

Received October 5, 2021, accepted January 7, 2022, date of publication January 27, 2022, date of current version February 17, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3146953

# A Blockchain-Based Framework for Distributed Agile Software Development

MUHAMMAD SHOAIB FAROOQ<sup>1</sup>, ZAREEN KALIM<sup>1</sup>, JUNAID NASIR QURESHI<sup>2</sup>,  
SAIM RASHEED<sup>3</sup>, AND ADNAN ABID<sup>1</sup>, (Member, IEEE)

<sup>1</sup>Department of Computer Science, University of Management and Technology, Lahore 54000, Pakistan

<sup>2</sup>Department of Computer Science, Bahria University Lahore Campus, Lahore 54000, Pakistan

<sup>3</sup>Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 80200, Saudi Arabia

Corresponding author: Muhammad Shoaib Farooq (shoaib.farooq@umt.edu.pk)

**ABSTRACT** Distributed Agile Software Development (DASD) is the most important approach for the modern software industry that allows geographically independent software development. In the past, different tools and frameworks were proposed to solve communication and collaboration issues in DASD but they lacked transparency, trust, traceability, and security. These shortcomings resulted in project failure or overdue, customer dissatisfaction, project deal cancellations, and payment clashes between the customers and development teams. This paper addresses and overcomes the major issues of transparency, trust, security, traceability, coordination, and communication in DASD by embedding blockchain technology. We have proposed a novel blockchain-based framework named as AgilePlus which executes the smart contracts on a private ethereum blockchain for acceptance testing, secure payment, verification of developer's payment requirements, and automatic payment distribution into the digital wallets of development teams. The execution of these smart contracts automatically assign penalties to the customers for late or non-payments and also to the developers for overdue tasks. Moreover, we have also solved the blockchain's scalability challenge in AgilePlus by utilizing Interplanetary File System (IPFS) as off-chain storage. Lastly, experimental results prove that the proposed framework enhances transparency, communication, coordination, traceability, security and solves trust issues of both customers and developers in DASD.

**INDEX TERMS** Distributed agile software development, blockchain technology, ethereum blockchain, smart contracts, interplanetary file system.

## I. INTRODUCTION

Over the last few decades, Distributed agile software development (DASD) has been used as a common practice with the increase in global software development requirements and most of the software companies have adopted this approach [1]. Software development teams working collaboratively at different geographical locations is known as Distributed software development [2]. Agile is a set of principles and values for software development based on incremental and iterative development, where requirements and solutions are evolved by the collaboration between self-organizing and cross-functional teams [3], [4]. Agile software development promotes early delivery, adaptive planning, testing for continuous improvement, and quick response to changes proposed by the customer [5]. The agile life cycle follows six phases for

successful software development as shown in Fig. 1. However, DASD is the combination of both agile software development and distributed software development or global software engineering [6]. The DASD is based on geographically dispersed agile development teams which offer various advantages such as increased quality, reduced development time, and cost [7].

Many challenges have been faced by development teams when the software development process is based on both agile and distributed software development [8]. The major challenge in DASD is to achieve frequent communication and collaboration [9]. Most of the risks and challenges in DASD have been associated with geographical distance which involves, trust, traceability, transparency, security, and communication [11], [12], [23], [7], [10]. Recent studies [21], [23] have suggested that current tools for communication can support DASD practices by overcoming the issues of communication and collaboration but these researches lacked

The associate editor coordinating the review of this manuscript and approving it for publication was Hailong Sun<sup>1</sup>.

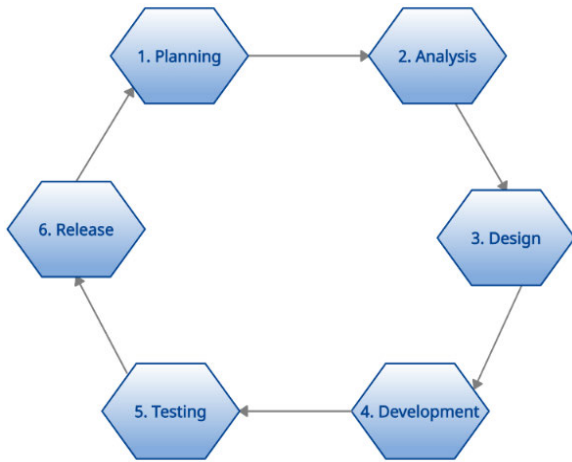


FIGURE 1. Agile software development life cycle.

trust, transparency, traceability, and security. Therefore, all these deficiencies in state-of-the-art studies resulted in overdue or project failure, project deal cancellations, customer dissatisfaction, and payment clashes between customers and developers.

Customers and developers face major challenges in distributed environments that need to be addressed for meeting the intended objectives of a project. Customers at geographically distributed locations are concerned about the software development process according to their requirements and satisfaction. However, developers do not trust customers for timely payments, project cancellation cost, and their coordination with the development team. Thus, all these issues can be solved by combining blockchain technology with DASD as blockchain plays a vital role in creating trust between geographically dispersed customers and developers around the globe.

Blockchain technology can have significant impacts on DASD because of its decentralized nature, immutability, distributed database, and distributed ledger [13]. Blockchain is a sequence of blocks that holds the complete record of transactions [14] verified and validated by peer-to-peer nodes in the network without using any central authority [15]. Recent innovations in blockchain technology have enabled the creation of smart contracts that automatically stores and executes code on the blockchain [16]. It also deals with the transferring of ether cryptocurrency in digital wallets without involving any trusted third party [17]. Most of the emerging systems have been shifted on a peer-to-peer, decentralized network, and using transparent distributed ledger systems with a diverse data structure [19]. Therefore, an efficient blockchain embedded framework is a major requirement to overcome all the challenges in DASD.

We have proposed a blockchain-based framework named as AgilePlus in order to address all the issues faced by customers and developers. This framework embeds a private ethereum blockchain which has been considered promising for improving security as it promotes secure payment transac-

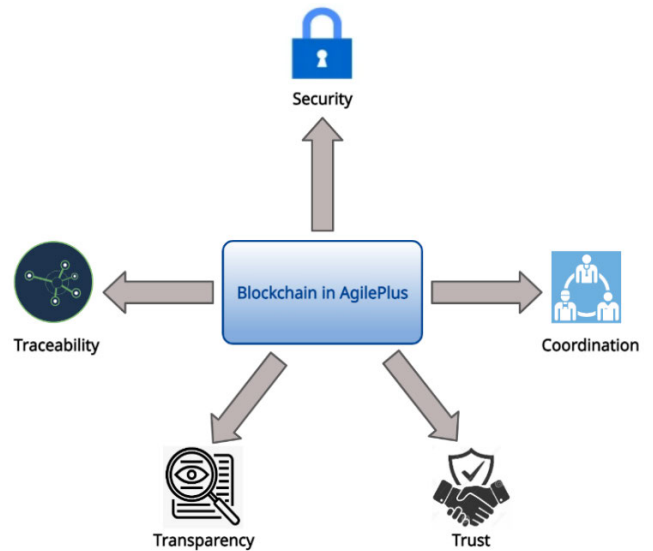


FIGURE 2. Blockchain features in AgilePlus.

tions and protects the system against 51% attacks. AgilePlus connect its users by providing a user-friendly interface and six virtual walls through Decentralized Applications (DApps). AgilePlus consists of six layers named as 1. Agreement, 2. Requirements Elicitation, 3. Prioritization, 4. Design and Development 5. Testing and 6. Payment layer. Each layer has its separate virtual wall which is activated when the process of previous layer is finished by the customer and development team. This framework also executes the smart contracts for acceptance testing, secure payment, verification of developer’s payment requirements, payment distribution, and assigns penalty to its users. All these transactional steps are stored in a private ethereum blockchain. The features of blockchain technology as shown in Fig. 2, ensure trust by storing data in a way that reduces the risk of hacking or tampering the system [18].

