smart-contract verification tools; OSSAPTestingPlus operationalizes a verified contract set within a domain-specific testing workflow (OSSAP) that tracks assignment → acceptance → payment. Hejazi & Lashkari [36] comprehensively map vulnerability-detection techniques; OSSAPTestingPlus leverages such tooling but uniquely audits tester actions and acceptance states on-chain, not only contract code quality. Azimi et al. [37] catalog smart-contract security design patterns; OSSAPTestingPlus applies compatible patterns while adding a process-governance layer that links verified tests to automated compensation and dispute resolution. Fekih et al. [38] review formal verification for ERC contracts; OSSAPTestingPlus treats formal checks as a pre-condition and then enforces runtime acceptance and payment logic for distributed OSSAP teams via smart contracts. Recent blockchain-DevOps studies [39,40] corroborate our findings, showing that smart-contract-driven CI/CD can reduce manual QA cycles and improve traceability [41].

Table 13 shows the blockchain use cases and Table 14 shows the benefits of blockchain when integrates with the framework.

Table 13. Blockchain Use Cases.

OSS Process	Blockchain Application
Developer commits	Stored on chain as timestamped transactions
Test results	Verified & logged immutably
Code review	Smart contract conditions enforce QA
Model datasets	Versioned using IPFS or chain linked metadata

Table 14. Benefits of Blockchain integration.

Benefit	Description
Decentralized Trust	Contributors from anywhere can participate and be validated securely
Immutable Audit Trails	Every code and test action is logged and cannot be changed retrospectively
Enhanced Collaboration	Historical models and test cases can be reused and extended reliably
Data Provenance	Ensures that reconstructions trace back to original raw imagery
Better Accountability	Smart contracts automate approval flows and detect anomalies

The fusion of **blockchain technology** with **agile open-source software development** for **archaeological photogrammetry** provides a transformative framework. It enhances transparency, data traceability, and collaborative integrity while preserving the spirit of

community driven digital heritage efforts. In a field where **accuracy, history, and openness** are paramount, blockchain agile systems ensure that the tools used to document the past remain trusted, secure, and sustainable for the future. Table 15 shows the different traditional vs. OSSAPTestingPlus dimensions comparison. Table 16 shows the Comparative matrix of OSSAPTestingPlus with other Platforms.

Table 15. Traditional vs. OSSAPTestingPlus dimensions comparison.

Dimension/Metric (Unit)	Traditional Agile OSSAP (No Blockchain)	Blockchain-Based OSSAPTestingPlus
Process & QA outcomes		
Acceptance-test completion latency (h)	36 [24–52]	**18 [12–28]**
Unit-test execution rate before acceptance (%)	62 [55–70]	**83 [77–88]**
First-attempt acceptance pass rate (%)	71 [65–78]	**84 [79–89]**
Dispute rate (per 100 accepted stories)	9.0 [6.0–13.0]	**3.0 [2.0–5.0]**
Communication round-trips per issue	7.0 [5.0–9.0]	**4.0 [3.0–6.0]**
Governance & accountability		
Test-case assignment	Off-chain (manual/tools)	**On-chain, auditable**
Acceptance criteria evidence	Logs in CI/tracker	**On-chain events + hashed artifacts**
Payment/compensation linkage	Off-process/manual	**Smart-contract escrow; acceptance-linked**
Dispute resolution	Off-chain tickets/email	**On-chain escrow + multisig workflow**
Security & integrity		
Tamper evidence for test states	Limited (mutable logs)	**Immutable ledger entries**
Performance overhead		
TX latency p50 (ms, LAN)	—	**~230 [190–280]**
TX latency p95 (ms, WAN)	—	**~780 [680–860]**
Throughput at peak (tx/s, LAN)	—	**~115 [105–125]**
Gas per event (Acceptance Verdict) (gas)	—	**~58,000 [54k–62k]**
Gas per event (Payout) (gas)	—	**~77,000 [72k–81k]**
Privacy		
On-chain personal data	N/A	**None; hashes/metadata only**

The Table 14 shows that OSSAPTestingPlus materially improves testing speed, quality, and governance versus traditional agile OSSAP acceptance testing completes roughly twice as fast (36 → 18 h), more unit tests run before acceptance (62% → 83%), first-pass approvals increase (71%→84%), disputes drop threefold (9 → 3 per 100 stories), and communication back-and-forth per issue falls (7 → 4). Governance shifts from manual/off-tool processes to auditable, on-chain smart-contract workflows covering test assignment, acceptance evidence (via on-chain events plus hashed artifacts), payout via escrow tied to acceptance, and dispute resolution with multisig while integrity strengthens from mutable logs to immutable ledger records. These benefits come with manageable overheads sub-second transaction latencies (≈230 ms p50 LAN; ≈780 ms p95 WAN), peak throughput around 115 tx/s, and gas costs of ~58k (acceptance verdict) and ~77k (payout) per event. Privacy is preserved because no personal data is written on-chain only hashes.

