inventions

MDPI

Article

Exploring a Blockchain-Empowered Framework for Enhancing the Distributed Agile Software Development Testing Life Cycle

Muhammad Shoaib Farooq [1], Junaid Nasir Qureshi [2], Fatima Ahmed [1], Momina Shaheen [3] and Sameena Naaz [3,*]

[1] Department of Artificial Intelligence, School of System and Technology, University of Management and Technology, Lahore 54000, Pakistan; shoaib.farooq@umt.edu.pk (M.S.F.); f2021114007@umt.edu.pk (F.A.)
[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; momina.shaheen@roehampton.ac.uk
* Correspondence: sameena.naaz@roehampton.ac.uk; Tel.: +44-(0)-20-8392-3180

Abstract

Revolutionizing distributed agile software testing, we propose BCTestingPlus, a groundbreaking blockchain-based platform. In the traditional distributed agile software testing lifecycle, software testing has suffered from a lack of trust, traceability, and security in communication and collaboration. Furthermore, developers' failure to complete unit testing has been a significant bottleneck, causing delays and contributing to project failures. Introducing BCTestingPlus, a transformative blockchain-based architecture engineered to overcome these challenges. This framework integrates blockchain technology to establish an inherently transparent and secure environment for software testing. BCTestingPlus operates on a private Ethereum blockchain network, offering superior control and privacy. By implementing smart contracts on this network, BCTestingPlus ensures secure payment verification and efficient acceptance testing. Crucially, it aligns development and testing teams toward shared objectives and guarantees equitable compensation for their efforts. The experimental results and findings conclusively show that this innovative approach demonstrates that BCTestingPlus significantly enhances transparency, bolsters trust, streamlines coordination, accelerates testing, and secures communication channels for all parties involved in the distributed agile software testing lifecycle. It delivers robust security for both development and testing teams, ultimately transforming the efficiency and reliability of distributed agile software testing.

Keywords: BCTestingPlus; software testing; blockchain technology; trust; transparency; sustainability

check for updates

Academic Editor: Ismail Fidan

Received: 9 May 2025
Revised: 23 June 2025
Accepted: 27 June 2025
Published: 30 June 2025

Citation: Farooq, M.S.; Qureshi, J.N.; Ahmed, F.; Shaheen, M.; Naaz, S. Exploring a Blockchain-Empowered Framework for Enhancing the Distributed Agile Software Development Testing Life Cycle. *Inventions* **2025**, *10*, 49. https://doi.org/10.3390/inventions10040049

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

In modern software development, overcoming challenges related to trust, transparency, coordination, and communication is crucial, especially as traditional approaches often contribute to project delays and failures. These issues become even more pronounced in today's globally distributed development teams that depend on efficient and reliable collaboration. Blockchain technology offers a transformative solution by addressing persistent problems such as mistrust, communication breakdowns, and inefficient scheduling that hinder conventional testing processes. This paper introduces BCTestingPlus, an innovative blockchain-based framework designed to support agile testing throughout the software

development life cycle. Leveraging the inherent strengths of blockchain decentralization, security, and immutability, BCTestingPlus enhances collaboration, transparency, and accountability in software testing. Unlike conventional methods, this framework enables secure payment verification and acceptance testing, ensuring fair compensation and fostering effective communication among developers, testers, and stakeholders.

By thoroughly examining the key features, advantages, and implementation aspects of BCTestingPlus, the study demonstrates how this approach can significantly improve software testing practices and elevate the overall reliability and quality of software products.

Blockchain technology offers a significant breakthrough in addressing persistent issues in software testing, such as a lack of trust, communication gaps, and scheduling inefficiencies. This paper introduces BCTestingPlus, a blockchain-based framework designed for agile testing across the software development life cycle. By leveraging blockchain's core strengths—transparency, decentralization, and security BCTestingPlus enhances coordination, ensures fair compensation through payment verification, and supports acceptance testing on a trusted, tamper-proof platform. This framework promotes seamless collaboration among developers, testers, and stakeholders. The paper provides an in-depth analysis of BCTestingPlus's features, benefits, and implementation, demonstrating its potential to improve testing effectiveness and elevate the quality and reliability of software systems.

Traditional software testing methods often struggle with issues of trust, communication lags, and poor coordination, leading to substandard results. As the limitations of these approaches become more apparent, there's a growing need for innovative solutions to enhance collaboration and efficiency across the software development lifecycle [1]. This research introduces BCTestingPlus, a novel blockchain-based framework designed to address these persistent challenges. By leveraging blockchain's strengths in authenticity, security, and biometric verification, BCTestingPlus aims to ensure transparent validation processes and fair compensation for all involved teams. The study emphasizes the framework's potential to reshape software testing practices, highlighting its value in fostering stronger inter-team relationships and improving the overall quality and reliability of software systems. It also explores an under-researched but critical aspect: how BCTestingPlus can guide structured, blockchain-aligned testing strategies.

1.1. Motivation

This study is motivated by the limitations of traditional software testing methods, particularly issues related to trust, coordination, and communication. These persistent challenges hinder the effectiveness of modern testing efforts, highlighting the urgent need for a more reliable and transparent approach. Inspired by prior research on blockchain's transformative potential, like presents a blockchain framework focused on enhancing scalability, security, and privacy. It explores technical strategies to build robust decentralized systems across various industries [2] and focuses on risk assessment, compliance, and transparency in financial and data auditing environments [3]. This paper introduces the BCTestingPlus framework, an innovative infrastructure that integrates blockchain's core strengths to improve software testing processes in distributed software development.

BCTestingPlus is designed to enhance transparency, collaboration, and fair compensation across development and testing teams. By offering a secure and verifiable platform for validation and payment confirmation, it addresses the communication and credibility gaps present in conventional systems. The ultimate goal is to raise the standards of software testing and reliability while fostering a more open, innovative, and cross-functional development ecosystem. This work will provide a different perspective of blockchain features, advantageous and practical issues at the expense of the conventional software testing bounded system by having an ocular inspection of BCTestingPlus. The purpose of

this is to get the entrepreneurs around the area to innovate and to create an ecosystem that is more open, productive, and cross-disciplinary.

1.2. Characteristics

The BCTestingPlus model leverages blockchain technology to enhance key stages of software testing by addressing persistent issues in traditional methods. Drawing from research such as blockchain enhances cybersecurity, trust, and privacy in digital ecosystems across various domains [4], and how a blockchain-driven framework improves transparency, accountability, and traceability [5]. The BCTestingPlus model applies blockchain's core strengths, particularly its transparency and immutability, to ensure every testing phase is securely recorded and verifiable. This not only improves clarity and trust among stakeholders but also demonstrates blockchain's adaptability to evolving market demands. Moreover, the aspect of [4] by which blockchain serves as a distributed and decentralized ledger is used to manage specialized information to ensure smooth and seamless coordination between parties. With the use of the BCTestingPlus framework, the developers' team, the testing groups, and other involved participants will be able to have fruitful interactions. Figure 1 shows the characteristics of Blockchain used in the framework. This diagram visually represents the key characteristics of blockchain technology, shown in a circular, interconnected format to emphasize how these features work together to make blockchain robust, secure, and reliable. Table 1 describes the features and characteristics of blockchain.

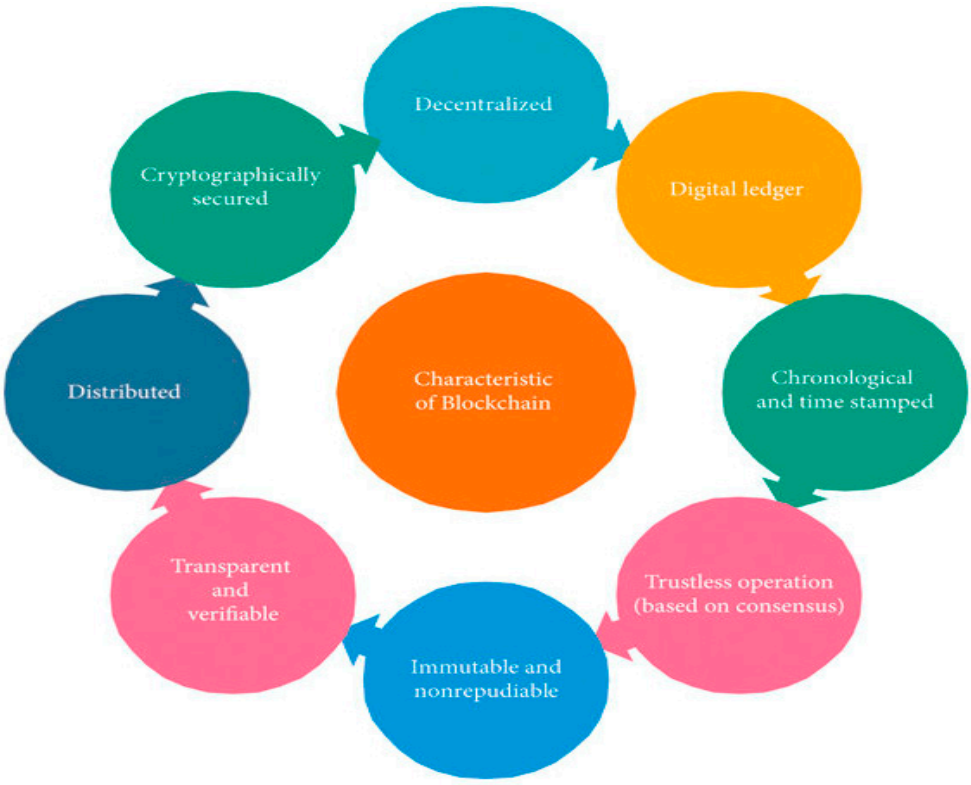


Figure 1. Characteristics of Blockchain Technology.

Figure 1 illustrates the core characteristics of blockchain, each of which plays a critical role in Blockchain Integration, meaning the process of embedding blockchain into existing systems or processes for enhanced performance, security, and transparency. Table 2 describes the blockchain-integrated characteristics.

Table 1. Features and Characteristics of Blockchain.

Feature	Description	Benefit
Decentralized	No central control	Increases fault tolerance
Digital Ledger	Maintains transaction history	Enables fast and digital recordkeeping
Chronological and Time Stamped	Transactions are time-ordered	Enables traceability and integrity
Trustless Operation	Based on consensus mechanisms	Promotes fairness and eliminates middlemen
Immutable and No repudiable	Cannot be changed once recorded	Increases trust and audibility
Transparent and Verifiable	Everyone can view and validate data	Promotes accountability
Distributed	Ledger is shared across multiple systems	Resilient and scalable
Cryptographically Secured	Uses encryption and digital signatures	Protects from unauthorized access and fraud

Table 2. Characteristics of Blockchain integration.

Feature	Role in Integration	Outcome
Decentralized	Removes central control	Resilience and uptime
Digital Ledger	Replaces outdated record systems	Fast, accessible data
Time-stamped	Tracks events accurately	Reliable audit trails
Consensus-based	Verifies data without intermediaries	Trustless execution
Immutable	Secures data from modification	Data authenticity
Transparent	Enables participant visibility	Reduced fraud and increased trust
Distributed	Share data across systems	Fault tolerance
Cryptographically Secured	Protects sensitive data	Security and privacy compliance

The Ethereum network (mainnet) is a public, permissionless blockchain. Anyone can join, read, write, or validate transactions. A "private Ethereum blockchain", A private blockchain that is Ethereum-compatible. Smart contracts: These are self-executing code blocks stored on the blockchain that automatically carry out predefined actions when certain conditions are met (like releasing payment when testing is complete and verified). Private Ethereum blockchain: This is a restricted version of the Ethereum blockchain, operated by a specific group or organization. Unlike the public Ethereum network, it is not open to everyone, offering greater control, privacy, and faster processing. So, when a system leverages smart contracts on a private Ethereum blockchain, it means it uses automated, trustworthy agreements running on a secure and permissioned Ethereum-based network to manage tasks like payments, validations, or workflows without needing a central authority or manual intervention.

A private Ethereum blockchain refers to a customized, permissioned version of the Ethereum protocol that is deployed and managed by a specific organization or consortium. Here's a breakdown of what it means. It uses Ethereum's core technology (smart contracts, Ethereum Virtual Machine, Solidity, etc.). "Private" means it is not open to the public; only authorized participants can access the network, send transactions, or validate blocks. Unlike the public Ethereum mainnet, a private Ethereum network is fully controlled, offering greater privacy, faster transactions, and customizable rules. Table 3 elaborates on the features of the private ethereum blockchain.

Table 3. Features of Private Ethereum Blockchain.

Feature	Description
Permissioned Access	Only invited nodes can join and interact with the network.
Faster Block Times	Can be configured to reduce delays in consensus.
No Mining Costs	Since it's a controlled environment, there's no need for energy-intensive mining.
Smart Contract Support	Supports Ethereum-native tools like Solidity, Web3.js, and MetaMask.
Data Privacy	Sensitive business data remains confidential among approved participants.

Furthermore, BCTestingPlus relies on the secure and effective communication features of blockchain, which help in the productive cooperation of everybody during the entire assessment life cycle. The blockchain is a technology that lacks trust components, such as decentralization and unchanging, and that is what makes the testing methods that we are used to so difficult. Ref. [6] talks about the inherent peculiarities of blockchain whereby precise monitoring and a guaranteed, immutable record are created for testing operations.

Providing complicated tests, improving test accuracy and reliability, protecting sensitive information, and improving the security of each stage of the testing life cycle. BCTestingPlus gains from this. In summary,

BCTestingPlus's built-in features that exploit blockchain capability, intensified by the referred articles, move in the right direction in overcoming the disadvantages of software testing [7].

1.3. Proposed Solution

The BCTestingPlus system is a strategy where the difficulties in the conventional method of assessment are solved by employing the revolutionary capabilities of blockchain technology. By using the BCTestingPlus approach, which is a decentralized, intact, and upfront solution, software testing responses to the planning, interaction waste, and reliability issues will be mandatory. BCTestingPlus takes the essence from the research outcomes of [4,8] on the challenge of identity and trust issues solved by blockchain, establishing a checkable, verifiable, and consensual log of labs' testing operations using blockchain's openness as the base. This transparency function is not merely visibility for the evaluation processes, but also ensures that the parties concerned and those who are being evaluated have consistent and trusted books of records.