We have also solved the scalability challenge in AgilePlus by storing the huge volume of records about customers, developers, and their communication on off-chain storage IPFS. It reduces the data load on blockchain and enhances the performance of blockchain transactions. However, all the data within six layers of AgilePlus is stored in blockchain to track the work progress of agile teams.

The major objective of this study is to address the key challenges that are associated with the geographical distance in DASD by using blockchain technology. Therefore, AgilePlus framework deploy and execute smart contracts to overcome the issues of trust, transparency, traceability, security, communication, and coordination.

The key contributions of this research are as follows:

We have proposed the AgilePlus framework for DASD which integrates private ethereum blockchain to prevent 51% attacks. The proposed framework uses DApp to provide its users with a user-friendly interface for communication and collaboration. AgilePlus deploy and execute smart contracts

for acceptance testing, secure payment, verification of developer's payment requirements, automatic payment distribution between development teams, and assign penalties to its users if they have not satisfied the terms and conditions on time. AgilePlus allows the users to send and receive ether cryptocurrency in digital wallets for payments which can be converted into fiat currency by using a currency conversion service/ exchange. AgilePlus stores all the records of six layers in blockchain for tracking the work progress of distributed agile teams. Moreover, we have utilized Interplanetary File System (IPFS) as off-chain secondary storage to solve the scalability challenge of blockchain and also implemented JSON objects. Lastly, we have evaluated the performance of AgilePlus based on the increase in chain size over the addition of block's new transactions and latency for retrieving the longest chain.

The novelty of this research is that we have embedded blockchain technology in a user-friendly framework as the social website for solving trust issues of both customers and developers. To the best of our knowledge, a blockchain embedded framework for DASD has not been proposed yet in previous studies. We have executed the smart contracts for acceptance testing, secure payment, verification of developer's payment requirements, and automatic payment distribution between development teams by transferring ethers into their digital wallets. These ethers can be converted into fiat money by using an exchange/ currency conversion service. The execution of smart contracts also assign penalties to both customers and developers, if they have not satisfied the terms and conditions within due date. Moreover, we have solved the blockchain's scalability issue by utilizing IPFS and also implemented JSON objects.

The remaining paper is organized as follows: Section II presents the related work and Section III describes the preliminaries used in our framework. Our proposed framework AgilePlus has been presented in Section IV and Section V presents the performance evaluation of AgilePlus. Then, Section VI presents the discussion of this research. Finally, Section VII draws the conclusion and presents the future directions of this research work.

## II. RELATED WORK

Many researchers have proposed different tools and frameworks for DASD to overcome the challenges faced by customers and development teams. Most of them solved the communication, collaboration, and coordination issues but little has been explored to solve the trust, traceability, transparency, and security issues by using blockchain technology.

Kukreja [20], [21] proposed Winbook framework similar to the social website that integrates user-friendly features such as virtual wall, posts, comments like Facebook, and content organization using color-coded labels, similar to Gmail. These Winbook features overcome the communication and collaboration issues. However, Winbook does not have the capability of prioritization and it lacks blockchain technol-

ogy to address security, traceability, transparency, and trust problems.

Lampropoulos and Siakas [22] investigated the communication challenges in DASD and the usage of social media as a collaboration and communication tool. They also explored that effective communication, cultural difference, time zone difference, coordination, and trust are the main challenges in DASD. However, the use of social media only overcomes coordination and communication issues.

Furthermore, this research does not overcome the major problems in DASD such as traceability, security, transparency, and trust.

Younas *et al.* [23] proposed and evaluated the Agile Development Cloud Computing framework (ADCC) that unifies agile development and cloud computing for local and distributed agile development environments. Although evaluation results proved that cloud computing in ADCC framework overcomes the problems of face-to-face communication, transparency, scalability, and resource management, yet it has several challenges of data security risks, privacy, interoperability, and high cost. Moreover, ADCC lacks blockchain technology to overcome traceability, security, and trust issues.

Stray *et al.* [24] discussed the usage of slack tool for coordination and communication in DASD. Slack acts as a forum for customers and development teams to share files, arrange meetings, and improve coordination between customers and developers. This tool is useful in solving collaboration and communication issues but it does not overcome the traceability, trust, transparency, and security challenges.

Qureshi *et al.* [25] proposed a novel Communication and Coordination (C&C) framework using scrum methodology. It involves distributed agile teams located at multiple locations having a scrum master who acts as technical support person and communication coordinator. Although proposed framework achieved goals of decision making, consensus on sprint backlog, problem-solving, communication, and coordination, yet it lacks blockchain technology to solve the traceability, security, trust, and transparency issues.

Although many researches have been conducted to overcome the communication and collaboration challenges in DASD, yet several studies have discussed the usage of blockchain technology to ensure security, transparency, traceability, and trust between users.

Demi [26] proposed a blockchain-enabled framework for requirements traceability and management throughout the software development life cycle. The proposed framework mitigates the need to involve tracing tools or third-party requirements management. It also provides a distributed and collaborative environment for increasing the trustworthiness, transparency, and accuracy of the requirements traceability process. However, it is not suggested for DASD as it only solves the issues of requirements engineering process.

Rene *et al.* [27] proposed ChainSoft, an efficient platform for outsourcing the development of software and automatic payments by using blockchain technology. This system has been implemented using Github/Travis CI and ethereum

smart contracts for solving the challenges of collaboration, security, trust, and testing in a distributed software development environment. However, this platform does not solve the challenges of transparency and traceability.

Marchesi *et al.* [28] presented the applications of blockchain technology in agile development. They suggested the approach for Scrum or Lean- Kanban processes to deploy smart contracts on ethereum blockchain for acceptance testing and payment. This approach solves the trust, coordination, and transparency issues for Scrum or Lean-Kanban processes but it is not suggested for the distributed environments. Moreover, the issues of security and traceability have not been addressed by using blockchain technology.