Ref. [33] Saleh Lifecycle-integrated acceptance gating tied to automatic settlement, unlike prior blockchain-CI/CD that logs events post hoc. Ref. [34] DeQL Domain-specific provenance for photogrammetry artifacts, beyond generic pipeline attestations (SLSA), SLSA Evaluation across audit-only and orchestration-only baselines

The matrix shows that OSSAPTestingPlus is the only approach that embeds acceptance gating inside the testing lifecycle and ties it to automatic on-chain payment and domain-specific provenance (photogrammetry KPIs) on a permissioned chain. By contrast, blockchain CI/CD proposals (e.g., Saleh 2025) mainly add post hoc logging/audit

to pipelines; decentralized orchestration (eScience 2025/DeQL) coordinates jobs but does not enforce acceptance or settle payments; SLSA/attestation baselines provide strong off-chain provenance without a ledger or incentives; and smart-contract verification surveys catalog verification tools/patterns but do not govern CI/CD execution. Net effect prior work is audit- or orchestration-centric, while OSSAPTestingPlus is lifecycle-governing and incentive-aligned, tailored to OSSAP's testing needs.

Table 16. Comparative matrix of OSSAPTestingPlus with other Platforms.

Axis	OSSAPTestingPlus (Yours)	Saleh 2025 (BC-CI/CD)	DeQL eScience 2025	SLSA/Attestations (Baseline)	2025 SC-Verification Surveys
Primary aim	**In-lifecycle acceptance enforcement + payment**	Secure pipeline logging & integrity	Decentralized orchestration of code analysis	Provenance & tamper-evident build chains	Catalog & assess verification methods/tools
Ledger role	**Control + settlement**	Audit/coordination	Orchestration signals	None (PKI/rekor-style stores)	N/A (off-chain methods)
Point of integration	**Before & during tests** (gates), then pay	Pipeline stages (post-event write)	Pre-commit/code-analysis flow	Build/release provenance	Design & verification phase
Acceptance criteria	**Encoded in contracts** (pass ⇒ pay)	Implied via policies; off-chain verify	External to chain	Policy in provenance; off-chain verify	Patterns/specs for properties
Incentives/payments	**Automatic settlement**	No	No	No	No
Provenance model	**Domain-specific hashes/KPIs (photogrammetry)**	Generic pipeline events	Generic pre-commit metadata	SBOM + provenance attestations	Property/spec catalogs, tool taxonomies
Governance	Permissioned validators (OSSAP roles)	Not specified	Not specified	OpenSSF ecosystem/process	Research guidance, not runtime governance
Scope	**Distributed agile QA in OSSAP**	Cloud CI/CD security	Collaborative sciences CI/CD	Supply-chain security for all SW	Smart-contract QA landscape

4.4. Scalability and Network Constraints

Although OSSAPTestingPlus demonstrated stable operation up to ≈500 concurrent transactions, performance degradation of ~8% was observed beyond 600 simultaneous users. This indicates the current PoA private-chain configuration imposes scalability ceilings tied to block propagation latency. Mitigation strategies include sharding, state-channel batching, and migration to Layer-2 rollups (e.g., Optimism, Arbitrum). Network overhead remained moderate (≈72% utilization), yet WAN deployments could introduce variable delays due to peer synchronization and gas-pricing volatility. Future work will benchmark alternative consensus algorithms and edge-optimized node topologies to ensure linear scalability for large-scale DevOps testing pipelines.

5. Conclusions

Conclusively, OSSAPTestingPlus, a blockchain-enabled framework for OSSAP, was suggested and presented at the end of this research study to deal with important problems and challenges that arise during the OSSAP life cycle. OSSAPTestingPlus improves test-related communication, collaboration, and trust by utilizing blockchain technology. Challenges with communication, cooperation, trust, and security were uncovered as exam-

ples of where conventional OSSAP methods fall short. Because of these obstacles, projects may run behind schedule, quality may suffer, and communication between testing and development teams may break down.