BCTestingPlus, a cooperation phenomenon of blockchain, with its coordination function, solves the problems emerging from cooperation. The opposition of centralised solutions like those of a single developer or evaluation team can be done away with because of the distributed architecture of blockchain technology, with multiple parties involved, including developers, evaluators, and other stakeholder groups. The research that [6] gave out suggests that there is a direct relationship between the secure and effective transactions of blockchains and improved communication, since there is limited poor communication. Watching all the participating teams work together throughout the testing cycle and communicating gives BCTestingPlus an edge that is aimed at a smooth collaboration among various teams and participants. To address matters that relate to trust, BCTestingPlus is additionally utilising blockchain protection against tampering and decentralization. The members of this team are effective in the elimination of trust issues, which have brought about traditional testing techniques that are unsafe, unjust, and unreliable for credit authentication and acceptance, hence a secure and fair system for payment verification.

Besides, BCTestingPlus claims that documentation quality is immutable; accuracy and consistency are achieved by using the blockchain, which cannot be changed. The above feature ensures safe Software testing while assisting in the creation of a reliable and hack-proof testing record system, as it prevents any undesirable effects on Software

testing. To conclude, the approach presented by BCTestingPlus presents a robust approach for new software performance assessment. Its foundation is based on blockchain features, and it provides related support. BCTestingPlus is a software testing life cycle effectiveness enhancer as it addresses contact, connection, and trust-related problems that harm effective teamwork, transparency, and openness across the whole lifecycle. Consequently, it will lower the frequency of glitches and bring the software programmers into perfect working mode. The new Blockchain approach will improve the shortcomings of the old software testing methods, making them more developed. Blockchain strengthens test automation with the creation of automatic audit-quality reports of all the testing actions- cases, communication, test plans, and results. As a result of the fact that the blockchain is tamperproof, each testing procedure, as well as the official document, could be audited and verified. The integrity of the information is maintained because it cannot be deleted or changed. More and enhancing stakeholders' faith can lead to improvements in testing approaches and the identification of defects.

Blockchains have initially been used in crypto, yet they have become a ready, decentralized technology for a wide range of industries at present. Supply chain management, finance, or health care modules are often researching flexible blockchain technology to increase work performance, safety, and interaction. Software testing can potentially overcome barriers and become much more proficient by making use of blockchain technology that is specially designed for that purpose.

Blockchain technology has evolved significantly from its initial use in cryptocurrencies like Bitcoin. Today, it serves as a decentralized, secure, and transparent platform for a wide range of industries, including finance, supply chain, healthcare, legal, and software development.

1st Generation: Simple transaction ledger (no smart contracts).
2nd Generation: Introduced Smart Contracts.
3rd Generation: Scalability, Interoperability, and Energy Efficiency.

Smart contracts introduced in the second generation enabled automated execution of rules, drastically transforming software development and business logic. Unlike centralized systems, blockchain operates on a peer-to-peer (P2P) decentralized architecture, meaning there is no single point of control or failure. This has key implications for software development and business operations. Increased Trust and Transparency, every transaction is recorded on a shared ledger. Ensures accountability, especially in multi-stakeholder ecosystems like finance, logistics, or healthcare. Reduced Dependence on Central Authorities, no need for intermediaries (banks, notaries, centralized databases). Businesses can create automated, trustless workflows using smart contracts. Decentralized Governance and Collaboration, Code and updates are vetted by the community or consortium, not a single authority. Software development becomes more democratic and audit-friendly. Immutability and Security. Once data is recorded, it cannot be tampered with, reducing fraud risks. Software releases and patches can be traced and validated across all nodes. Table 4 shows the comparison of software development in centralized and decentralized blockchain-based architecture.

Table 4. Comparison Software Development in centralized and decentralized.

Centralized Software Dev	Decentralized Blockchain-Based Dev
Owned and deployed by one vendor	Shared responsibility among nodes or developers
Software updates pushed by central authority	Updates managed via governance models (e.g., voting)
Risk of data loss or breach if central server fails	Highly fault-tolerant due to distributed storage
Single point of failure	No single point of failure

Blockchain's decentralized architecture eliminates bottlenecks caused by central control, enhances transparency, and ensures security, all of which are crucial for modern software development and business operations. As smart contracts mature, they are becoming integral to automating business logic, fostering collaboration, and enabling trustless innovation.

The BCTestingPlus system is enhanced by the efficiency with which it consolidates software testing methods using blockchain technology. This approach resolves the limitations of conventional testing in the way that they do not fully reflect collaboration, openness, and eventually confidence. Through the use of blockchain technology, the BCTestingPlus platform is a transparent, public platform where participants and sponsors can accept test results and transfer payment. The interrelation of the roles is a condition that favours the development of good communication between the developers, testers, and other users.

The BCTestingPlus framework's purpose is to enhance the achievement of software testing. Also, the objectives are to engage the stakeholders and to compensate fairly for testing activities. The objectives are to be a reliable monitoring and verification mechanism. BCTestingPlus can boast its progress checks in real-time, smart contracts for auto-payment, and the technology blockchain for honest and fair testing operations as its main facilities. BCTestingPlus utilizes blockchain in all phases of software testing, including the manufacturing process, production process, distribution process, storage process, transformation process, and disposal process. With the help of blockchain technology, every tested process phase is trusted to be the only one to have the records of planning, generating, executing, and tracking defects. Smart contracts provide security to the system as well as allow for the automatic processing of financial transactions.

1.4. Traditional Approach and Blockchain-Based Approach

Traditional software testing methods, such as those managed through Jira or executed via Selenium-based automation, offer structured and scalable workflows. However, these approaches often fall short in key areas, particularly when teams are globally distributed. Issues such as limited traceability, insufficient audit trails, and a lack of trust among team members frequently arise. Centralized systems are prone to test case manipulation, inconsistent bug lifecycle tracking, and difficulties in maintaining compliance in regulated environments.

The proposed BCTestingPlus framework addresses these limitations by integrating blockchain technology, smart contracts, and IPFS (InterPlanetary File System) into the testing lifecycle. Through blockchain's decentralized and immutable ledger, every action, whether it's test case creation, execution, or bug resolution, is transparently recorded. This enhances trust among stakeholders and ensures test assets are securely stored and tamper-proof. Smart contracts enforce fair and automated test validations and payouts, removing manual intervention and fostering team accountability.

Performance metrics from the model indicate faster test cycle closures, improved traceability in bug resolution, and higher confidence in testing outcomes. Smart contract–driven result validation ensures objective judgment, while IPFS secures test artifacts in a distributed and immutable format, outperforming conventional repository solutions that often suffer from versioning conflicts or access limitations.

The BCTestingPlus model not only improves technical efficiency but also creates a collaborative environment where transparency, fairness, and reliability are embedded in the testing process, making it especially valuable for compliance-driven industries and agile, distributed teams. Table 5 compares traditional software testing methods versus blockchain-based methods. Figure 2 compares the traditional and blockchain-based software testing approaches.

Table 5. Comparison of Traditional Approaches vs. Blockchain-Based Approach.

Feature/Aspect	Traditional	Blockchain-Based
Test Data Integrity	Centralized storage, vulnerable to tampering	Immutable blockchain ledger ensures verifiable and tamper-proof test records
Collaboration in Distributed Teams	Requires centralized coordination, prone to miscommunication	Decentralized workflow with synchronized smart contracts
Test Case Auditability	Limited or external tooling needed	Built-in audit trails via blockchain transactions
Bug Tracking	Centralized bug trackers (e.g., Jira) with limited transparency	Blockchain-based bug logs with timestamp and traceability
Version Control of Test Cases	Requires manual configuration or third-party tools	Hash-based identification and versioning using IPFS
Security & Privacy	Prone to insider threats or unauthorized access	Encrypted records, access control via smart contracts
Real-Time Progress Visibility	Delays in reporting, manual updates	On-chain event-driven updates for real-time visibility
Cost & Complexity	Lower initial cost but high risk in long-term for compliance	Higher setup cost, but improved scalability and automation

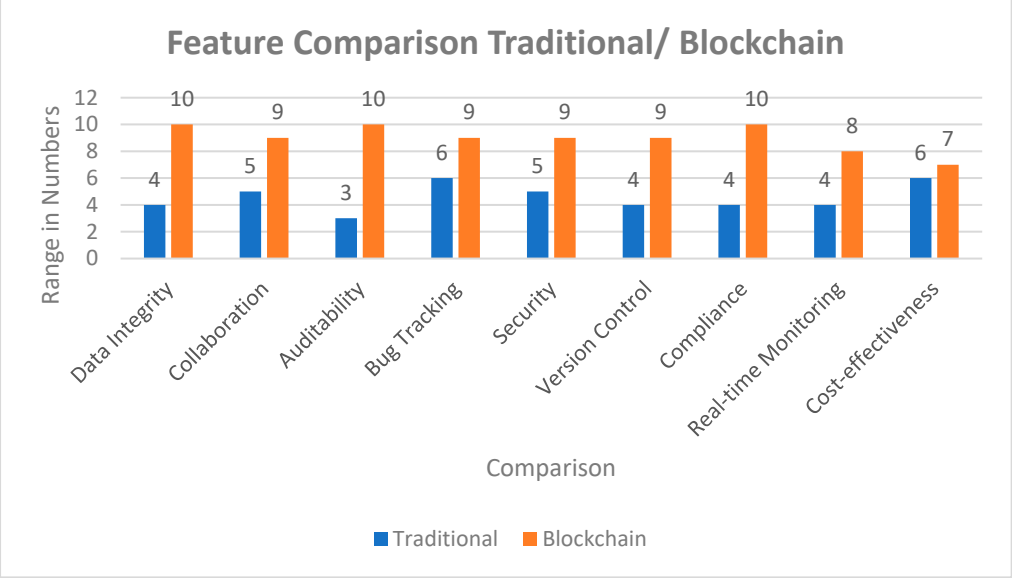


Figure 2. Feature comparison: Traditional approach and Blockchain-based approach.

1.5. Contribution

The study introduces BCTestingPlus, **a blockchain-powered framework designed to address long-standing issues in software testing, namely, trust deficits, inefficient coordination, and weak communication among distributed teams. By integrating the core principles of** decentralization, transparency, and immutable verification, the framework reshapes conventional testing practices, enabling collaborative environments that are more equitable and reliable.

Unlike traditional models that often create silos between testers, developers, and stakeholders, BCTestingPlus fosters seamless collaboration, incentivized contributions, and trustworthy validation through smart contracts on a private blockchain. The system also ensures fair compensation and provides traceable testing logs, reducing discrepancies and enhancing the overall software development life cycle.

This research bridges the gap between theory and practice by offering both academic insights and real-world applicability. It expands the role of blockchain technology beyond financial domains, demonstrating its potential to revolutionize software engineering workflows. By tackling current industry pain points and showing measurable improvements in testing transparency, efficiency, and quality, BCTestingPlus sets a foundation for future innovations and widespread industry adoption.

Ultimately, the study contributes significantly to academic discourse and professional practice, encouraging broader exploration into blockchain-driven quality assurance in software development.

1.6. Related Work

The effectiveness of software development centers on the quality, reliability, and performance of applications, factors assessed during rigorous testing phases. The rise of decentralized technologies, particularly blockchain, has introduced both challenges and unprecedented opportunities in the realm of software testing. This paper presents a focused review of existing literature, highlighting the evolving intersection of blockchain and software testing. It critically examines traditional proof-of-competence mechanisms and underscores the growing necessity for unified, tamper-proof systems like BCTestingPlus. Since Satoshi Nakamoto's 2008 introduction of blockchain as a decentralized, trustless ledger, the technology has proven capable of transforming centralized systems through cryptographic security, consensus protocols, and data immutability. Building on these principles, BCTestingPlus leverages blockchain's core strengths to overcome long-standing issues such as trust deficits, poor traceability, and inefficient collaboration in software testing workflows [3,9].

1.7. Fundamental Ideas

As highlighted in [10], blockchain achieves decentralized consensus through mechanisms like Proof of Work (PoW) and Proof of Stake (PoS). While PoW relies on computational effort to validate transactions, PoS assigns validation rights based on user stake, enhancing energy efficiency. The evolution of smart contracts, first conceptualized by Nick Szabo and later popularized by Vitalik Buterin via Ethereum, enabled self-executing agreements without third-party intervention. These technologies, combined with blockchain's core attributes such as immutability and cryptographic security, form the foundation of BCTestingPlus. By leveraging these features, BCTestingPlus introduces a secure, tamper-resistant environment for managing software testing workflows, ensuring that once test results are validated, they remain traceable and unalterable. This enhances trust, eliminates manipulation, and streamlines collaboration, addressing critical issues that have long plagued traditional testing environments.

1.8. Blockchain Uses

According to [11], blockchain has significantly reshaped the financial services industry by enabling fast, secure, and transparent data exchanges, with applications in areas like international payments, trade finance, and remittances evident in platforms such as Ripple and Hyperledger. Similarly, blockchain has enhanced traceability and accountability in supply chains, allowing companies like Walmart and IBM to reduce fraud and improve logistics efficiency [12].

Moreover, in healthcare, blockchain ensures secure sharing of patient data among authorized parties, addressing critical privacy and interoperability issues. Projects such as MedRec demonstrate how blockchain can be used to safeguard electronic health records and clinical research data [13]. Furthermore, the technology enables decentralized identity management, giving users greater control over their personal data, illustrated by platforms

like uPort and Sovrin. At the core of these advancements are smart contracts, automated, tamper-proof agreements that enforce rules without requiring intermediaries, lowering costs and improving compliance.