All the above discussions show that still there is a need for an efficient framework to overcome all the major issues of traceability, security, trust, and transparency in DASD by using blockchain technology. However, there is not an adequate framework proposed yet to address all these problems which are faced by customers and agile teams having geographical differences. In this research, we have proposed a novel AgilePlus framework to fulfill the gaps in previous frameworks and researches.

The novelty of our work is that we have used blockchain technology in a user-friendly framework to improve DASD. We have solved the customer's trust issues by executing smart contracts for acceptance testing. We have also solved the development team's trust issues for secure payment, verification of their payment requirements, and payment distribution between them by executing smart contracts. The execution of smart contracts automatically assign penalties to the customers and developers for not satisfying the requirements within due date. AgilePlus also provide development teams with the option of using an exchange/ currency conversion service for converting ethers into fiat currency. Furthermore, we have used IPFS as off-chain storage to solve the blockchain's scalability issue and also implemented JSON objects.

### III. PRELIMINARIES

This section presents the preliminaries used in our proposed framework. It describes the major components of blockchain that have been used in AgilePlus which include Ethereum Blockchain, Smart Contracts, Decentralized Applications (DApps), and Interplanetary File System (IPFS).

#### A. ETHEREUM BLOCKCHAIN

Ethereum [29] is an open-source, and decentralized blockchain that consists of the smart contract functionality. It is also known as a public ledger for recording and verifying transactions. Ethereum software platform has its digital cryptocurrency known as ether (ETH) which can be shared between accounts that are connected with the ethereum blockchain network. Users interact with their ethereum accounts through ethereum wallets or applications. This distributed blockchain network is also used to run DApps, smart

contracts, and store data. It uses a peer-to-peer (P2P) network that makes it distributed.

#### B. SMART CONTRACTS

Smart contracts [30] are the lines of code or computer programs that are stored in the ethereum blockchain and execute automatically when certain terms and conditions are satisfied. These smart contracts or transaction protocols for ethereum are written in solidity language and executed on the blockchain in order to document events according to the conditions of an agreement or contract. The objective of using smart contracts is to reduce the involvement of trusted third parties and malicious exceptions.

#### C. DECENTRALIZED APPLICATIONS (DApps)

DApp [31] is one of the main features of ethereum which is also known as a digital program or an application that runs on a peer-to-peer network. DApps are comprised of smart contracts which have the user-defined code in order to perform the defined tasks. Ethereum Virtual Machine (EVM) is used to deploy and execute this code. Therefore, DApps consist of a frontend or interface to communicate with users and a smart contract or backend code that runs on a decentralized P2P network.

#### D. INTERPLANETARY FILE SYSTEM (IPFS)

IPFS [32], [33] refers to the protocol that uses a P2P network for sharing and storing data in a content-addressable distributed file system. The data stored on IPFS is secure from modification as it uses the cryptographic identifier. Therefore, the cryptographic identifier is changed if data needs to be modified. A unique hash value is generated cryptographically for the identification of data files stored on IPFS which does not allow the duplication of files. This cryptographic hash is stored in the blockchain to solve its scalability issue by reducing the data load on it.

### IV. THE PROPOSED FRAMEWORK: AGILEPLUS

This section presents a transparent, scalable, and secure framework to manage DASD for geographically scattered customers and development teams by integrating blockchain. We have proposed AgilePlus framework which follows the agile software development life cycle and embeds private ethereum blockchain to manage the transactions of AgilePlus efficiently for posts, customer's quick feedback, acceptance testing, verification of developer's payment requirements as per business logic, and transferring of payments securely in the digitals wallets of development teams. The blockchain technology embedded in AgilePlus framework automatically assigns penalty to the customers for late or non-payments and also to the developers for overdue tasks. All the characteristics of blockchain make it suitable technology for agile software development in a distributed environment with distributed ledger, distributed database, and strong security properties. The ecosystem of AgilePlus is presented in Fig. 3 which covers all the major features of blockchain.

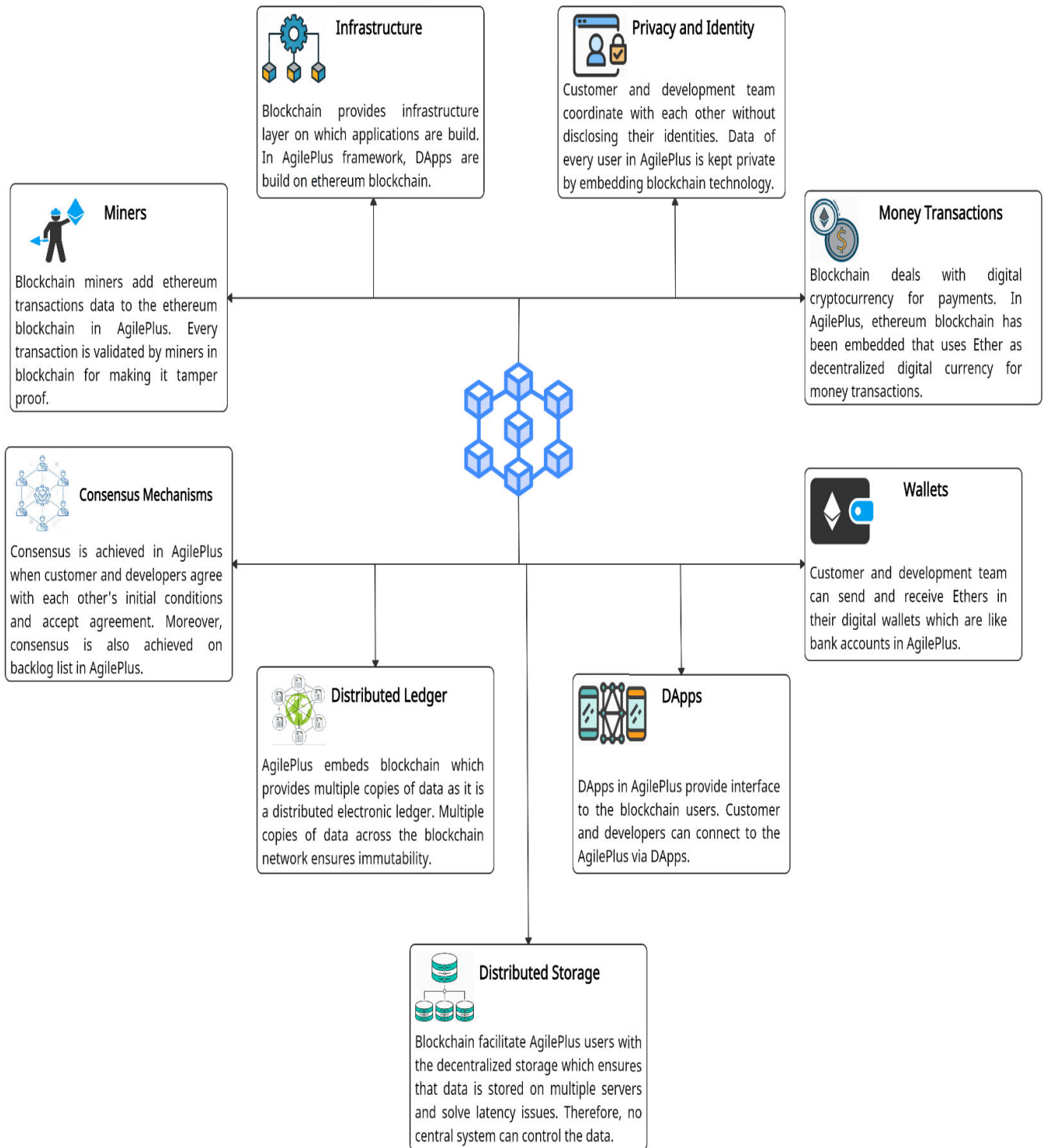


FIGURE 3. AgilePlus blockchain ecosystem.