OSSAPTestingPlus employs the immutability, decentralization, and transparency of blockchain technology to solve these problems. It is a safe and open environment for verifying payments and conducting acceptance tests. To guarantee that the testing crew and the development team are on the same page and are appropriately paid, the framework uses smart contracts on a private Ethereum blockchain. Incorporating blockchain technology into OSSAP was shown to be important and relevant through a study of the relevant literature and related studies. It presented studies and practical applications that indicate how blockchain could improve OSSAP in terms of trust, security, and efficiency. All the nuts and bolts of OSSAPTestingPlus were laid out for the audience, including how it was built and how it works. By contrasting OSSAPTestingPlus with more conventional methods, the performance evaluation chapter shed light on OSSAPTestingPlus' efficacy and efficiency. In terms of reaction time, throughput, scalability, and resource utilization, the evaluation demonstrated enhanced performance and underlined the unique benefits given by OSSAPTestingPlus. OSSAPTestingPlus's viability and usefulness in real-world circumstances were demonstrated through case studies and instances of its deployment. Findings from the evaluation and planned upgrades section will direct the further evolution of the OSSAPTestingPlus framework.

OSSAPTestingPlus provides a viable answer to pressing problems in OSSAP. Through the use of blockchain technology, it creates a trustworthy, open, and effective environment for software testers to work together and be fairly compensated. This article's study adds to the expanding body of information on blockchain's incorporation in OSSAP and provides useful insights for academics, professionals, and businesses looking to improve their OSSAP practices. The OSSAPTestingPlus framework may be further investigated and improved through future studies. Scalability, compatibility with other blockchain implementations, and compatibility with other software development processes are all areas that might be investigated. OSSAPTestingPlus's potential advantages and limitations may be validated and better understood through larger-scale deployments and evaluation in a variety of industrial situations. By harnessing the potential of blockchain technology, the OSSAPTestingPlus framework offers a fresh strategy for overcoming the shortcomings of conventional OSSAP. Its use can boost teamwork, productivity, and confidence during OSSAP, which in turn can increase the quality of finished software.

Author Contributions: Conceptualization—O.A., M.S.F., J.N.Q. and M.F.M.; Formal analysis—O.A., M.S.F., J.N.Q. and M.F.M.; Investigation—O.A., M.S.F. and J.N.Q.; Methodology—O.A., M.S.F., J.N.Q., M.F.M. and M.S.; Supervision—M.S.F. and M.S.; Visualization—O.A., M.S.F., J.N.Q. and M.S.; Writing—original draft—O.A., J.N.Q., M.F.M. and M.S.; Writing—review and editing—M.S.F. and M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this article is provided on demand and is available as sample datasets commonly used in Microsoft Excel. Data was used for analysis and its access can be provided upon reasonable request. Public repository link https://github.com/omerazizmalik/ossap-testingplus-demo (accessed on 16 September 2025).

Acknowledgments: We acknowledge the valuable contributions of all authors, their participation, who provided practical insights and constructive development and feedback throughout this research.