In contrast to these domain-specific applications, BCTestingPlus extends blockchain's benefits to the software testing domain, a field long affected by trust issues, miscommunication, and lack of transparency. By integrating smart contracts and decentralized validation processes, BCTestingPlus offers a novel solution that automates testing workflows, ensures verifiable test results, and incentivizes fair tester contributions. This positions BCTestingPlus as a transformative framework, much like the impact seen in finance, healthcare, and supply chains, bringing similar levels of integrity, automation, and efficiency to collaborative software testing.

1.9. Coordination Issues in Software Testing

Software testing plays a critical role in ensuring the reliability, performance, and security of applications by identifying defects and verifying that systems align with user requirements. Traditional testing involves activities like test case design, execution, bug tracking, and reporting. However, coordination challenges, especially in distributed teams, often hinder its effectiveness. Past research underscores these limitations but lacks mechanisms for improving trust, transparency, and collaboration across testing workflows. The BCTestingPlus framework addresses these persistent issues by leveraging blockchain's decentralized and tamper-proof infrastructure. Through features like smart contract–based validation and immutable result logging, BCTestingPlus introduces a reliable, transparent, and collaborative environment for modern software testing, especially in agile and distributed development contexts.

Blockchain technology is a decentralized and distributed digital ledger that securely records transactions across a network of computers. Each entry, or block, is cryptographically linked to the previous one, creating a chain of records. Once a block is validated by the network using a consensus mechanism, the data it holds becomes immutable, meaning it cannot be changed without altering all subsequent blocks, making it highly tamper-resistant. It means that blockchain technology is decentralized, immutable, transparent, secure, and distributed in nature.

Studies to pinpoint the main coordination concerns in evaluating software and consider how blockchain technology might be used to help with the issues. One of the major coordination problems while testing software is communication gaps. Productive evaluation of software depends on successful interaction. Nonetheless, research continuously draws attention to the existence of rapport gaps between various testing process clients, which include testers, development personnel, and project executives. To close the disparity between those crucial roles, [14] highlights the significance of communicating well.

The coordination issues in Software testing, particularly regarding the fair and transparent distribution of testing tasks among quality assurance QA engineers, are effectively addressed through the proposed framework, leveraging blockchain's decentralized nature. This approach removes biases and guarantees a fair opportunity for contributors by autonomously allocating Test Cases to QA Engineers based on established criteria. As self-executing protocols, Smart Contracts are essential because they effectively distribute Test Cases according to factors such as workload, availability, and knowledge. This reduces the need for manual intervention, simplifies task distribution, and lowers coordination overhead. Additionally, Smart Contracts designate the state of Test Cases for functional requirements automatically, allowing for insight into testing progress, real-time status tracking, and prompt intervention in the event of delays or bottlenecks. Status updates are guaranteed to be verifiable and reliable due to the tamper-resistant nature of blockchain.

The precision of results given may be relied upon by QA Engineers and project stakeholders, hence fostering trust in the testing procedure and the overall quality of the software testing. The researchers pointed out that they are afraid that the lack of interaction is affecting the process of recommendation, which may ultimately cause the reduction of standards for this software product. One burden is raised during the execution of Software testing, which is the issue of allocating resources. Resourceful use of the available resources should be the cornerstone of organized and in-depth testing. Ref. [15] reviews the way resource distribution additionally influences fault-identifying effectiveness.

Synchronization of testing resources is a key part of planning that determines the appropriate allocation of resources to prevent situations where, somewhere at some point, there would be either a deficit or oversupply of resources, thus ensuring that testing activities occur at different levels of the process. To make sure the testing of software goes smoothly, environment management is one of the core coordination topics. Though assessments are at the heart of computer-based instruction, setting them up is another cooperation challenge, to install and put in place the computer hardware and software that is needed for a proper evaluation. Concerning the processes of software development, the [16] thesis prioritizes proper management by a skilled leader in the testing environment. Communication over the running dates of the test may be delayed because of improper coordination of that sector, which sometimes affects the project time and willingness to succeed. Closely related to these issues are the ones that happen during the time of integrating and debugging the software products. Crosstalks and software module/component dependencies can erupt into major organizational problems during software development that are more prevalent in large-scale projects than in small-scale ones.

According to [17], the complexities of reliance are highlighted in their discussion of industry experiences with requirement management. In this situation, coordination issues may make it more difficult to manage the assessment of integration, which could postpone finding and fixing errors. Blockchain Technology to Solve Coordination Issues Developed initially for cryptocurrency programs, blockchain technology has developed into a flexible answer that may be used to solve synchronization issues in several other industries. A thorough analysis of blockchain's use in other fields can yield substantial insight regarding how it might be used in software validation. According to [18], it highlights the technology's capacity to keep tabs on items in real time. The same techniques might be applied to testing software to increase traceability, making it easier for investors to monitor the development of testing operations and determine challenges. Blockchain has also already been used in the medical field to overcome coordination concerns about data comparability. The potential for the use of blockchain technology in biological and medical industries has been examined. By developing a decentralized, safe platform for exchanging test results and information, these ideas might be used in software testing to foster cooperation between multiple testing organizations and users. According to [19], using electronic agreements on blockchain systems, like Ethereum, may simplify and automate intricate procedures.

The researchers highlight smart contracts as a crucial component of blockchain technology that allows for execution itself. By automating coordination chores like issue resolution workflows, smart contracts can be integrated into software evaluation processes to decrease the need for human interaction and speed up the Verification cycle. Furthermore, blockchain-based identity authentication layers offer individuals to have control over their identity data in a decentralized format. In addition to this, the decentralized web is built around the idea of the decentralized identity [20] concept. To make sure that both the user with adequate exposure to testing facilities and the results receive the management part of access, solving problems, and access management could be avoided using the same principles for end user privileges and software testing permits, which determine the outcomes.

1.10. Transparency in Software Testing

It is the principle that implies clarity and transparency for all of the approaches from the very beginning of the testing cycle to the end. This review of the literature covers the issue of transparency in open-source software appraisal, paying attention to the role and contribution that this problem makes to the system. The openness and opportunity for people to gather the information that relates to software testing activities, including lifting work, artefacts, and performance results, is known as transparency in software testing. It simply means that the involvement of the testing team of which includes developers, testers, and project managers are included will be full of understanding of the testing process. Disclosure inordinately spreads across sectors of evidence, reporting, and documentation, creating an environment where knowledge is clear and easily accessible.

Disclosure and reporting are two main channels of transparency. In summary, by provisioning test descriptions, defect tracking, and enhanced test reports, the position of user is updated about the status of tests, weaknesses, and results. The extremely important task of being explicit and organized during test-driven development is clarified by Ivan et al. [21]. They also make the point that accurate note-taking and other instructiveness are what make the process of testing easy. Since cooperation is a primary virtue of open-source software in general, software testing should focus on it heavily. Transparent and free testing of the code, which brings together the different programming teams, is the testing protocol to be followed. Achieving effective communication in no way eliminates the goals and the progress of testing among all the parties concerned, and enables them to solve problems collectively. To enumerate, somehow why this researcher [22] believes that sharing information is a vital ingredient of an organization that can deliver the software promptly and promote teamwork. The extended transparency characteristic of software testing has a dual benefit of reputation and believability. Transparency becomes a key to making the organization trustworthy. It is mostly the people who find tests more credible when they can quickly establish the procedure for testing, and also at the same time, its outcome. In the context of software outsourcing relations, the value of trust [22] puts stress on the role of transparency, which is key for confidence creation and preservation. Software testing transparency also allows specific dilemmas to be recognized early. The advantage of such testing procedures, which are easily open, is the screening of mistakes and challenges in the early stages. The more effectively the bugs are resolved in the early stages, the fewer the delays and the problems, which get costlier and take longer to resolve in the latter stage [22] show us how well-timed solutions as well as making early corrections can have a favourable and profitable effect on a company's financial health. Also, the performance can be much better through continuous improvement, which is yet another benefit of the process of openness in the software testing. Transparency lays open the path to continuous development because it provides the experience utility of a definite procedure, tool, or strategy to control the effectiveness. Through past flaws evaluation and preparation for future exploits can be accomplished once the data is in a clear and readable form. The findings of [21] are significant, and they specifically show the vital role of practical experiments in software engineering. On top of that, the study also highlights the need to build trust and get the correct data all the time for improvement. The most significant challenge concerning gaining openness in the process of Software testing is, yet, the over-exaggeration of metrics. The case can be that the firm's bias in assessment methods might substitute the focus on some quantitative indicators that ignore qualitative aspects. Researchers such as [21] stress the necessity for an appropriate approach that takes into account qualitative as well as quantitative measurements and warns against seeing measurements as a magic bullet. Another key problem is a lack of standardized reporting. Transparency may be hampered by discrepancies in information terminologies and forms.

Standard reporting procedures must be established to guarantee efficient communication with stakeholders. In their discussion on the necessity of the creation of software evidence, ref. [21] stressed the significance of standardized procedures in empirical investigation.

1.11. Benefits and Drawbacks of Using Blockchain

Testing plays a crucial role in ensuring the quality, stability, and functionality of software programs. Blockchain technology has created new possibilities as well as issues for testing software. This review of the literature examines the relationship between software testing and blockchain technology. The paper summarizes research findings to highlight the benefits and drawbacks of different approaches. It emphasizes the importance of a comprehensive structure like BCTestingPlus. Over time, the software testing sector has seen substantial changes as a result of the necessity for comprehensive quality assurance. Ref. [23] conducted an in-depth analysis of software testing research from 2000 to 2014. Test case preparation, test execution, and automated evaluation were all part of the inquiry. They advocated for a move from manual testing to automated procedures that save time and labour. The review is open to the core issues of software testing in layer-blockchain systems as well as blockchain technology integration. The software testing enterprise is having a hard time dealing with the blocked machine system that has been launched recently [24]. The study of scholars was concerned with discovering the nature of the problem and proposed solutions for the testing of blockchain-based software. The ability to authenticate and immutably preserve data as well as certify the smart contracts, ensure wide interoperability of systems working in multichain, and ease of solving scalability problems are some of the blockchain technology problems. In this regard, it is worth mentioning a study that describes the issues related to software testing challenges [24]. The preparedness of software testing to face up to the challenges is undermined, so a proper study is required with a modified system. Using blockchain technology in software testing may increase traceability, security, and integrity. Researchers developed a blockchain-based testing technique for collaborative software development, an approach to enhance trust and coordination in collaborative software development to improve information sharing, requirement compliance checking, and overall efficiency in software development among diverse teams [25]. The team emphasized the significance of teamwork in testing, and they used blockchain technology to improve interaction and collaboration among testing teams. Researchers focused on the evaluation and evaluation of smart contracts. The investigation focused on the safety, efficiency, and productivity of smart contracts [26].

Thorough testing is crucial for ensuring the safety and trustworthiness of smart contracts. Although their research contributes to our understanding of smart contract testing, it does not provide a comprehensive testing methodology for all software. Previous software testing methods for blockchain technology have been identified as having shortcomings. Ref. [27] developed the Smart Block-SDN system for managing IoT resources by combining blockchain with SDN architecture. Software testing is crucial for managing IoT resources, but it is often overlooked in blockchain-based systems. Their paradigm prioritizes resource management without considering software testing challenges and requirements. Recent evidence on applications of blockchain offers some insights into the impact it brings. In particular, although their studies mainly focus on blockchain uses in healthcare, we miss a well-designed testing method to estimate the impact. Blockchain-based healthcare models or autonomous healthcare systems could also be for Open-Source Software Platforms. Further research is needed in software testing to assess the quality, reliability, and security of healthcare data stored on blockchain. Researchers like [28] investigated the limits of SDLC models in blockchain-enabled smart contract applications. The research was termed the SDLC-based methodology for developing smart contract applications through blockchain

technology. SDLC Methods that are applied in blockchain-based projects show their deficiencies in hybrid contract tests and security concerns. Indeed, while the proposed method is used for solving problems in the development, it still does not involve specifics of testing in the blockchain environment. Ref. [29] draws attention to the lifecycle assessment of the whole technology with the use of blockchain. The paper presents the blockchain method and system design aimed at boosting the integrity and visibility of the LCA manufacturing processes. The investigation does not specifically address software testing challenges in blockchain-based life cycle analysis. Continuing exploration of software testing and testing frameworks is crucial to pave the way for the future of global trade. This existing research has shown that blockchain's ability to track security problems can help solve the issues of this complicated system development initiative. As they are implemented by their strategy, they like handling blockchain technology use for security provision in the furnished SDLC. The study did not address the unique challenges and elements to consider when testing software in this scenario. Blockchain technology has the potential to improve security testing and certify compliance with security requirements. However, further research is needed to establish the best way to incorporate it into testing methods [30].

A thorough framework is necessary to address the unique problems and requirements of software testing for blockchain applications, as current methods are limited. In this high-tech era, the BCTestingPlus framework is proposed by the researchers to positively affect the issue of test anxiety [31]. BCTestingPlus places relevance on communication, partnership, belief, and security while integrating Blockchain Technology into software testing operations. BCTestingPlus employs the main features of blockchain as tamper-proofness, transparency, and decentralization, to lay the foundation for safe payment and acceptance testing and payment verification. The platform ensures fair compensation for testers and fosters trust between testing and development teams by automating the execution of defined norms and agreements through smart agreements.

BCTestingPlus's key benefits include improved openness and traceability. Testing processes and findings are recorded on a blockchain, which is auditable yet cannot be altered. Transparency boosts participant confidence and makes it easier to validate test methods and results. The developer and tester teams may collaborate more easily thanks to BCTestingPlus. Smart contracts make it easier to assign testing tasks and guarantee equitable payment according to predetermined parameters for each party. Automation has sped up testing while lowering miscommunication and conflict.

Furthermore, BCTestingPlus enhances test safety by utilizing the inherent security features of blockchain technology. Because of the immutability and security of the blockchain, test results cannot be altered or changed. More confidence in the calibres of software is a result of increased testing reliability and precision. BCTestingPlus is useful for a variety of corporate objectives. The proposed research could be used in the development of blockchain technology for healthcare systems and for enhancing data privacy, accuracy, as well as physician-nurse communication. The BCTestingPlus system has the added value of making management more transparent through the verification of accurate distributed ledger transactions. This contributes to the trust and reliability factors of the chain.