The main components of AgilePlus blockchain ecosystem are infrastructure, DApps, identity and privacy, money transactions, exchanges, wallets, miners, consensus mechanisms, distributed ledger, and distributed storage. AgilePlus framework deals with secure payment transactions, payments, and

privacy of user data. The blockchain provides infrastructure for building DApps in order to facilitate AgilePlus users with a user-friendly interface. Customers and developers can connect with AgilePlus framework via DApp and access the blockchain network. All the stakeholders can use six

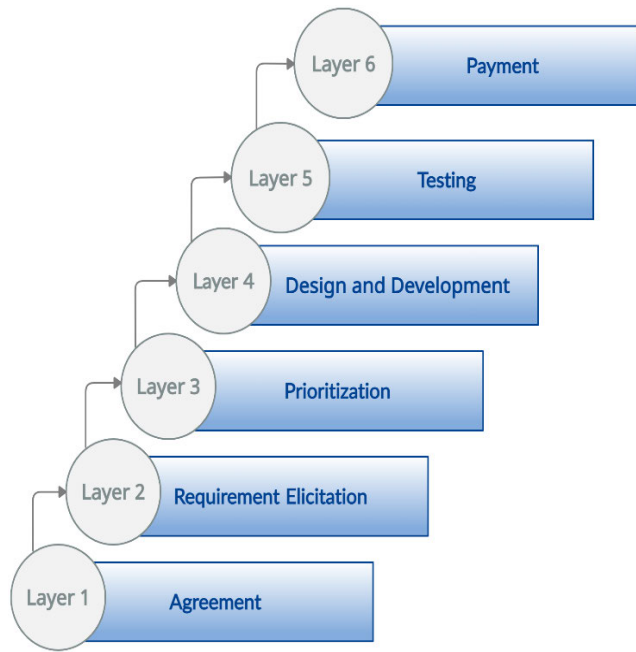


FIGURE 4. Layers in AgilePlus framework.

virtual walls for effective communication and coordination. The components of AgilePlus blockchain ecosystem provide more control over information to the geographically dispersed users and promote data transparency in the framework. AgilePlus consists of six layers as shown in Fig.4. It involves the participation of customers in all these layers to track the work progress of development teams by using blockchain traceability.

The AgilePlus framework has been shown in Fig. 5 which execute smart contracts in the testing layer for acceptance testing and in the payment layer for verification of developer's payment requirements and payment distribution in the form of ethers between development teams across the globe. AgilePlus also verifies all the terms and conditions agreed by customers and developers by executing smart contracts. It assigns automatic penalty to the developers if user story has not been completed within due date. Moreover, AgilePlus assigns penalty to the customers in case of late or non-payments. All these steps are recorded and stored in the private ethereum blockchain.

#### A. AGILEPLUS REGISTRATION PROCESS

All the stakeholders register AgilePlus framework by providing their credentials. The project managers and developers register themselves as a candidate by providing their Name, Email, Contact Number, Qualification, Skills, and Experience. A formal interview of the shortlisted candidate has been scheduled by admin of AgilePlus in front of selection board using the English language as a medium of communication. The selected candidates as developers have been recruited within the software project according to their area of expertise. However, the customer registers AgilePlus by provid-

ing several credentials which include Name, Email, Contact Number, and Project Domain to complete the registration process. When a new customer registers AgilePlus, the admin sets up a new project by defining its specific name so that if any user is assigned to multiple projects, he can easily switch between these projects. Admin assigns the customer's project to the project manager who assign tasks to the expert developers. Admin also assigns a private key to all the stakeholders so that they can log in to the specific project by selecting its name and entering the private key or passcode which allows them to initiate the first layer of agreement. This private key is recorded in the blockchain database of AgilePlus.

We have also solved the blockchain's scalability problem as its design does not allow to store the huge volumes of records on it. Therefore, AgilePlus utilize IPFS as off-chain secondary storage in order to store all the data of registered customers, developers, messages, group chat, and video conferences for enhancing the performance of blockchain transactions by reducing data load on it. This data has a unique IPFS hash which is stored in the blockchain to view records. However, the data of six layers in AgilePlus is stored in ethereum blockchain.

#### B. SCALABILITY

Scalability refers to the property of a system to manage a large amount of data or storage volume and perform efficiently by adding resources to the system. It is one of the major issues in blockchain as it manages a large amount of transaction data in a short time. Therefore, AgilePlus uses IPFS as off-chain distributed storage system to overcome the scalability challenge in blockchain. IPFS uses cryptographic hash which is stored by using P2P network in a decentralized manner. The large volume of AgilePlus user record has been stored on IPFS instead of blockchain. Customers data stored on IPFS involves their registration details such as Name, Email, Contact Number, and Project Domain while the developers data stored on IPFS includes their registration details such as Name, Contact Number, Email, Qualification, Skills, and Experience Duration in the software industry. Moreover, all the messages, group chat, and video conference meetings on AgilePlus have also been stored on IPFS. This decrease in the large amounts of AgilePlus data on blockchain results in the faster performance of transactions.

#### C. THE LAYERED ARCHITECTURE OF AGILEPLUS

The proposed framework follows blockchain architectural style and the seven-layered blockchain architecture for AgilePlus has been presented in Fig.6.

##### 1) INTERFACE LAYER

The interface layer consists of AgilePlus user-facing app, decentralized applications, and a web portal that connects the customers and developers to the AgilePlus system. All the users having digital wallets in AgilePlus interact with the user interface to initiate the agile software development process.