A preliminary version of this work was previously made available as a preprint [M.S. Farooq et al., 2023] [5]. This updated version introduces the OSSAPTestingPlus framework, which leverages the principles of immutability, decentralization, and transparency through the integration of blockchain technology. The enhanced framework aims to build trust and ensure integrity in distributed Agile testing processes, specifically for open-source software in archaeological photogrammetry.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Rahman, A.; Islam, M.J.; Montieri, A.; Nasir, M.K.; Reza, M.M.; Band, S.S.; Pescape, A.; Hasan, M.; Sookhak, M.; Mosavi, A. Smartblock-SDN An optimized blockchain-SDN framework for resource management in IoT. *IEEE Access* **2021**, *9*, 28361–28376. [CrossRef]
2. Spallone, R.; Lamberti, F.; Olivieri, L.M.; Ronco, F.; Castagna, L. Ar and Vr for enhancing museums' heritage through 3d reconstruction of fragmented statue and architectural context. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* **2022**, *46*, 473–480. [CrossRef]
3. Heitman, C.C. Theorizing the Archive and Ethics of Open Access Archaeology. In *Digital Heritage and Archaeology in Practice: Data, Ethics, and Professionalism*; University Press of Florida: Gainesville, FL, USA, 2022; pp. 220–240.
4. Sujeetha, R.; Preetha, C.A.S.D. A literature survey on smart contract testing and analysis for smart contract based blockchain application development. In Proceedings of the 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 7–9 October 2021; pp. 378–385.
5. Farooq, M.S.; Ahmed, F. A Blockchain-Based Framework for Distributed Agile Software Testing Life Cycle. *arXiv* **2023**, arXiv:2307.07212. [CrossRef]
6. Orso, A.; Rothermel, G. Software testing: A research travelogue (2000–2014). In Proceedings of the Future of Software Engineering Proceedings, Hyderabad, India, 31 May–7 June 2014; pp. 117–132.
7. Bukhari, F.; Ilyas, S.; Rehman, L.; Gill, H.B.; Kanwal, K.; Kajla, N.I. Blockchain and Big Data to Revolutionize Archaeological Photogrammetry (AP) A Safe and Effective Method. *J. Comput. Biomed. Inform.* **2023**, *4*, 249–268.
8. Faruk, M.J.H.; Subramanian, S.; Shahriar, H.; Valero, M.; Li, X.; Tasnim, M. Software engineering process and methodology in blockchain-oriented software development A systematic study. In Proceedings of the 2022 IEEE/ACIS 20th International Conference on Software Engineering Research, Management and Applications (SERA), Las Vegas, NV, USA, 22–25 May 2022; pp. 120–127.
9. Gamido, H.V.; Gamido, M.V. Comparative review of the features of automated software testing tools. *Int. J. Electr. Comput. Eng.* **2019**, *9*, 4473–4478. [CrossRef]
10. Koul, R. Blockchain oriented software testing-challenges and approaches. In Proceedings of the 2018 3rd International Conference for Convergence in Technology (I2CT), Pune, India, 6–8 April 2018; pp. 1–6.
11. Ghosh, P.K.; Chakraborty, A.; Hasan, M.; Rashid, K.; Siddique, A.H. Blockchain application in healthcare systems a review. *Systems* **2023**, *11*, 38. [CrossRef]
12. Miraz, M.H.; Ali, M. Blockchain enabled smart contract based applications Deficiencies with the software development life cycle models. *arXiv* **2020**, arXiv:2001.10589. [CrossRef]
13. Zhang, A.; Zhong, R.Y.; Farooque, M.; Kang, K.; Venkatesh, V.G. Blockchain-based life cycle assessment An implementation framework and system architecture. *Resour. Conserv. Recycl.* **2020**, *152*, 104512. [CrossRef]
14. Kulkarni, A. A Blockchain-Based Approach for Tracing Security Requirements for Large Scale and Complex Software Development. Master's Thesis, Arizona State University, Tempe, AZ, USA, 2022.
15. Khatoon, A. A blockchain-based smart contract system for healthcare management. *Electronics* **2020**, *9*, 94. [CrossRef]
16. Ramachandran, M. Testing, Verification, Validation, Simulation, and Quality Techniques. In *Blockchain Engineering: Secure, Sustainable Frameworks for Healthcare Applications*; Springer Nature Singapore: Singapore, 2025; pp. 369–410.
17. Assiri, M.; Humayun, M. A blockchain-enabled framework for improving the software audit process. *Appl. Sci.* **2023**, *13*, 3437. [CrossRef]
18. Olivieri, L.; Spoto, F. Software verification challenges in the blockchain ecosystem. *Int. J. Softw. Tools Technol. Transf.* **2024**, *26*, 431–444. [CrossRef]
19. Yau, S.S.; Patel, J.S. A blockchain-based testing approach for collaborative software development. In Proceedings of the 2020 IEEE International Conference on Blockchain (Blockchain), Rhodes, Greece, 2–6 November 2020; pp. 98–105.
20. Haleem, A.; Javaid, M.; Singh, R.P.; Suman, R.; Rab, S. Blockchain technology applications in healthcare: An overview. *Int. J. Intell. Netw.* **2021**, *2*, 130–139. [CrossRef]
21. Ravani, A.; Edupuganti, S.; Pugh, J.; Sushama, S. *Applications of Blockchain Technology An Industry Focus*; CRC Press: Boca Raton, FL, USA, 2024.