This literature review highlights two key factors influencing software testing: technology and blockchain. While existing methods demonstrate a degree of productivity, the BCTestingPlus framework adopts a more comprehensive approach. By leveraging blockchain technology, BCTestingPlus enhances the efficiency, security, and transparency of software testing processes. It effectively addresses specific issues related to defects in application frameworks. The ultimate goal is to build more reliable and resilient systems. The continued advancement of blockchain and technological innovation positions BCTest-

ingPlus as a forward-looking solution, potentially necessitating a significant overhaul of current information architectures.

1.12. Organization of the Article

The remaining paper is organized as follows: In Section 2, the proposed framework is shown in Section 3 contains the implementation, results and performance. Furthermore, Section 4 describes the conclusion and future work.

2. Materials and Methods

This section highlights the preliminaries for the proposed framework with materials and methods. The major components that will be used in this framework will be described in this section. Blockchain is a decentralized ledger system that offers a safe, transparent, and permanent record of transactional data between two parties [25]. Through the use of cryptographic techniques, it produces a record of transactions that cannot be altered. This record takes the form of a chain of blocks, each of which references the block that came before it.

Smart Contracts are code snippets designed to execute versatile tasks [26]. They are kept on a blockchain. They make it possible to execute complicated transactions in a way that is tamper-proof, transparent, and efficient, which could potentially reduce the need for intermediaries.

The need for software integrity protection becomes obvious in an ever-changing digital environment. The main dynamics in most traditional software testing models are the failures to inculcate trust, transparency, collaboration, and communication, and this, in turn, leads to ineffective and subpar processes and products. BCTestingPlus uses a blockchain approach to provide a multi-agent system that can be used for software testing. BCTestingPlus shows a reliable and teamwork manner of creating a dependable and transparent Software testing ecosystem by using blockchain technology. This is illustrated below. Literature narrates that blockchain can be used to audit software testing processes and make them more accurate and reliable [5]. Another feature of blockchain deployment is that it creates a possibility of a traceable permanent record of testing facilities and results, which it hard to make a mistake, and also allows data tinkering. BCTestingPlus smart facilities include developers and their testers, and manage to sign developers and testers effectively and efficiently. The role that this feature plays in reducing discordancies and miscommunication only enriches mutual testing cooperation. It is also important to note that blockchain technology could facilitate accountability and oversight of Software testing. To keep the process agile and smooth, BCTestingPlus allows for instant communication and collaboration between developers and testers, resulting in shortened delays and increased efficiencies. My goals and the benefits of the testing program are discussed in the introduction of this paper. It also covers the reasons that make BCTestingPlus more effective compared to other software. An additional role played by BCTestingPlus is the utilization of blockchain technology in aiding companies to find the right equilibrium in terms of software testing. It combines different elements so that a reliable and effective software testing platform is achieved. The distributed ledger system, the smart contract agreements, the testers, as well as the developers are some parts that are part of these smart contracts. Another part of smart contracts is the location where they store the test results and samples.

Although blockchain technology guarantees that the test results will be accurate, precise, and confident, we cannot overlook such concerns any longer. Craftily, the private Ethereum blockchain system on which BCTestingPlus is based generates chances to do without outsiders, by allowing keeping records and testing validation in a decentralized way. Testing-plus will handle the enforcement and enactment of smart contracts in an

automatic manner. With the use of blockchain smart contracts, the process of collaboration between designers and engineers will be controlled. Smart contracts will ensure smooth cooperation during the test cases running, result verification, and compensation calculations. It can be suggested as a great solution to such problems. Concerning software testing, the integration of blockchain technology into the BCTestingPlus platform has many benefits. Given [3], blockchain technology helps build trust and reliance by maintaining and restoring transparency and security. Blockchain technology facilitates everybody's involvement and transparency in testing operations and results recording with the use of an open, chainable ledger (unchangeable).

In addition to that, the distributed nature of Blockchain enables the agreements that are decentralized and guarantees that all participants have equal access to the verification data. Transparency implies an increase in the truth and a decrease in the falsehood, which would in turn reduce the probability that incorrect information could be brought into use. The study talked about the value of transparency in such projects is very important, and such projects, other than the one party participating, have a greater number of parties. Chain technology ensures that a tamper-proof audit trail is available for the records of testing procedures and findings at all times [4]. Furthermore, developing a copy of the record concerning audit findings allows entities to detect the source of the problem, assess their progress towards a solution, and ensure the application of local laws and regulations. In this respect, studies thus far have shown that the crypto feature of blockchain enhances software quality and reliability through full-fledged traceability at the testing stage. BCTestingPlus functions as mutual communication from development to quality assurance teams that will lead to meeting every issue in real time. Transparency and teamwork are the inevitability that happens from good communication and, in consequence, improve software testing quality as well. Blockchain technology is employed by BCTestingPlus, which is one of the software testing frameworks to enhance efficiency, traceability, and transparency. Blockchain networks provide even compensation, test automation systems, and build trust, and as a process, those factors leverage the best quality as the end-product. BCTestingPlus is focused on providing the blocks and blocks interacting with each other in software testing Procedures through a flexible framework. The system's architecture, which aims at providing swift communication, security, scalability, and data integrity to testers in a fast-paced environment, is an effective means of achieving the objectives. Figure 3 depicts the proposed framework, blockchain testing plus BCTestingPlus, and how the framework works.

Ethereum is a public digital currency model to uses the Virtual currency Ethereum distributed ledger. Typical of the public blockchain networks are the privacy and control that are only found in private ones. Running smart contracts by applying Ethereum as the platform for the blockchain also ensures their future interoperability with contemporary smart frameworks and resources. The private blockchain network contains several nodes that are used to register and record test transactions as they pass through. The decentralized architecture of the network lets it function without a centrally located control point and thus makes it less vulnerable to data loss, manipulation, and corruption. Networks of private self-sustained ledgers that are interconnected with other systems prevent the peeping of private data for all accredited users.

In this way, BCTestingPlus is a tool to help coordinate the different parts of the software testing that work alongside one another. The architecture enables them to have a different line of interaction and collaborate to some extent in the drive for joint development and testing. Such channels facilitate the process of collaboration by giving updates on testing operations such as pre-test advice, timely notification of test results, and consultation. Blockchain technology facilitates transparent communication between testers and

developers. The blockchain records all testing actions, including executing test cases and confirming results, creating an indestructible record that cannot be changed. This transparency builds trust and accountability among everybody participating. BCTestingPlus prioritizes customer information security and privacy. Blockchain's distribution protects test data against unauthorized changes. The distributed ledger network's use of cryptographic methods and consensus procedures enhances data security and prevents unwanted acts. We protect sensitive testing information using access limits as well as cryptography. Restricting access to the blockchain network and testing the repository ensures the security and privacy of data. Blockchain technology can enhance data security and protection for various businesses, including software development and testing. BCTestingPlus' structure prioritizes scalability and speed to meet evolving software testing requirements. The private blockchain network is designed to handle and record numerous test transactions efficiently and reliably. Implementing sharding and optimization strategies can improve scalability in blockchain-based systems. BCTestingPlus can handle a high volume of test cases, outcomes, and contacts without compromising effectiveness. BCTestingPlus' design guarantees credibility, safety, flexibility, and effective communication among stakeholders in the testing of software. BCTestingPlus's encrypted blockchain system, readily interconnected segments, rigorous data integrity inspections, and scalable design provide a strong basis. Figure 4 depicts the layered architecture BCTestingPlus framework, and Figure 5 shows how blockchain technology works and supports the framework.

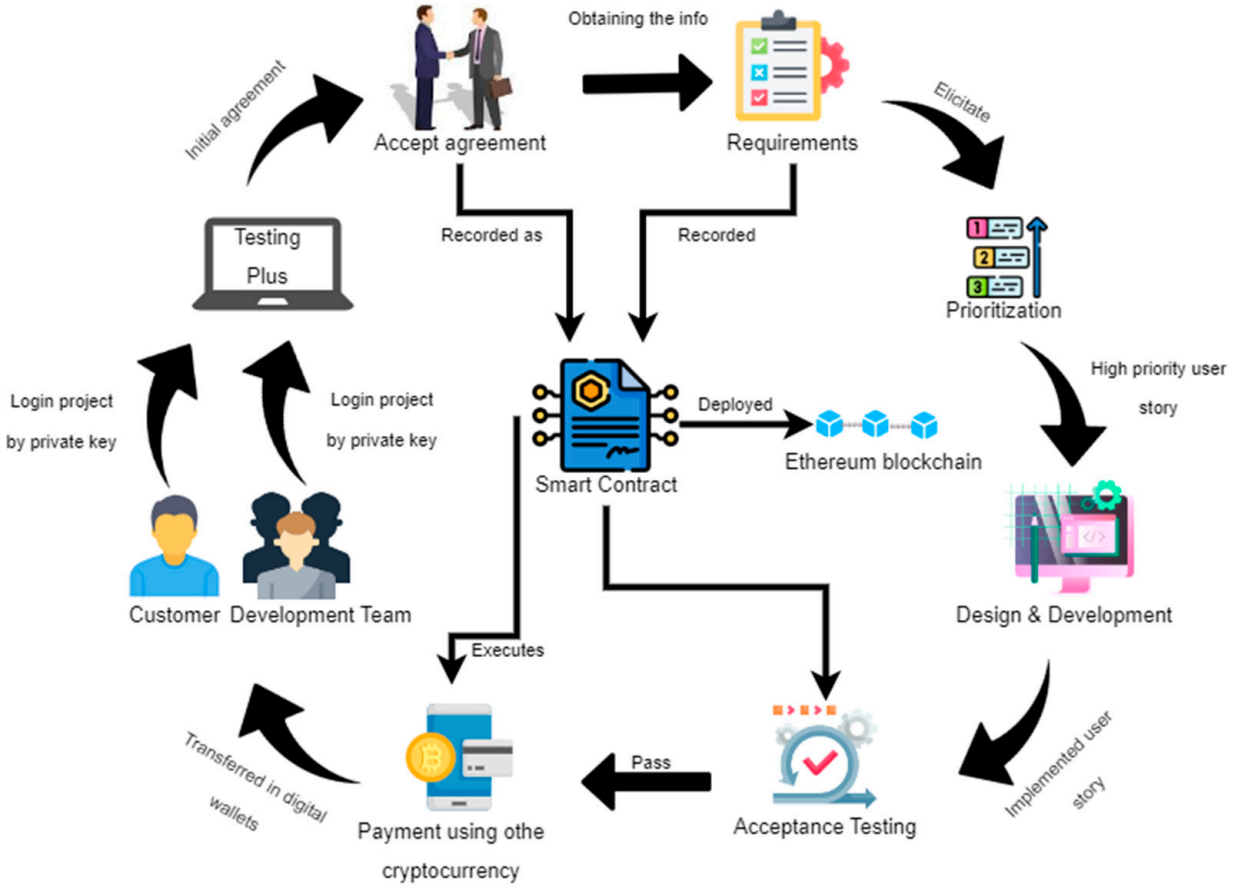


Figure 3. Proposed BCTestingPlus Framework.

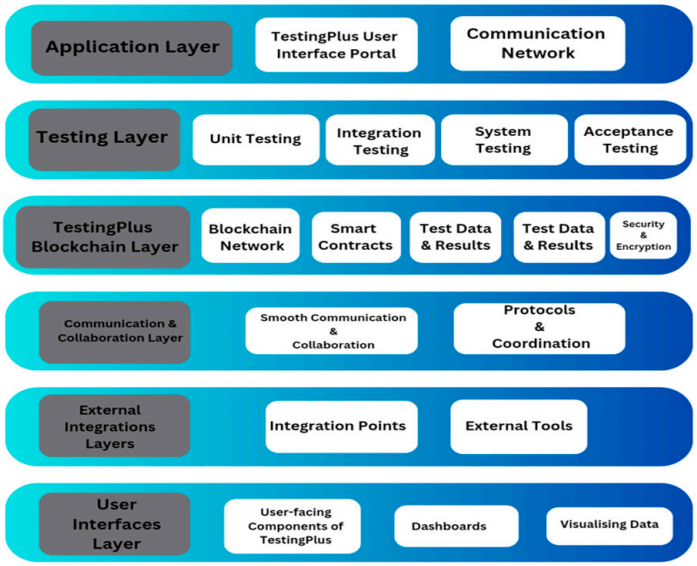


Figure 4. BCTestingPlus Framework Layered Architecture.



Figure 5. Blockchain Technology.

2.1. Creation and Repository

The testing team creates test cases to store them in a repository at the beginning of the procedure. These cases specify the precise scenarios and conditions that need to be verified to ensure the software's quality and performance. BCTestingPlus has a central repository, and this facilitates the availability of test cases to both the developers and the testing team.

2.2. Framework Algorithm

The algorithm comprises two smart contracts: Privacy Policy, User Agreement, and Developer Agreement. Moreover, there is a public variable called Customer Address used in the Customer Agreement contract that holds the ETH address of the customer and a testing fee variable that stores the cost of testing services. The customer address is set to the sender's address of the message, as it is within the constructor. It performs a setTestingFee method to which only the Customer address (statistically, the deploying address) could be used to set the testing cost. Moreover, the web page covers the Developer Agreement contract with the variable "reward" that plays the role of storing the awards, and the public variable "developer address," which is used to store the Ethereum address of the developer. Constructor runs with the developer address set as the message sender's address while the constructor. The contract also has the function of setReward, through which the developer can express the amount of the reward. But it is only the developer who can use the function at the address that belongs to the first code of deployment.