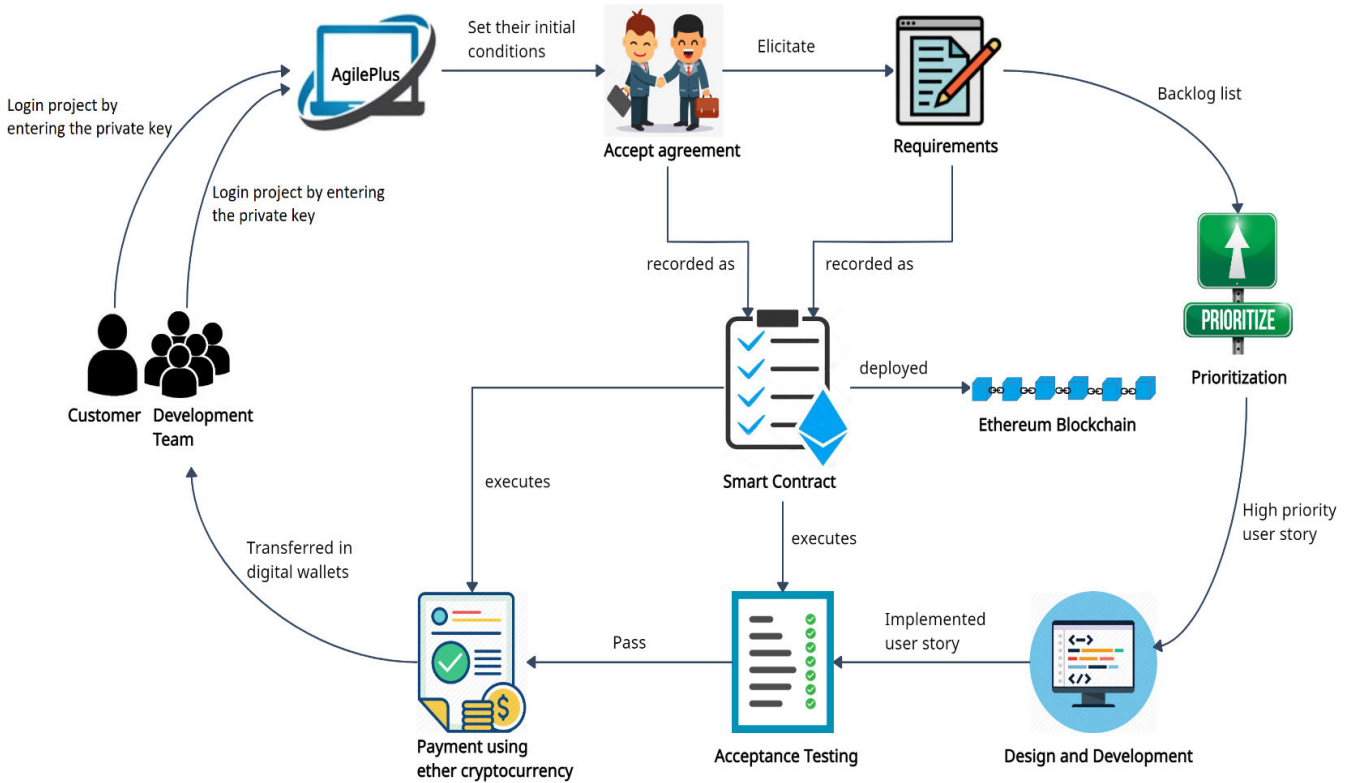


FIGURE 5. AgilePlus framework.

## 2) APPLICATION LAYER

The application layer involves metadata of transactions, payments, and various records in AgilePlus such as posts, prototypes, and project agreements between customers and development teams. This layer is also composed of ether cryptocurrency. It connects the interface layer with the business logic in the form of smart contracts.

## 3) BUSINESS LOGIC LAYER

The business logic layer is comprised of smart contracts within AgilePlus and it deals with the terms and conditions. For smart contracts, the business logic layer acts as an active database which includes invocation, execution of contracts, and rules of communication.

## 4) TRUST LAYER

The trust layer deals with the consensus algorithms such as byzantine fault tolerance and proof-of-work. It also deals with the smart contract's security analysis, verification of newly added blocks, and consensus protocols for transactions. The execution results are stored in blockchain layer.

## 5) TRANSACTION LAYER

In transaction layer, the transactions are triggered by AgilePlus users or smart contracts. It also deals with mining, transaction validation, and block validation.

## 6) INFRASTRUCTURE LAYER

The infrastructure layer consists of peer to peer network which distributes, verifies, and forwards the transaction mapped on ethereum blockchain within AgilePlus. Moreover, it involves verification, communication, and distributed networking mechanisms. The transaction in AgilePlus gets broadcasted to the entire network. Then, each node verifies it according to the predefined parameters and the verified transaction gets stored in blockchain.

## 7) SECURITY LAYER

The security layer is essential for the network of blockchain as it is vulnerable to several security attacks such as 51% attacks. It connects and works parallel to the system and is composed of security algorithms and protocols in order to maintain the security of AgilePlus framework.

The customers and development teams having digital wallets in AgilePlus interact with a user interface that consists of web portal and DApp. Web portal allows the customers and development teams to post, comment, chat, send and receive project payments in the form of ether cryptocurrency. The development teams can also use a currency conversion service in order to convert these ethers into fiat currency. Smart contracts are set of rules which define business logic in AgilePlus and the protocols for transactions are defined by these rules. The chain's credibility is maintained when the rules are defined by smart contracts and the trust layer