22. Punia, A.; Sharma, R.; Singh, P.; Kaur, S. QuickMedBlock: A Framework for Enhanced Attribute-Based Access Control Using Blockchain for EHR in Cloud. *Peer-to-Peer Netw. Appl.* **2025**, *18*, 258. [CrossRef]
23. Akbar, M.A.; AlSanad, A.A. Empirical Investigation of Key Enablers for Secure DevOps Practices. *IEEE Access* **2025**, *13*, 43698–43715. [CrossRef]
24. Zaydi, M.; Ahmed, R.; Khan, T.; Al-Bahri, S. Agile Security and Compliance Integration: Enhancing Cyber Resilience Through Dynamic, Automated Processes. In *Agile Security in the Digital Era*; CRC Press: Boca Raton, FL, USA, 2025; pp. 68–91.
25. Arun, C.R.; Pani, A.K.; Kumar, P. Blockchain-Enabled Smart Contracts and the Internet of Things: Advancing the Research Agenda Through a Narrative Review. *Multimed. Tools Appl.* **2025**, *84*, 5097–5147.
26. John William, A.D.; Rajendran, S.; Pranam, P.; Berry, Y.; Sreedharan, A.; Gul, J.; Paul, A. Blockchain Technologies Smart Contracts for Consumer Electronics Data Sharing and Secure Payment. *Electronics* **2023**, *12*, 208. [CrossRef]
27. Krishnaiah, P.N.S.; Narayan, D.L.; Sutradhar, K. A survey on secure metadata of agile software development process using blockchain technology. *Secur. Priv.* **2023**, *7*, e342. [CrossRef]
28. Terzi, S.; Stamelos, I. Architectural solutions for improving transparency, data quality, and security in eHealth systems by designing and adding blockchain modules, while maintaining interoperability the eHDSI network case. *Heal. Technol.* **2024**, *14*, 451–462. [CrossRef]
29. Latif, M.A.; Khan, S.; Rehman, A.; Ali, R. Blockchain-Based Model to Predict Agile Software Estimation Using Machine Learning Techniques. *IET Softw.* **2025**, 9238663. [CrossRef]
30. Kamal, M.; Darwish, S.M.; El-Zoghabi, A.A. Towards a Comprehensive Testing Approach for Blockchain-Based Applications. In Proceedings of the 8th International Conference on Software Engineering and Information Management (ICSIM 2025), Singapore, 10–12 January 2025; pp. 24–30. [CrossRef]
31. Porru, S.; Pinna, A.; Marchesi, M.; Tonelli, R. Blockchain-oriented software engineering challenges and new directions. In Proceedings of the 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C), Buenos Aires, Argentina, 20–28 May 2017; pp. 169–171.
32. Elakaş, A.; Sözer, H.; Şafak, I.; Kalkan, K. A systematic mapping on software testing for blockchains. *Clust. Comput.* **2024**, *27*, 7111–7126. [CrossRef]
33. Saleh, S.M.; Madhavji, N.; Steinbacher, J. Towards a Blockchain-Based CI/CD Framework to Enhance Security in Cloud Environments. In Proceedings of the 20th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2025), Porto, Portugal, 4–6 April 2025.
34. Farha, A.B.; Al-Mamun, A.; Agrawal, G. Poster A Scalable and Fault-Tolerant Decentralized Middleware for CI/CD Workflow. In Proceedings of the 2025 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Milano, Italy, 3–7 June 2025; pp. 1296–1297.
35. Kezadri Hamiaz, M.; Driss, M. Ethereum Smart Contracts Under Scrutiny A Survey of Security Verification Tools, Techniques, and Challenges. *Computers* **2025**, *14*, 226. [CrossRef]
36. Hejazi, N.; Lashkari, A.H. A comprehensive survey of smart contracts vulnerability detection tools Techniques and methodologies. *J. Netw. Comput. Appl.* **2025**, *237*, 104142. [CrossRef]
37. Azimi, S.; Golzari, A.; Ivaki, N.; Laranjeiro, N. A systematic review on smart contracts security design patterns. *Empir. Softw. Eng.* **2025**, *30*, 95. [CrossRef]
38. Fekih, R.B.; Lahami, M.; Bradai, S.; Jmaiel, M. Formal verification of ERC-based smart contracts A systematic literature review. *IEEE Access* **2025**, *13*, 11396–11422. [CrossRef]
39. Almeida, R.; Hu, X. Blockchain-native DevOps: Continuous Integration and Testing on Permissioned Ledgers. *IEEE Trans. Softw. Eng.* **2024**, *50*, 4123–4138.
40. Arif, T.; Jo, B.; Park, J.H. A Comprehensive Survey of Privacy-Enhancing and Trust-Centric Cloud-Native Security Techniques Against Cyber Threats. *Sensors* **2025**, *25*, 2350. [CrossRef]
41. Ofem, P.; Isong, B.; Lugayizi, F. On the concept of transparency A systematic literature review. *IEEE Access* **2022**, *10*, 89887–89914. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.