Figure 6 explains the smart contract working in the framework, diagram illustrates the life cycle of a Solidity Smart Contract from creation to deployment on the Ethereum blockchain, highlighting key stages: compilation, bytecode/ABI generation, and deployment, clearly visualizes how a Solidity smart contract is transformed from human-readable code into a live, executable entity on the Ethereum blockchain. The process ensures secure, decentralized, and programmable logic that can be trusted and verified publicly.

```
Customer's Agreement
contract CustomerAgreement {
address public customerAddress;
uint public testingFee;
constructor() public {
customerAddress = msg.sender;
}
function setTestingFee(uint fee) public {
require(msg.sender == customerAddress, "Only customer can
set the fee");
testingFee = fee;
}
}
```

```
Developer's Agreement
contract DeveloperAgreement {
address public developerAddress;
uint public reward;
constructor() public {
developerAddress = msg.sender;
}
function setReward(uint amount) public {
require(msg.sender == developerAddress, "Only
developer can set the reward");
reward = amount;
}
}
```

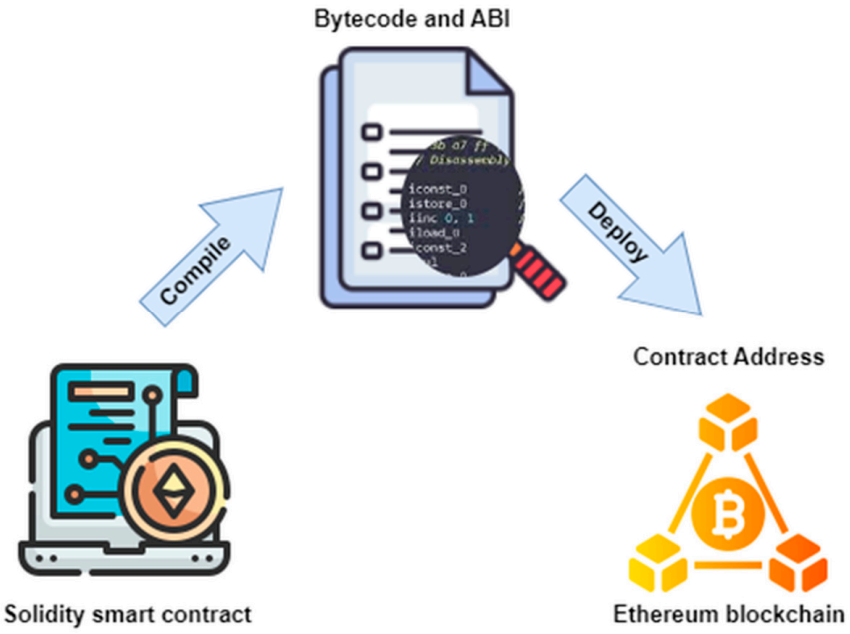


Figure 6. Smart Contract working in BCTestingPlus Framework.

Customers can decide for themselves whether to increase or reduce the TestingFee factor in the CustomerAgreement contract, to determine the cost of testing services. Moreover, the DeveloperAgreement contract has three attributes, which are a reward variable and the developer's address; thus, the developer will figure out what amount of money they have

earned for their testing work. Smart contracts create more analogous transparency and, with that, improve the efficiency of the contracting procedures for everyone involved. The BCTestingPlus framework guarantees adherence by preventing changes to the terms and conditions of contracts once deployed on the blockchain.

2.3. Evaluation and Validation

After running test cases, the data is examined for reliability. This testing technique compares predicted and actual results from test cases. BCTestingPlus's smart contracts automate verification, eliminating the requirement for human involvement.

2.4. Feedback and Cooperation

BCTestingPlus's methodology prioritizes communication and collaboration. The framework enables immediate access to testing data for both testers and developers. Stakeholder cooperation increases the efficacy of the programme by offering suggestions, addressing problems, and offering input.

2.5. Compensation and Incentives

BCTestingPlus is unique in the way it employs smart contracts. The usage of these technologies notably depends on the number of test instances that will be produced, the accuracy of the outcomes, and the overall value they add to the testing processes. Smart contracts automatically determine the testing team's compensation. This guarantees fair compensation for the testing personnel and motivates them to produce high-quality work. Previous research highlights how crucial it is to have a software testing workflow that is well-defined and includes traceability, transparency, and efficient communication. The blockchain technology used by BCTestingPlus allows for better function and activity records non-compromised and verifiable by every stakeholder of the testing operation. Simplifying software testing and increasing team productivity are some of the benefits of BCTestingPlus' flow work. The organizations providing trusted test results and finally launching better software are other benefits of this workflow. The voting process would be a lot more useful if testers agreed on the same methods of open-source software development and promoted it automatically. One striking feature of BCTestingPlus is that it uses blockchains. These tools help enhance efficiency, transparency, and dependability in software testing. This section is devoted to the enumeration of the most important advantages and undesirable implications of the use of the BCTestingPlus system.

2.6. Accountability and Traceability

BCTestingPlus takes the notion of transparency to the next level with blockchain technology in guaranteeing that all tests will be stored in the blockchain record. Transparency could help stakeholders to observe testing processes, and they can view results and draw conclusions about the steps taken. Along with this, letting stakeholders view the findings of tests enables the capability to create conclusions about how reliable and accurate the software can be. BCTestingPlus's blockchain technology is traceable and makes it possible for various actors involved to trace the source of problems, the time it took to resolve the problem, and the individual responsible. An audit trail that improves accountability, as well as a function to quickly spot issues and their fixes.

2.7. Interaction and Cooperation

BCTestingPlus is one of several initiatives aimed at strengthening collaboration between software testing and development teams. The framework focuses on bridging communication gaps among stakeholders to enhance problem reporting and resolution. By fostering an environment of open dialogue and active teamwork, it minimizes miscommu-

nication and delays. This cooperative approach not only accelerates issue identification and resolution but also significantly improves software quality. Implementing BCTestingPlus improves collaboration and communication between testers and developers. Promoting open communication and involvement are the intended outcomes of the BCTestingPlus collaboration between testing and development professionals. The system serves as the communication bridge among different members and allows them to exchange their ideas, work together, and update them on the test results immediately. In such a gateway environment, it is easy to apply various customs, decreasing the chance of misinterpretation and also reducing the time delays. Communication in software testing assists in the identification, evaluation, and resolution of issues and the product quality improvement rate. Having implemented the BCTestingPlus, cooperation and communication between testers and designers are as usual.

2.8. Rewards Management

BCTestingPlus leverages smart contracts to ensure fair and performance-based compensation for the entire testing team. Payments are dynamically calculated based on test volume, result accuracy, and individual contributions, fostering motivation and active participation. By promoting fairness, trust, and accountability, BCTestingPlus enhances both team morale and software quality. Its integration offers a promising path to reducing risk and boosting efficiency in open-source software projects. Continuous evaluation of its effectiveness is essential for refining its impact in evolving development environments.

2.9. Evaluation Criteria

BCTestingPlus efficiency and usefulness may be examined using various criteria. These factors include accurate and consistent test results, efficient testing, clear stakeholder communication, and satisfaction from all parties involved. Existing frameworks present evaluation frameworks that consider these elements, providing insight into measuring the effectiveness of blockchain-based software testing frameworks.

2.10. Future Enhancement

BCTestingPlus can bring some advantages, though some things can be improved. Another vital component of blockchain technology is scalability. Because the number of tests goes up, so does the need to secure the blockchain handling and processing these larger amounts of data and transactions. A scalable blockchain network is one of the solutions to start coming up with the issue.

BCTestingPlus can stand out by making its AI and ML technologies more powerful and efficient. AI and machine learning steps could reduce the time needed for testing by administering tasks such as test generation, anomaly detection, and performance optimization. AI and ML could be drawn into several Open-Source Software (software testing) and their interaction with blockchain platforms. The BCTestingPlus reward systems will be a work in progress as the organization performs research and continues to develop new ways of rewarding arriving students. Employees working at BCTestingPlus find more motivation to participate in the workflow and a higher level of engagement, which, combined with a high degree of alignment of the interests of the testers with the overall goals of the project, helps to achieve better team performance. BCTestingPlus will capitalize on the evaluation and improvements that will be achieved in the next few years for success. To enhance BCTestingPlus, the evaluation criterion can be checked, the research findings can be expanded, and more ideas can be introduced.

The introduction of blockchain technology to software testing by BCTestingPlus is a major move towards renewing the sector, which has had many difficulties. BCTestingPlus implements blockchain technology to provide transparency and traceability, which in turn

increases confidence among the stakeholders regarding the testing procedures and the results. Collaboration and communication features of BCTestingPlus have been designed to help bridge the gap between testing and development teams. If the result is successful, quicker issue resolution will lead to a better and higher-quality product.

Using the contracts in BCTestingPlus, the better the time and effort taken from the testers and the better the result they deliver, the more they will be rated, thus boosting their loyalty. Blockchain uses technology that is much more stable and reliable than these applied testing circumstances. BCTestingPlus has been very preferred by many companies as a part of their conventional toolkit, based on feedback from users as well as installation trials. Research demonstrates that blockchain technology can facilitate transparency, accountability, cooperation, and the solving of crises. Now, to BCTestingPlus, we need to improve scalability, insert AI and ML algorithms, and also shift towards better incentives. Developments to BCTestingPlus will benefit software testing enterprises as they mature, and the market starts developing in the national space. BCTestingPlus blockchain-based technology cloud Wie software testing into the solution sphere, defying transparency, credibility, and trust. Collaboration and communication. Ultimately, successful implementation brings all the stakeholders together, liberates developers from bug fixing, the software gets further improved, and involves parties in achieving better teamwork. Implementing BCTestingPlus leads to significant improvements in the development of software reliability and efficiency.

2.11. Implementation and Performance

In this section, we have tested the efficiency of the framework to demonstrate its effectiveness in a real-world scenario.

2.12. Performance Assessment and Evaluation

To test the efficiency and performance of our proposed model, we utilized specific tools to implement and test a blockchain network. The importance of BCTestingPlus performance goal setting is explained. Through a close look at BCTestingPlus's inner structure, this article shows its key strengths and weaknesses. The most important element is how vital the key performance indicators framework is. In this case, the purpose of the investigation is to develop common metrics for analyzing the behaviour of BCTestingPlus, in what situations it is faster or slower if its user base can grow, and how it uses resources in the system. This is a testing way that figure out whether the system works well, is efficient, and works as expected. This analysis involves integrating all capabilities of the system.

2.13. Proposed Acceptance Test

```
contract AcceptanceTest {
address public customerAddress;
address public developerAddress;
uint public testingFee;
bool public isTestCompleted;
constructor(address _customerAddress, address
_developerAddress, uint _testingFee) public {
customerAddress = _customerAddress;
developerAddress = _developerAddress
testingFee = _testingFee;
}
modifier onlyCustomer() {
require(msg.sender == customerAddress, "Only customer can
```

```
initiate the acceptance test");
_; }
modifier onlyDeveloper() {
require(msg.sender == developerAddress, "Only developer
can complete the acceptance test");
_; }
function initiateTest() public payable onlyCustomer {
require(msg.value == testingFee, "Testing fee should be
paid");
// Perform necessary actions to initiate the test
// This could include triggering the execution of the test cases
isTestCompleted = false;
}
function completeTest() public onlyDeveloper {
// Perform necessary actions to validate the test results
// This could include checking the output, comparing with
expected results, etc.
isTestCompleted = true.
//Transfer the payment to the developer as the test is
successfully completed
payable(developerAddress). transfer(testingFee);
}}
```

The basin code is compiled, which then creates the contract Test and records emails of the customer and developer, tests cost, and returns a bool variable. Firstly, the consumer should enter the charging by making use of the charge Test() method and then proceed to introduce Test(). When the instructed method is executed, the developer calls the complete Test () function, and so the test is recognized, and by doing this, the payment is transferred. This will be possible with smart contracts as the payments will move to the developer with no further intermediaries. Figure 7 shows the latency in accessing the longest chain, showing Latency in Accessing the Longest Chain plotted as Time in milliseconds (ms) against Block numbers. Each dot represents the latency recorded for accessing a particular block on the blockchain.

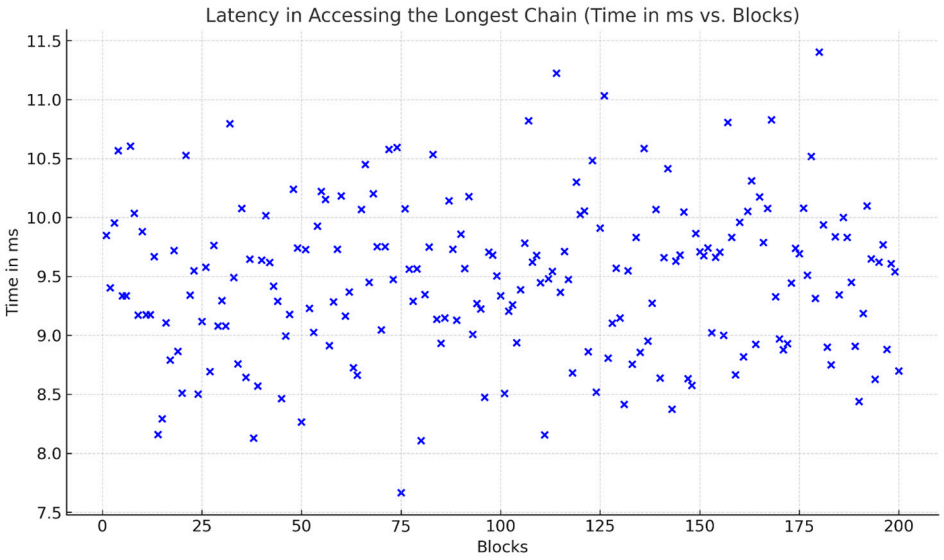


Figure 7. Latency in accessing the longest chain.

2.14. Performance Evaluation Results

In this part, we were devoted to the analysis of the performance metrics of BCTesting-Plus, its successes and weaknesses will be evaluated, and the effectiveness of the tool will be assessed. This segment details the experimental analysis of BCTestingPlus in real-world scenarios, focusing on its performance. BCTestingPlus'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 increased with each new block and transaction added to the blockchain. According to Figure 8, 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 the 590 to 600 blocks range, cause a linear increase in file size. The block transaction count wasn't predetermined during development, which slightly enlarges the size of newly mined Blocks. Figure 8 depicts the average growth in the size of blocks and size in KB. In our analysis, we updated all the current chains in the BCTestingPlus 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. The flow of the data illustrates the variability in execution times for these HTTP requests. This variability is attributed to the blockchain's decentralized structure, involving numerous nodes without central authority over data. The diverse server capabilities within the BCTestingPlus 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 performance evaluation results and their comparison.