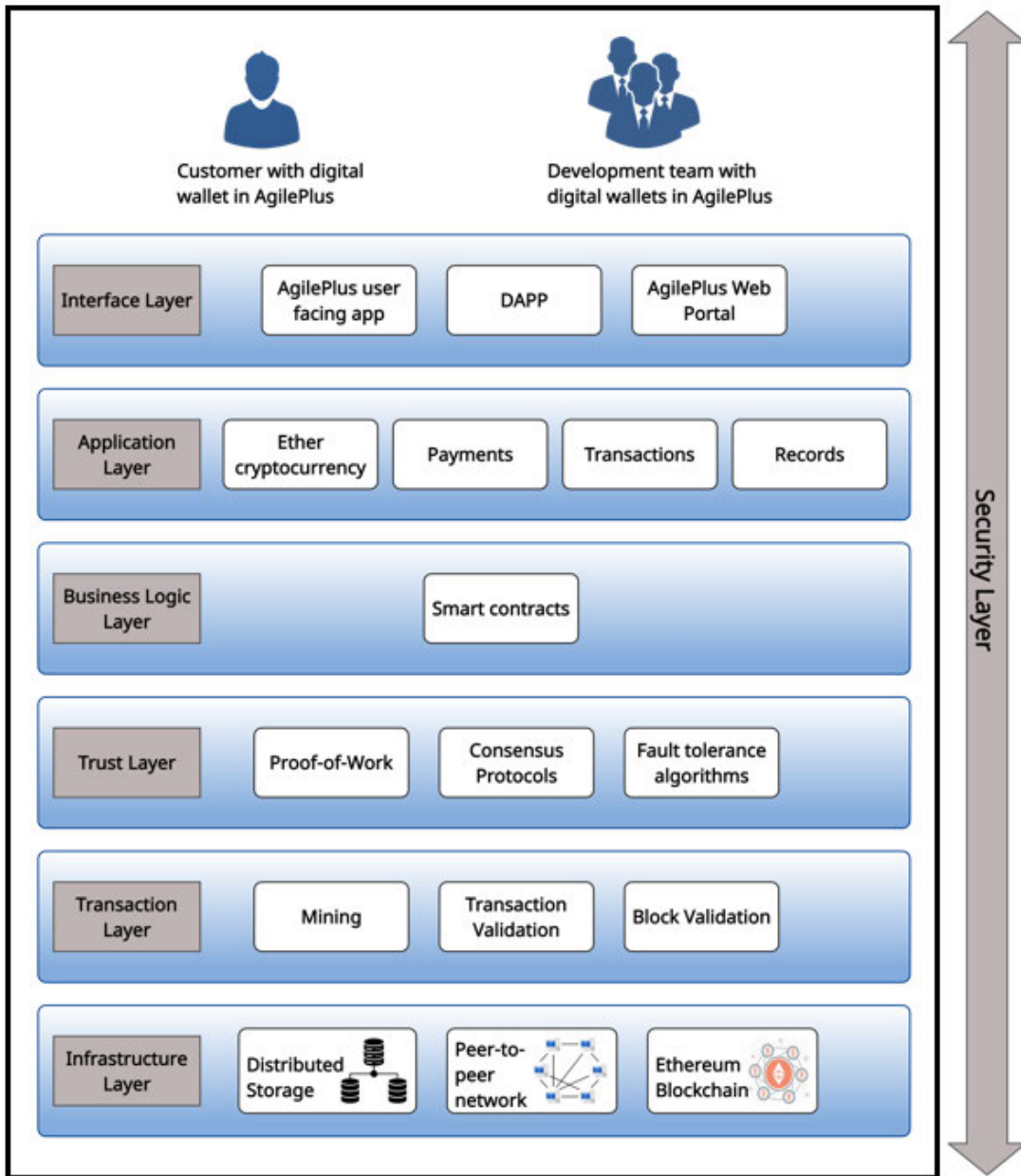


FIGURE 6. AgilePlus blockchain architecture.

integrates consensus algorithms. AgilePlus embeds ethereum blockchain along with proof-of-work implementation for ensuring the chain’s temporal property. Although hashes are stored in blockchain, the data of six layers and private keys to log in to the software project are stored on peer-to-peer networks and in database storage. The security layer is integrated with all other layers to ensure the authorized access of nodes.

**D. THE LAYERED STRUCTURE OF AGILEPLUS**

AgilePlus framework is based on a six-layered structure which has been presented in Fig.7. The transactions of each layer are stored in blockchain notifying all the members of

a blockchain network. Customers and development teams initiate their software projects by following the process in layers of AgilePlus framework as shown in Fig. 8.

**1) LAYER 1: AGREEMENT LAYER**

In the first layer, customers and developers set their initial terms and conditions which are recorded as smart contracts in the blockchain. These terms and conditions must be achieved in order to clear the acceptance test and enable payment option in the AgilePlus framework. An agreement is achieved when the customer and development team reach a consensus. Customer agrees with the terms and conditions



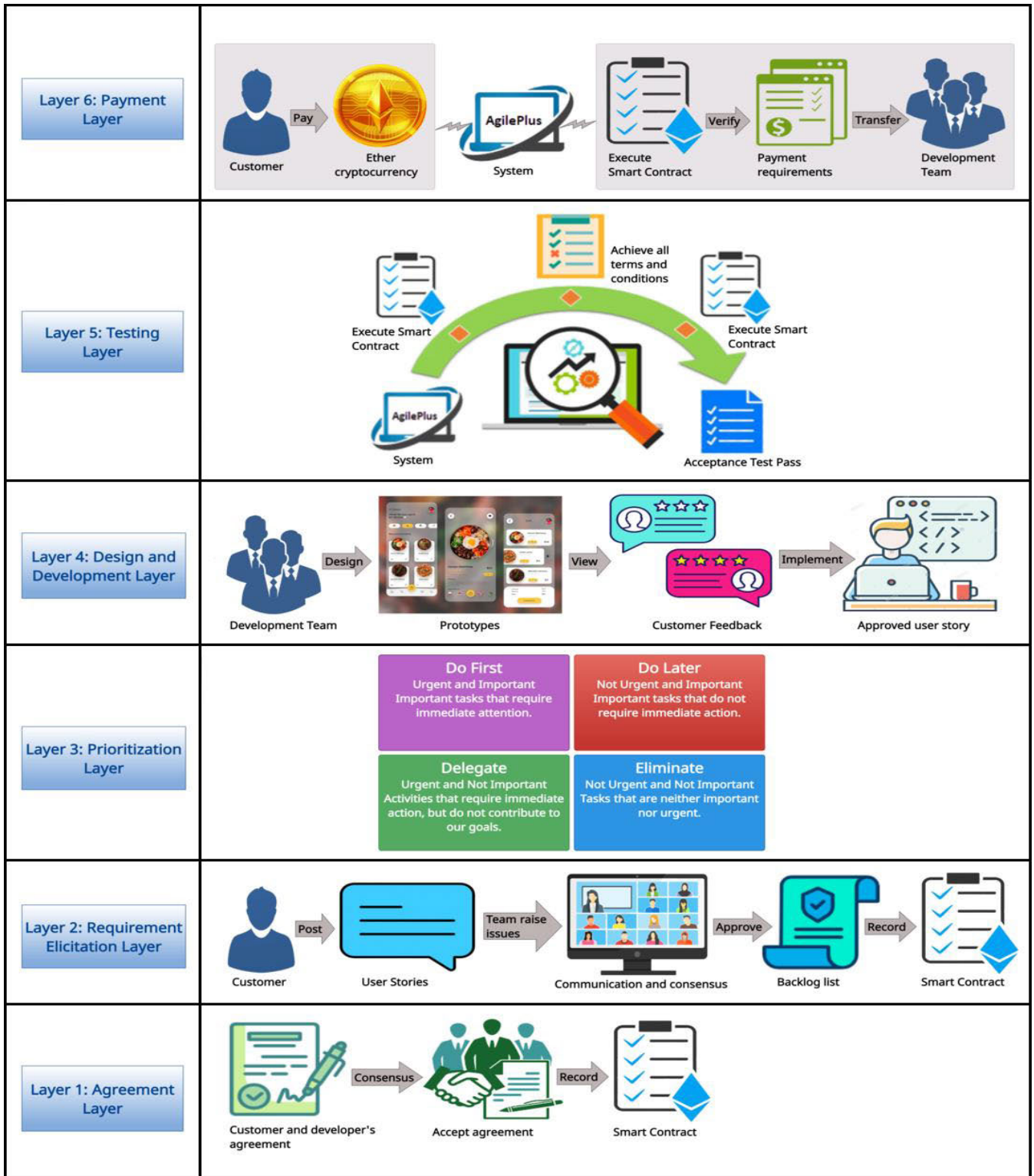


FIGURE 7. AgilePlus layered structure.

set by development team while development team accepts the terms and conditions set by customer. Therefore, agreement layer involves the acceptance of terms and conditions

from both customers and developers to initiate the project. This agreement between customer and development team has been recorded as a smart contract and stored in the