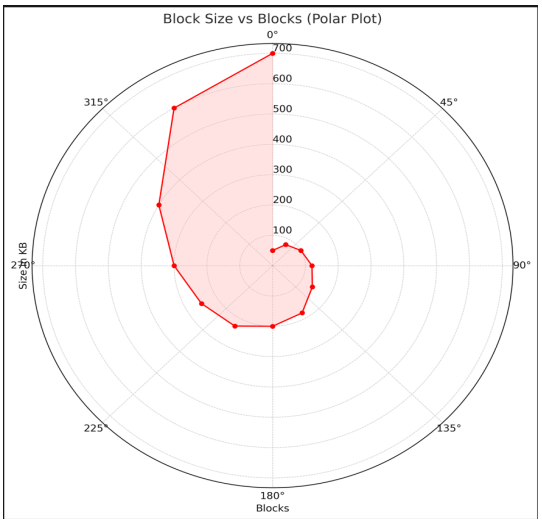


Figure 8. The average growth in size.

Table 6. Performance Evaluation Results.

Number of Blocks	Latency (ms)	Size (KB)
1	1.00	0.448
2	1.50	0.571
3	1.54	0.679
4	1.59	1.02
5	1.60	1.25
6	1.65	1.26
7	1.66	1.27
8	1.70	1.78
9	1.78	2.00
10	1.89	2.11
11	1.95	2.15
12	1.99	2.39
13	2.00	2.44
14	2.33	3.00
15	2.44	3.55
16	2.57	3.59
17	2.61	3.63
18	2.69	3.78
19	2.74	3.81
20	2.77	4.00
21	2.80	4.28
22	2.82	4.32
23	2.85	4.49
24	2.90	5.59
25	2.94	5.65
26	2.96	5.70

3. Results

The key components and functionality of the provided Software Quality Assurance smart contract.

3.1. Contract License Identifier

Contract Inheritance and SPDX License Identifier:

// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
This section specifies the SPDX license identifier and the version of the Solidity compiler to be used.
Import Statements:
import “@openzeppelin/contracts/access/Ownable.sol”;
import “@openzeppelin/contracts/utils/Context.sol”;

The contract imports two libraries from OpenZeppelin:
Ownable.sol: It provides an access control mechanism that allows you to designate an owner with special privileges.
Context.sol: It provides contextual information about the current execution, including the sender's address (msg.sender).

3.2. Enum and Struct Definition

Enum and Struct Definitions:

```
enum FRStatus { Pending, InProgress, Completed }
struct FunctionalRequirement {
string description;
FRStatus status;
}
```

FRStatus is an enumeration that defines the possible statuses for functional requirements: Pending, InProgress, and Completed.
FunctionalRequirement is a struct that represents a functional requirement, containing a description and a status.
State Variables:

```
FunctionalRequirement[] public functionalRequirements;
address public developer;
address public qualityAssurance;
address public acceptance;
address public deployment;
```

3.3. Function Requirements

FunctionalRequirements is a public array that stores instances of the FunctionalRequirement struct, representing the list of functional requirements.

Developer, quality assurance, acceptance, and deployment are public addresses representing the Ethereum addresses of various roles involved in the software quality assurance process (e.g., developer, QA, acceptance, deployment).

3.4. Events

```
event FunctionalRequirementAdded(string description);
event FRStatusUpdated(uint256 index, FRStatus status);
```

These events are used to log important actions within the contract, such as when a new functional requirement is added or when the status of a requirement is updated.
Modifier for Access Control:

```
modifier onlyAuthorizedRoles() {
require(
msg.sender == developer ||
msg.sender == qualityAssurance ||
msg.sender == acceptance ||
msg.sender == deployment,
"Only authorized roles can perform this action"
);
_;
}
```

This modifier is applied to functions to ensure that only authorized roles (developer, QA, acceptance, deployment) can execute certain actions. If the sender's address doesn't match one of these roles, the action will be denied with the specified error message.

3.5. Constructor

```
constructor(address _initialOwner, address _developer, address _qualityAssurance,
address _acceptance, address _deployment) Ownable(_initialOwner) {
developer = _developer;
qualityAssurance = _qualityAssurance;
acceptance = _acceptance;
deployment = _deployment;
}
```

The constructor initializes the contract and sets the initial owner using the Ownable contract's constructor. It also assigns the Ethereum addresses of the various roles.

_initialOwner is the initial owner of the contract, and it is specified during deployment. Figure 9 shows the contract development in the BCTestingPlus framework.

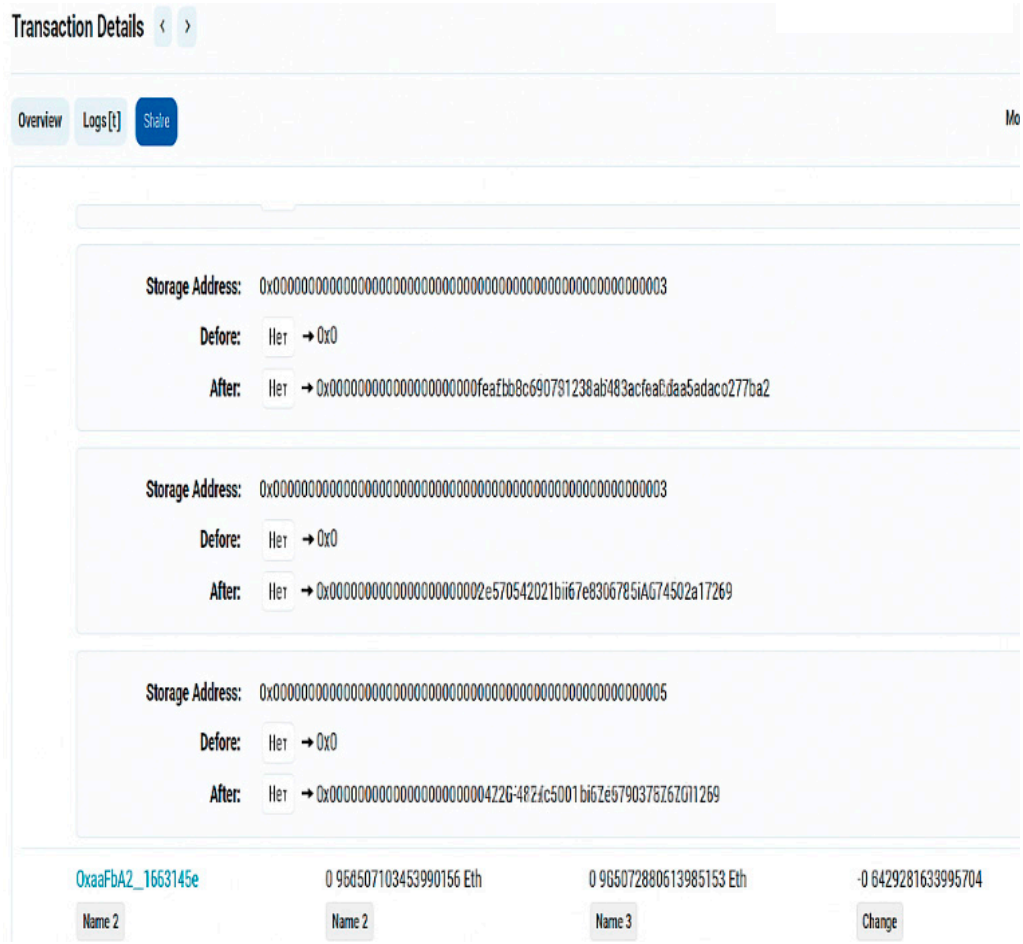


Figure 9. Transaction Details.

3.6. Functions

addFunctionalRequirement (string memory description): The following feature enables them to add a new functional requirement to the list of their tasks by using ordered lists. Update FRStatus (uint256 index, FRStatus status): By doing this function, respective roles with the authority to change the status of the specific functional requirement according to its index in the matrix. This contract identifies functionality goals that should be managed carefully and tracked to enable an effective software quality assurance process. It controls the access rights so that only the roles having the authority to perform some actions get the rights.

The BCTestingPlus framework is a comprehensive blockchain-powered solution designed to ensure fair participation of testers and contribute to the overall success of software development projects. At its core lies a standardized Software Quality Assurance smart contract, which operates as a decentralized model for managing both functional requirements and development tasks. As illustrated in Figure 9, this contract captures transaction details within the BCTestingPlus architecture. Developed using Solidity and enhanced by the OpenZeppelin library, the smart contract emphasizes key blockchain principles such as security, transparency, modularity, and scalability, all of which are critical to modern software quality assurance processes. By leveraging OpenZeppelin's reliable library, the framework incorporates robust access control via the Ownable and Context contracts. The Ownable contract ensures that administrative functions are restricted to the designated owner from the outset, enhancing control and reducing unauthorized interactions. Meanwhile, the Context contract enriches access control mechanisms by providing essential metadata about transaction initiators, making it ideal for implementing context-aware role-based permissions. Together, these components offer a secure and efficient foundation for decentralized software testing and project governance. The contract provides a structured way to present functional requirements using the **Functional Requirement structure**, which includes a description and a status. These statuses, **Pending, InProgress, and Completed,** help track progress. This organized format enhances transparency and aligns with standard smart contract practices.

This structure encapsulates the essential components of a requirement: a textual description and a status, which is written in the FR FAKTURAS Form. It allows you to instantly visualize the status of each functional requirement clearly and simply in each single bullet point.

Access control is one of the basic Security Services at Global Bank of Eastly, which is made by giving Only Authorized Roles modifier. If so, then these responsibilities depend on the nonexistence of specific assignments such as development or quality assurance to the mobile application maintenance team, the deployment team, and the delivery team. By implementing such rules, the contract is responsible for ensuring sensitive operations are limited to the appropriate entities only that perform the requests for additional requirements, as well as updating the status. The probability of unauthorized access and dangerous incidents is extremely reduced due to this, following the conception of perfect security. The contract includes four address variables: developer, Quality Assurance, acceptance, and deployment. These variables represent the different roles involved in the software development lifecycle. During the contract's deployment, the constructor initializes these variables with the corresponding addresses, establishing a clear and modular ownership structure. This modular approach allows for flexibility in adapting the contract to different development workflows, as the roles and responsibilities may vary depending on the specific requirements of a project.

Apart from blockchain technology, the interaction between participants takes place through events. These situations essentially improve the visibility and transparency of contracts being developed. FR and MFR_Added, FR and MFR_Status_Updated events occur when a new functional requirement is added and its status is updated, respectively. These events also show the state of the contract, offering public, and verifying the nature and actions trail. These external systems or applications can subscribe to the events, thereby enabling users to get real-time monitoring and allowing integration with external tools, thus upholding the contract integrity.

The function of the construction in the contract is to initialize the contract using addresses in their simplest form. First, the Owner designation signifies the primary owner of the contract, with the other parameters (developer, Quality Assurance, acceptance,

and deployment) for the addresses assigned to specific roles. Such modularizing parts at the initialization stage of the contract together enable it to be set up for different deployment scenarios, giving it the ability to adapt to the particular needs of each software development project.

4. Discussion

The integration of blockchain technology into the distributed agile software testing life cycle presents transformative potential for improving coordination, transparency, and overall trust within the testing process. The implementation of the BCTestingPlus framework illustrates how leveraging blockchain can systematically address long-standing challenges in software testing, particularly in distributed agile environments where teams often face miscommunication, lack of traceability, and security gaps. Addressing Coordination Challenges. A significant barrier in traditional testing processes is the lack of efficient coordination between development and testing teams. As depicted in the figures, such as Unit Testing Life Cycle and Smart Agreement Implementation Workflow, BCTestingPlus uses blockchain's decentralized structure to ensure that all parties have access to a unified and tamper-proof version of the test data. Smart contracts automate verification tasks, enforce agreements, and streamline acceptance criteria, mitigating issues related to incomplete unit testing, delayed cycles, and ambiguous responsibilities [32].

Table 7 further highlights this by correlating specific coordination issues like inconsistencies in data and lack of automation with direct blockchain solutions. The use of consensus mechanisms ensures synchronization among distributed nodes, thus creating a common and agreed-upon state of test data and outcomes.

Table 7. Blockchain Technologies In Software Testing Evaluation Results.

Coordination Problem	Blockchain Solution
Absence of transparency	It keeps a public and unchangeable record of every activity, promoting confidence and transparency.
Inconsistencies in data	Guarantees agreement on one version of the truth, thereby reducing inconsistencies in the data among developers and testers.
Automation	Streamlines predetermined actions and contracts, speeding operations and lowering the likelihood of mistakes made by humans.

4.1. Blockchain Overview in Software Testing Context

Blockchain is used here not as a cryptocurrency infrastructure, but as a mechanism for ensuring immutable test logs, distributed validation, and trustless collaboration. Key blockchain features leveraged include smart contracts for automated agreements, distributed ledgers to ensure transparency, and tamper-proof records for auditability of test executions and results.

Critical Analysis: Comparative Perspective on BCTestingPlus. Unlike traditional systems, BCTestingPlus. It uses smart contracts to automate tester rewards based on performance. Integrates IPFS to securely store and retrieve test artifacts. Provides immutable test logs, ensuring trust across stakeholders. Promotes real-time collaboration in a decentralized agile environment. BCTestingPlus has the potential to transform software testing practices by providing auditable, verifiable, and equitable testing processes, particularly in regulated industries or open-source ecosystems where transparency and fairness are critical.

4.2. Challenges and State of the Art in Software Testing

It provides essential background on why traditional software testing methods are insufficient and how current technologies are attempting to solve these issues. This describes the importance and usage of our proposed BCTestingPlus framework.