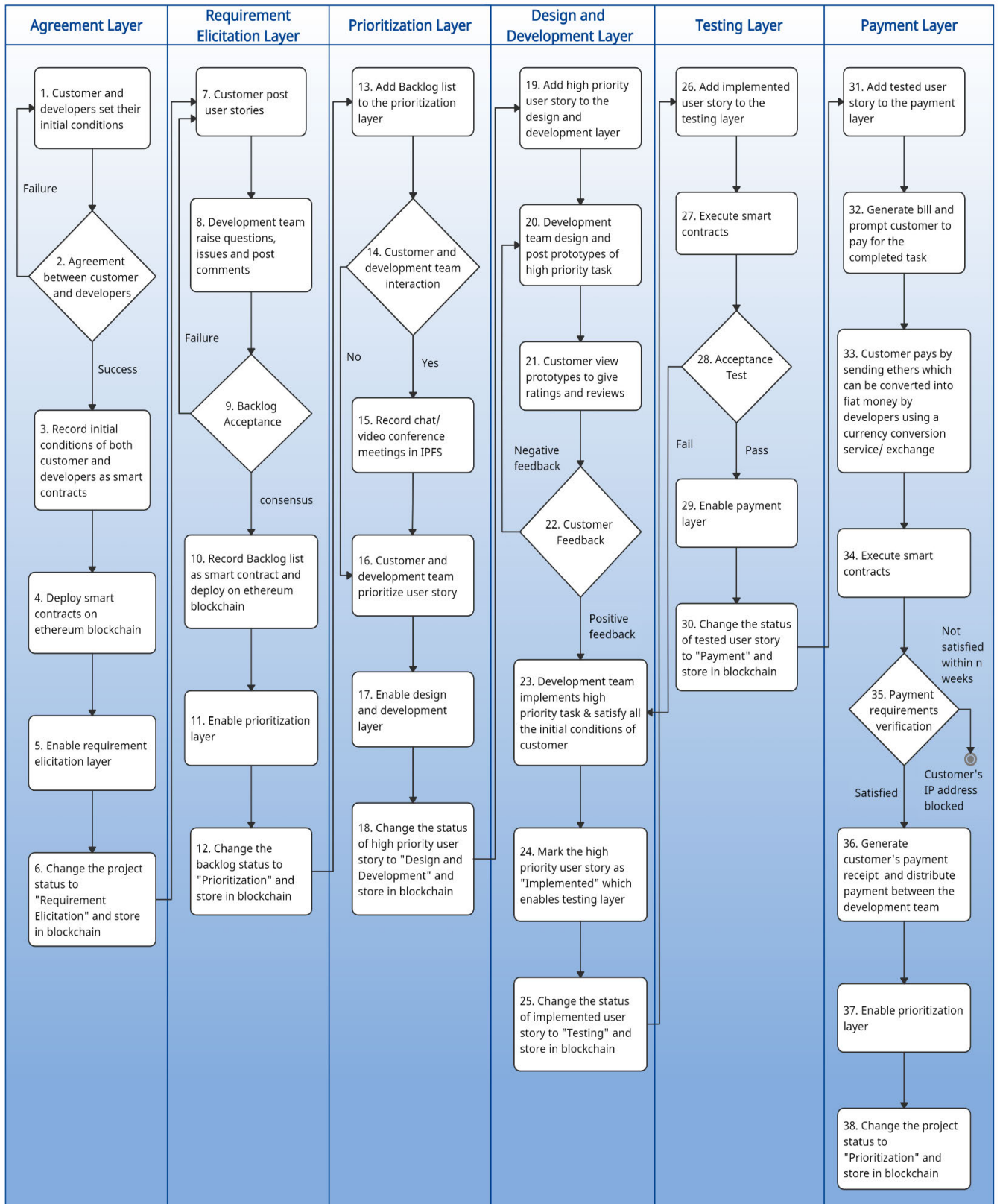


FIGURE 8. Process in layers of AgilePlus framework.

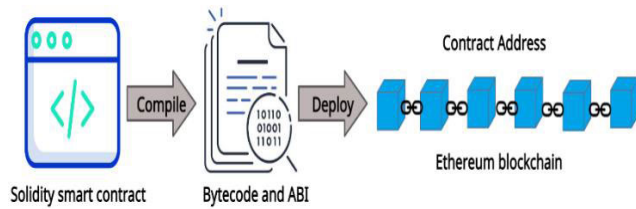


FIGURE 9. Deployment of smart contracts within AgilePlus framework.

TABLE 1. JSON objects for customer’s agreement in agreement layer.

Customer’s Agreement
<pre>{   "Project budget": "\$n decided by the customer"   "Project deadline": "n weeks decided by the customer"   "Each user story’s deadline": "n weeks decided by the stakeholders through consensus"   "Requirements": "Project must be completed within time and budget" }</pre>

blockchain network which enables the requirement elicitation layer. Then, status of the project changes to “Requirement Elicitation” and it has been stored in blockchain notifying all the members in a network.

Ethereum in AgilePlus framework involves the deployment of smart contracts in blockchain i.e the data of transactions and code implementing the logic for transactions, amongst actors. So, the logic and data have been stored irreversibly which composes of smart contracts. According to the immutability principle of blockchain, the deployment of a smart contract does not allow its code to be changed [34]. The developer has to develop another smart contract in case of bugs or errors. He corrects the errors and this new smart contract will be deployed on the blockchain transferring all the existing data to the new smart contract.

In AgilePlus framework, smart contracts have been compiled into an executable ethereum virtual machine (EVM) bytecode or low-level programming language, and application binary interface (ABI). This byte code has been used to deploy the smart contract on ethereum which has been compiled by the compiler while ABI provides the interface between two modules of the program. The specification of contract’s function also uses ABI which has to be invoked. A smart contract’s ABI and contract address created during its deployment have been used to interact with the smart contract. These smart contracts must be compiled before deployment so that EVM can interpret and store the contract. Deployment of the smart contract as shown in Fig.9 has been recorded as a transaction in blockchain.

JSON objects as shown in Table.1 and Table.2 presents the contractual information of the customers and developers deployed on ethereum which are recorded as smart contracts

TABLE 2. JSON objects for developer’s agreement in agreement layer.