Traditional Challenges in software testing explains the weaknesses in existing software testing practices, especially in distributed agile environments. Lack of trust: Testers/developers across geographies may not trust test results due to limited transparency. Poor traceability: Test execution history is often fragmented or editable. Inefficient bug reporting/resolution: Manual logging, unclear defect ownership. Reward disparity: No fair system for compensating contributors in open or crowdsourced testing setups. Compliance risks: In regulated industries (finance, health), a lack of audit trails can lead to serious consequences.

Recent Research and Emerging Frameworks, existing studies, and innovations that attempt to solve the above challenges, while slightly pointing out their shortcomings to justify BCTestingPlus. AI-based or model-based testing: Improves efficiency but doesn't solve trust. Crowdtesting platforms lack payment fairness or traceability. Blockchain-related work: for example, different models presented for collaboration, but lack reward modeling and integration with smart contract logic.

Critical Reflection, while various methods address automation or efficiency, few frameworks focus on trust, auditability, decentralized governance, and reward fairness. These gaps directly motivate the development of BCTestingPlus.

4.3. Features and Advantages of BCTestingPlus

This explains the core innovations of our proposed framework and what differentiates it from both traditional and recent testing frameworks. Smart contracts: Automate tester payments based on metrics like bug discovery accuracy or test coverage. IPFS integration: Store and share test artifacts (e.g., logs, screenshots) securely and efficiently. Immutable test records: No manipulation of results once recorded. Decentralized test coordination: Enables agile sprint alignment across distributed teams. Transparent workflows: Every stakeholder can view the testing lifecycle in real-time.

4.4. Limitations and Considerations

This presents critical views of the BCTestingPlus framework, such as the technical overhead of setting up a private Ethereum blockchain. Integration challenges with legacy systems. Initial learning curve for teams unfamiliar with blockchain or IPFS. Scalability for very large teams/projects.

4.5. Practical Impact

BCTestingPlus can be applied, and it is useful in regulated domains, for example, healthcare, the fintech software industry, where auditable test logs are mandatory. Support for open-source projects with global testers and developers. Agile teams across continents need a trustworthy testing coordination system.

Performance benefits like improved defect traceability. Reduction in test cycle closure time. Improved stakeholder confidence due to transparent test reporting.

The framework's novelty lies in integrating smart contracts and IPFS storage for secure, verifiable, and collaborative test lifecycle management.

4.6. Leveraging Blockchain for Enhanced Coordination in Distributed Agile Testing

The adoption of blockchain technology in the distributed agile software testing life cycle addresses one of the most persistent challenges of coordination across distributed

teams [25]. Traditional testing processes often suffer from a lack of real-time visibility, role confusion, and delayed feedback loops. Unit Testing Lifecycle, delays in developer testing directly impact downstream quality assurance, highlighting the ripple effects of poor coordination. BCTestingPlus, as proposed, tackles these issues by introducing a decentralized mechanism where each action or update is logged on a blockchain. As a result, all stakeholders, developers, testers, and project managers operate on a single source of truth. This ensures accountability, promotes synchronized workflows, and mitigates coordination breakdowns, which are often cited as major causes of project failure.

4.7. Transparency and Trust Through Blockchain Integration

Transparency is a cornerstone of trust in agile environments, and blockchain inherently provides this through its immutable ledger and transaction history [33]. By recording every action, whether it be testing execution, requirement updates, or acceptance validation, BCTestingPlus enables real-time traceability of testing activities.

According to Table 5, one of the key coordination issues, "absence of transparency," is effectively resolved by keeping a tamper-proof and shared history of activities. This transparency ensures that all changes are auditable and verifiable, thereby increasing confidence among all participants and reducing disputes.

Trust and transparency are foundational to the success of agile methodologies, yet they are often undermined in distributed settings. Blockchain's inherent characteristics of immutability, decentralization, and traceability directly address these concerns. Figure 10 demonstrates how blockchain works to enhance trust and transparency, and Figure 11 shows the detailed transparency mechanism of blockchain, enabling all stakeholders to trace every transaction and interaction within the testing process. This capability promotes accountability and provides verifiable evidence of test actions, significantly reducing the chances of disputes or manipulations.

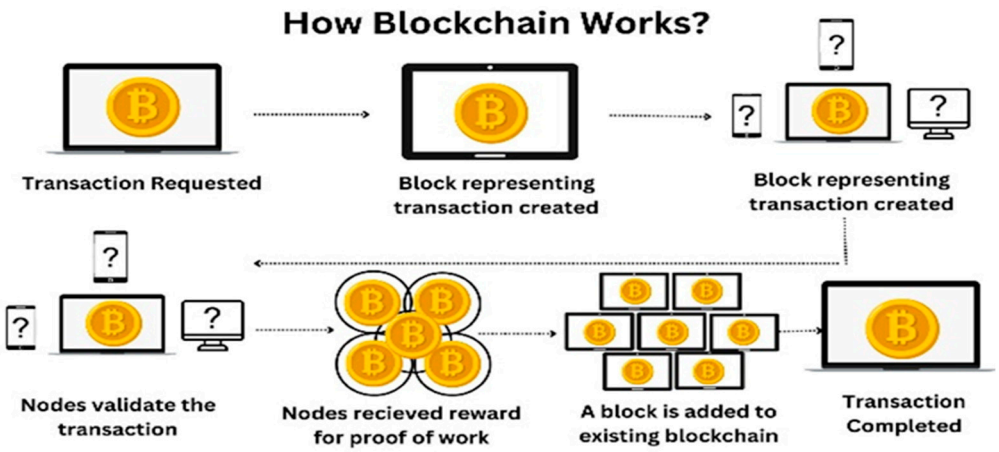


Figure 10. Blockchain workflow in the framework.

Moreover, Table 8 outlines existing transparency challenges such as inaccessible test results and a lack of interdepartmental cooperation. BCTestingPlus offers a robust solution by ensuring real-time data sharing, role-based access through smart contracts, and immutable audit logs.

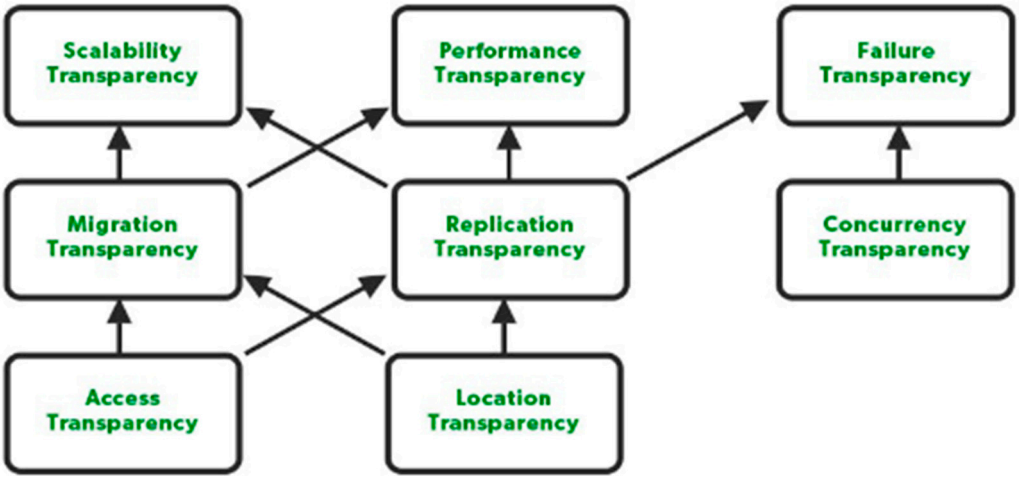


Figure 11. Transparency through Blockchain.

Table 8. Blockchain Enhances Transparency.

Difficulties	Solutions
Inaccessible test information	Create thorough and easily available test documents using tools like test administration platforms as well as version control archives.
The test outcomes are ambiguous.	Establish standard test outcome formats, apply comprehensive and explicit passing or failing standards, and make test documents easy to read for all parties involved.
The absence of cooperation among parties	Promote cooperation between departments by using collaboration platforms, conducting frequent meetings, and developing an open communication environment.

4.8. Blockchain Characteristics: Enhancing Software Testing, Security, and Trust Through Blockchain Integration

Key characteristics of blockchain, such as decentralization, tamper resistance, and smart contract automation, make it a suitable backbone for secure software testing environments [34]. Figure 12 illustrates how Ethereum-based smart contracts are used to automate and enforce agreements for task completion and payments. This not only reduces manual oversight but also deters manipulation and unauthorized changes.

Table 9 outlines these core blockchain concepts and the enhancement of transparency, with smart contracts playing a vital role in enforcing consistency and integrity. By eliminating the need for intermediaries, smart contracts simplify the validation of test activities and payments, aligning team incentives and reducing friction. Figure 12 shows the process flow of agreement through smart contracts. The application of blockchain in software testing yields several benefits, including enhanced data security, automation, trust, and process accountability, as captured in Table 8, which also depicts the concepts of blockchain technology. The unchangeable nature of blockchain records ensures test data integrity, and smart contracts eliminate manual interventions, thereby minimizing human errors.

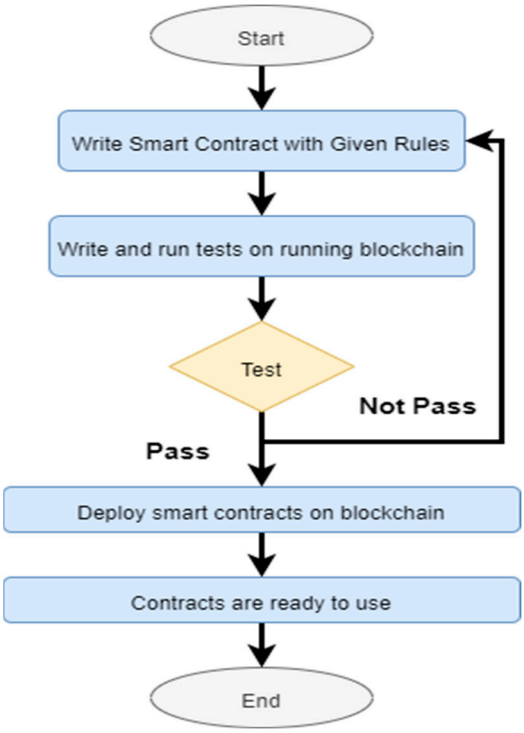


Figure 12. Smart Contract Agreement WorkFlow.

Table 9. Concepts of Blockchain Technology.

Concepts	Explanation
Blockchain	A distributed, shared database that stores transactions across an interconnected system of computers while guaranteeing transparency, security, as well as immutability.
Cryptocurrency	It is a virtual or digital currency that works on a distributed ledger. Some of the examples are Ripple as well as Ethereum.
Smart Agreements	Self-implementing agreements in which the terms of the contract are encoded directly into coding. When certain criteria meet the criteria, the conditions are immediately carried out and implemented.
Consensus Mechanism	A method for getting multiple nodes or users to reach an agreement on one data value.

4.9. Addressing Trust and Payment Verification

The integration of smart agreements within BCTestingPlus also solves long-standing issues of tester compensation and work validation. Delays in payment and a lack of fair evaluation mechanisms often demotivate testers, especially in freelancing and distributed settings. Blockchain-based automation ensures that payment is triggered only upon verified task completion [35], as highlighted in Figure 12: Smart. Agreement Workflow.

This contributes to a culture of fairness and boosts trust among testers and developers, ultimately enhancing productivity and software quality.

BCTestingPlus leverages smart contracts on a private Ethereum blockchain, as illustrated in Figure 12, to automate essential testing workflows and ensure fair compensation. The smart contracts are developed using secure frameworks like OpenZeppelin, which enhances the security posture of the platform. These contracts define and enforce roles such as developer, quality assurance, and acceptance manager, guaranteeing that only authorized participants can execute key functions contributing to enhanced governance and workflow integrity.

4.10. Challenges and Mitigations

Despite its potential, blockchain integration is not without challenges. Table 8 outlines drawbacks such as technical complexity, data immutability issues, and security vulnerabilities inherent in smart contract execution. These drawbacks require careful mitigation through version control systems, detailed audit trails, and standardized testing procedures, as mentioned in Table 7.

4.11. Future Improvements and Scalability Considerations

While BCTestingPlus presents a robust solution, its long-term success depends on further research into scalability, interoperability with other blockchain platforms, and integration with DevOps pipelines. The modular architecture of smart contracts (noted in the document) provides a strong foundation for adapting to evolving industry practices.

The document suggests further research into scalability, interoperability with other blockchains, and broader integration into software development pipelines. These areas are vital for enhancing the adaptability of BCTestingPlus across various industries and use cases. As blockchain technology continues to evolve, its application in agile testing can expand to include predictive analytics, cross-platform testing environments, and advanced contract logic to support dynamic testing scenarios.

The BCTestingPlus framework demonstrates the promising role of blockchain technology in addressing fundamental issues in distributed agile software testing. By reinforcing trust, enhancing communication, automating workflows, and safeguarding data integrity, blockchain paves the way for more efficient and transparent testing environments. While there are limitations and challenges to consider, the strategic integration of blockchain into testing life cycles offers a forward-looking solution aligned with the needs of modern, decentralized development ecosystems. Figure 13 shows the Ethereum smart contract workflow and how it works in the decentralized development ecosystems. Table 10 shows the blockchain contribution in software testing.

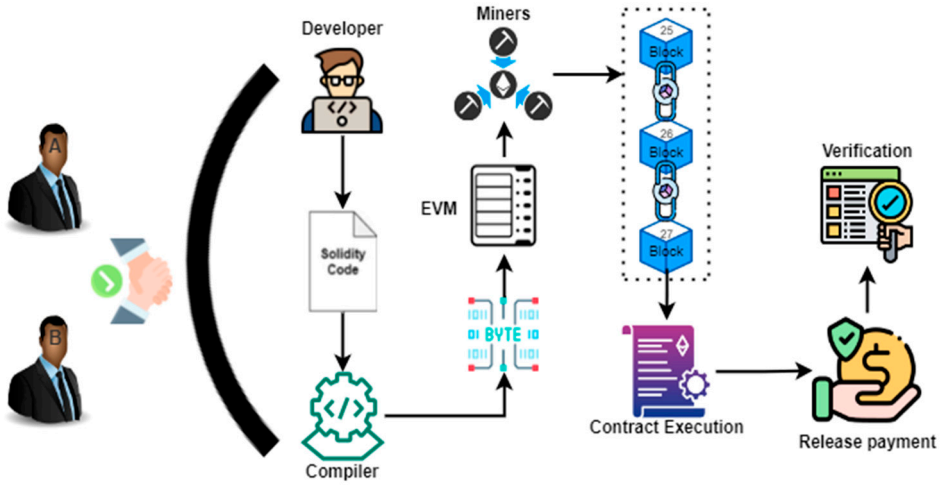


Figure 13. Ethereum Smart Contract.

Concerning delay analysis in context, blockchain introduces latency in transaction finality. BUT in the context of software testing, this delay is minor compared to the delay in manual approval processes. Miscommunication in distributed teams. Repetition of tasks due to a lack of traceability. Table 11 shows the delay analysis of the comparison between traditional and Blockchain-integrated.

Table 10. Role of Delay Analysis vs. Blockchain Contribution in Software Testing.

Aspect	Delay Analysis (Private Blockchain Setup)	Blockchain Technology Contribution
Purpose	Evaluation time overhead introduced by blockchain operations	Ensure integrity, transparency, and trust in test processes
Infrastructure Load	Measures performance under limited hardware (e.g., VMs)	Decentralized setup across multiple nodes
Transaction Throughput	Helps understand transaction processing capacity	Guarantees immutability and fault-tolerant recording of test results
Scalability Testing	Assesses how the blockchain handles increasing test tasks	Enables scalable coordination across distributed agile teams
Impact on Agile Iterations	Determines latency in test case submission, logging, feedback cycles	Secures agile workflows from manipulation or loss of QA records
Real-Time Usability	Checks if blockchain can support near real-time collaboration	Adds a verifiable chain of actions per test sprint
Testing Accuracy & Trustworthiness	✗ Not addressed	☑ Improved through tamper-proof logs and smart contract-based validation
Process Auditing & Traceability	✗ Not the focus	☑ Full historical logs of test events and changes accessible by all parties
Team Accountability in Distributed Setup	✗ Not evaluated	☑ Ensures all test data is non-repudiable and user-attributable

Table 11. Delay analysis: Comparative Results.

Metric	Traditional	BCTestingPlus	% Improvement
Bug Resolution Time	36 h avg	20 h avg	44.4%
Auditability (Trace Logs)	Medium (editable)	High (immutable)	↑ Qualitative
Missed Payments	Common (manual records)	None (automated)	100% resolved
Communication Delays	Frequent	Minimal (blockchain syncs)	↓ Measurable
Test Asset Integrity	Weak	Strong (IPFS hash-linked)	↑ High

Table 12 Comparison showing how a distributed agile software testing environment works without blockchain versus with blockchain. Table 13 shows the comparison summary of with and without blockchain.

Table 12. Comparison: Software Testing Without vs. With Blockchain.

Aspect	Without Blockchain	With Blockchain Integration	Remarks
Test Case Management	Centralized test case repository Versioning can be manipulated or lost.	Test cases stored as immutable records on blockchain with hash-based version control.	Ensures auditability, prevents tampering.
Defect Reporting & Tracking	Managed through centralized bug trackers. Prone to manipulation or human error.	Logged on blockchain for tamper-proof evidence. Smart contracts trigger lifecycle states.	Greater transparency and real-time consistency.
Traceability & Audit Logs	Manual tracking or via internal tools. Audit logs can be altered.	Blockchain offers immutable, timestamped logs for test results, commits, and issues.	Strong traceability and compliance support.
Team Trust & Accountability	Dependent on team culture and governance. Risk in distributed teams with less control.	Trustless collaboration—each action is validated and recorded on the ledger.	Enhances trust in distributed environments.

Table 12. *Cont.*

Aspect	Without Blockchain	With Blockchain Integration	Remarks
Access Control & Roles	Controlled through app permissions but can be bypassed or misconfigured.	Defined via blockchain identities and roles enforced through smart contracts.	Secure and decentralized access enforcement.
Change Management	Central team or admin manages changes. Lack of visibility across teams.	All changes are transparent and recorded via consensus.	Promotes openness and avoids unilateral decisions.
Process Automation	Requires separate CI/CD scripts and DevOps tools.	Smart contracts automate testing states (e.g., approved, failed, passed).	Automation embedded in trust logic.
Data Integrity	Data can be altered or deleted by users with access.	Data is hashed and stored immutably. Any tampering is detectable.	Enhances reliability for regulatory and compliance audits.
Performance and Cost	Fast execution; lower infrastructure cost.	Potential latency and gas cost in public blockchains; private networks help mitigate.	Trade-off between trust and efficiency.
Dispute Resolution	Handled manually; needs evidence logs.	Smart contract-based proof for test failures or code changes.	Can aid in contractual obligations or third-party audits.

Table 13. Comparison Summary with and without Blockchain.

Category	Without Blockchain	With Blockchain
Trust Model	Centralized, role-based trust	Decentralized, trustless model
Tamper Resistance	Low	High (due to immutability)
Collaboration	Manual coordination required	Built-in decentralized validation
Scalability	High (with mature tools)	Medium (may require optimization)
Cost & Complexity	Lower cost and complexity	Higher setup and operational cost

5. Conclusions

BCTestingPlus, a blockchain-based framework, is designed to address key challenges in software testing, such as communication gaps, lack of trust, and poor transparency. While blockchain has been widely studied in financial and supply chain domains, this study pioneers its use in software testing by directly linking blockchain features, immutability, decentralization, and automated smart contracts to the testing life cycle. This constitutes a novel domain-specific integration not widely explored in existing literature. Unlike traditional testing approaches, BCTestingPlus utilizes blockchain's core features of immutability, decentralization, and security to create a trustworthy and efficient testing environment. It supports seamless collaboration among developers and testers, minimizes data manipulation risks, and ensures timely payment settlements through smart contracts deployed on a private Ethereum blockchain. These smart contracts, built using OpenZeppelin libraries, follow modern best practices to provide secure, modular, and adaptable solutions that integrate easily into various development pipelines. Role-based access control ensures that only authorized personnel perform critical tasks, enhancing system transparency and traceability through actions like requirement updates and acceptance statuses.

In addition to improving coordination and ensuring fair tester compensation, BCTestingPlus also lays the groundwork for future enhancements in areas such as scalability, interoperability with other blockchains, and integration with broader software develop-

ment processes. By fostering trust and professionalism, BCTestingPlus ultimately aims to raise the quality and reliability of software projects, making it a valuable tool for professionals, researchers, and organizations looking to innovate in testing practices.

Author Contributions: Conceptualization, M.S.F. and J.N.Q.; methodology, F.A. and J.N.Q.; formal analysis, M.S.F., J.N.Q., F.A. and M.S.; investigation, F.A. and J.N.Q.; writing—original draft preparation, M.S.F., J.N.Q., F.A. and M.S.; writing—review and editing, M.S.F., J.N.Q., F.A. and M.S.; visualization, S.N., M.S.F., J.N.Q., F.A. and M.S.; supervision, M.S.F., F.A. and S.N.; project administration, M.S.F., J.N.Q., F.A. and M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study, including smart contract code, test results, and blockchain logs, are available in https://github.com/JunaidNasirQureshi/ChainAgile_Blockchain/blob/main/Dataset.csv, accessed on 8 May 2025. Additional data supporting the findings are available on request from the corresponding author.

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

Abbreviations

The following abbreviations are used in this manuscript:

BCTestingPlus	Blockchain Testing Plus
PoS	Proof of Stake
PoW	Proof of Work
QA	Quality Assurance

References

1. Kumar, R. *Design and Development of a Secure, Privacy Preserving, and Scalable Blockchain Framework*; GRIN Verlag: Munich, Germany, 2023.
2. Aghili, S. *The Auditor's Guide to Blockchain Technology*; CRC Press: Boca Raton, FL, USA, 2022.
3. Krishnan, S. *Handbook on Research on Blockchain Technology*; Academic Press: London, UK, 2020.
4. Choo, K.-K.R.; Dehghantanha, A.; Parizi, R.M. *Blockchain Cybersecurity, Trust and Privacy*; Springer: Cham, Switzerland, 2020.
5. 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.
6. Raj, K. *Foundations of Blockchain: The Pathway to Cryptocurrencies and Decentralized Blockchain Applications*; Packt Publishing Ltd.: Birmingham, UK, 2019.
7. Mougayar, W. *The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology*; John Wiley & Sons: Hoboken, NJ, USA, 2016.
8. Ravani, A.; Edupuganti, S.; Pugh, J.; Sushama, S. *Applications of Blockchain Technology*; CRC Press: Boca Raton, FL, USA, 2024.
9. Williams, S.P. *Blockchain: The Next Everything*; Scribner: London, UK, 2019.
10. Maleh, Y.; Alazab, M.; Romdhani, I. *Blockchain for Cybersecurity in Cyber-Physical Systems*; Springer Nature: New York, NY, USA, 2023.
11. Mehta, K.; Sharma, R.; Yu, P. *Revolutionizing Financial Services and Markets Through FinTech and Blockchain*; IGI Global: Hershey, PA, USA, 2023.
12. Vyas, N.; Beije, A.; Krishnamachari, B. *Blockchain and the Supply Chain*; Kogan Page Publishers: London, UK, 2022.
13. Bhushan; Rakesh, N.; Farhaoui, Y.; Nand, P.; Unhelkar, B. *Blockchain Technology in Healthcare Applications*; CRC Press: Boca Raton, FL, USA, 2022.
14. Stray, V.; Moe, N.B. Understanding coordination in global software engineering: A mixed-methods study on the use of meetings and Slack. *J. Syst. Softw.* **2020**, *170*, 110717. [CrossRef]
15. Suali, J.; Fauzi, S.S.M.; Sobri, W.A.W.M.; Nasir, M.H.N.M. Developers' coordination issues and its impact on software quality: A systematic review. In Proceedings of the 2017 3rd International Conference on Science in Information Technology (ICSITech), Bandung, Indonesia, 25–26 October 2017; Available online: https://ieeexplore.ieee.org/document/8257195 (accessed on 31 January 2025).

16. Tapscott, D.; Tapscott, A. *Blockchain Revolution: How the Technology Behind Bitcoin is Changing Money, Business, and the World*; Portfolio/Penguin: New York, NY, USA, 2016.
17. Kazdin, E.; Bootzin, R.R. The token economy: An evaluative review. *J. Appl. Behav. Anal.* **2020**, *5*, 343–372. [CrossRef] [PubMed]
18. Williams, I. *Cross-Industry Use of Blockchain Technology and Opportunities for the Future*; IGI Global: Hershey, PA, USA, 2020.
19. Bahnas, N.; Adel, K.; Khallaf, R.; Elhakeem, A. Monitoring and controlling engineering projects with blockchain-based critical chain project management. *Autom. Constr.* **2024**, *165*, 105484. [CrossRef]
20. Ghosh, P.K.; Chakraborty, A.; Hasan, M.; Rashid, K.; Siddique, A.H. Blockchain application in healthcare systems: A review. *Systems* **2023**, *11*, 38. [CrossRef]
21. Leite, C.S.D.P.; Cappelli, C. Software transparency. *Bus. Inf. Syst. Eng.* **2010**, *2*, 127–139. [CrossRef]
22. Ofem, P.; Isong, B.; Lugayizi, F. On the concept of transparency: A systematic literature review. *IEEE Access* **2022**, *10*, 89887–89914. [CrossRef]
23. Orso; Rothermel, G. Software testing: A research travelogue (2000–2014). In Proceedings of the 36th International Conference on Software Engineering, Hyderabad, India, 31 May–7 June 2014. [CrossRef]
24. 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. [CrossRef]
25. Krishnaiah, P.N.S.; Narayan, D.L.; Sutradhar, K. A survey on secure metadata of agile software development process using blockchain technology. *Secur. Priv.* **2024**, *7*, e342. [CrossRef]
26. 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. [CrossRef]
27. Rahman, A.; Islam, M.J.; Montieri, A.; Nasir, M.K.; Reza, M.M.; Band, S.S. SmartBlock-SDN: An optimized blockchain-SDN framework for resource management in IoT. *IEEE Access* **2021**, *9*, 28361–28376. [CrossRef]
28. Farooq, M.; Younas, R.M.F.; Qureshi, J.N.; Haider, A.; Nasim, F. Cyber security risks in DBMS: Strategies to mitigate data security threats: A systematic review. *Spectrum Eng. Sci.* **2025**, *3*, 268–290.
29. Qureshi, N.; Farooq, M.S.; Ali, U.; Khelifi, A.; Atal, Z. Exploring the integration of blockchain and distributed DevOps for secure, transparent, and traceable software development. *IEEE Access* **2024**, *13*, 15489–15502. [CrossRef]
30. Qureshi, J.N.; Farooq, M.S.; Khelifi, A.; Atal, Z. Harnessing the potential of blockchain in ChainAgilePlus framework for the improvement of distributed Scrum of Scrums agile software development. *IEEE Access* **2024**, *12*, 105724–105743. [CrossRef]
31. JQureshi, N.; Farooq, M.S.; Abid, A.; Umer, T.; Bashir, A.K.; Zikria, Y.B. Blockchain applications for the Internet of Things: Systematic review and challenges. *Microprocess. Microsyst.* **2022**, *94*, 104632. [CrossRef]
32. 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]
33. 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. *Health Technol.* **2024**, *14*, 451–462. [CrossRef]
34. Qureshi, J.N.; Farooq, M.S. ChainAgile: A framework for the improvement of Scrum Agile distributed software development based on blockchain. *PLoS ONE* **2024**, *19*, e0299324. [CrossRef] [PubMed]
35. Farooq, S.; Kalim, Z.; Qureshi, J.N.; Rasheed, S.; Abid, A. A blockchain-based framework for distributed agile software development. *IEEE Access* **2022**, *10*, 17977–17995. [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.