Developer’s Agreement
<pre>{   "Each user story’s price": "\$n decided by all the stakeholders through consensus"   "Payment deadline": "n weeks for each tested user story decided by the development team"   "Requirements": "Customer must pay for the completed and tested user story within due date"   "Project Manager": "\$n"   "Software Developer": "\$n"   "Tester": "\$n"   "UX Designer": "\$n"   "UI Designer": "\$n" }</pre>

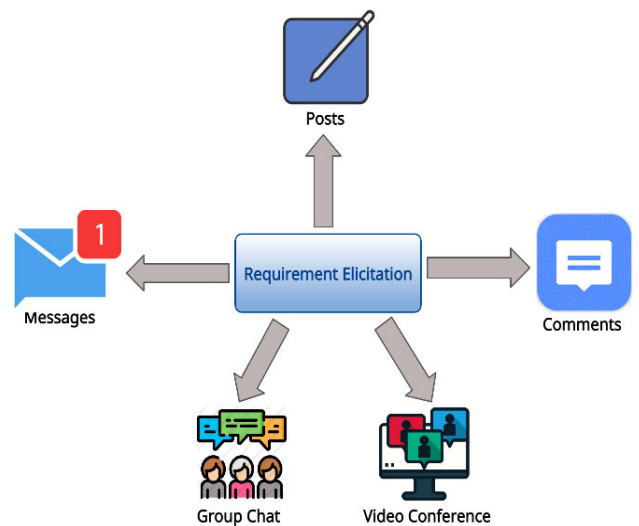


FIGURE 10. Modes of communication for requirement elicitation within AgilePlus.

to ensure that all the terms and conditions fixed by them in the agreement layer have been satisfied.

## 2) LAYER 2: REQUIREMENT ELICITATION LAYER

In AgilePlus, customers post user stories on the virtual wall to elicitate requirements. Users can also use five different modes of communication in this framework as shown in Fig.10. All the user stories posted in requirement elicitation layer have been stored as transactions in blockchain. Then, development team interacts with the customer by replying with the comments on their posts or scheduling a video conference in case of having an issue with any user story. Customer answers all the questions and provides other solutions to the development team. All the posts along with comments have been stored in blockchain while group chat, messages, and video conferences have been stored on IPFS. When customer

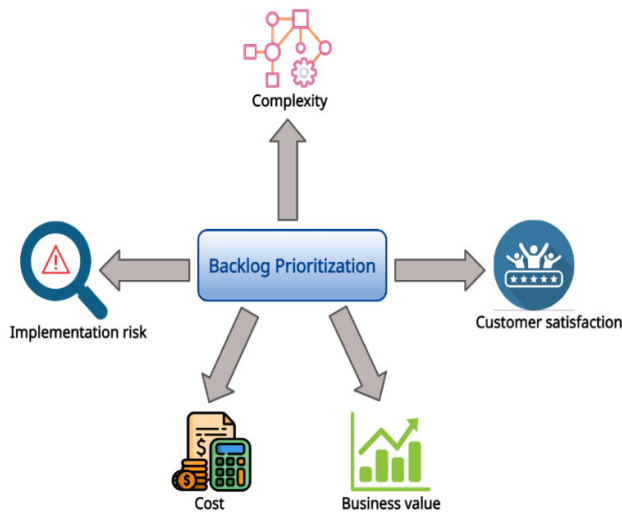


FIGURE 11. Backlog prioritization factors in AgilePlus framework.

and development team reach a consensus on user stories, the backlog list has been recorded as a smart contract which enables the prioritization layer. The status of this approved backlog list changes to “Prioritization” and it moves to the prioritization layer. This transfer has been registered in blockchain notifying all the members of blockchain network. However, if the development team and customer have failed to agree on a backlog list, then customer adds a new list of user stories in the requirement elicitation layer. This step has also been registered in the blockchain network and authorized when both customer and development team agrees with all the posted user stories. The customer can also change or add user stories at any stage of the project. All the members of blockchain network get notified of these changes.

3) LAYER 3: PRIORITIZATION LAYER

The approved backlog list has been added to the prioritization layer where customer interacts with the development team by posting comments, posts, scheduling a video conference or group chat for selecting the important user story that must be completed initially by the development team. A user story in AgilePlus is prioritized keeping in view the following factors as shown in Fig.11. Finally, customer and development team prioritize user story from the backlog list according to its relevance and importance to the system which enables the next layer of design and development. The status of this prioritized user story changes to “Design and Development” and the high priority user story moves to the design and development layer. This transfer has been registered in the blockchain network notifying all the members.

4) LAYER 4: DESIGN AND DEVELOPMENT LAYER

The high priority user story moves to the design and development layer where development team designs and posts the prototypes on virtual wall of this layer. Customer view prototypes and then give feedback by rating and reviewing these

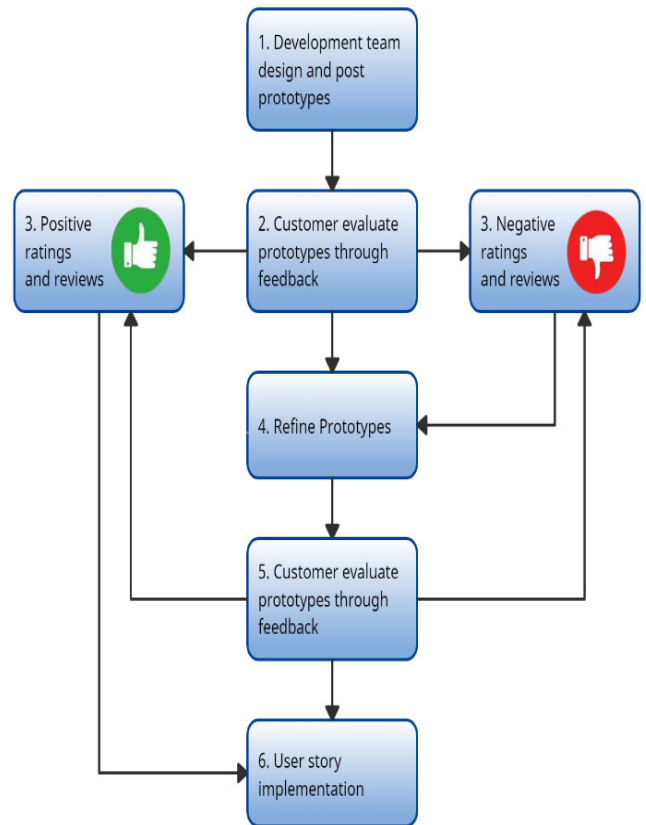


FIGURE 12. Design and development in AgilePlus framework.

prototypes. Both customers and developers can also schedule a meeting on video conference to communicate their ideas and solutions to each other. If customer’s feedback is positive for the posted prototypes, development team implements the specific user story while in case of customer’s negative feedback, the development team redesigns the prototypes for high priority task. When development team implements the user story with positive feedback, status of the high priority user story changes to “Implemented” and is stored in the blockchain. Then, the next layer of testing has been enabled for the implemented user story. The status of this implemented user story changes to “Testing” and stored in blockchain network notifying all the members. Fig. 12 shows the design and development of user stories in the AgilePlus framework.

5) LAYER 5: TESTING LAYER

Testing layer involves testing of each implemented user story. The proposed framework executes the smart contracts for acceptance testing in order to validate implemented user stories. This layer verifies whether all the terms and conditions fixed by the customer in agreement and requirement elicitation layer have been successfully achieved. When the implemented user story satisfies all the terms and conditions present in smart contracts, the acceptance test is considered as “Pass”. This step enables the next layer of payment and status

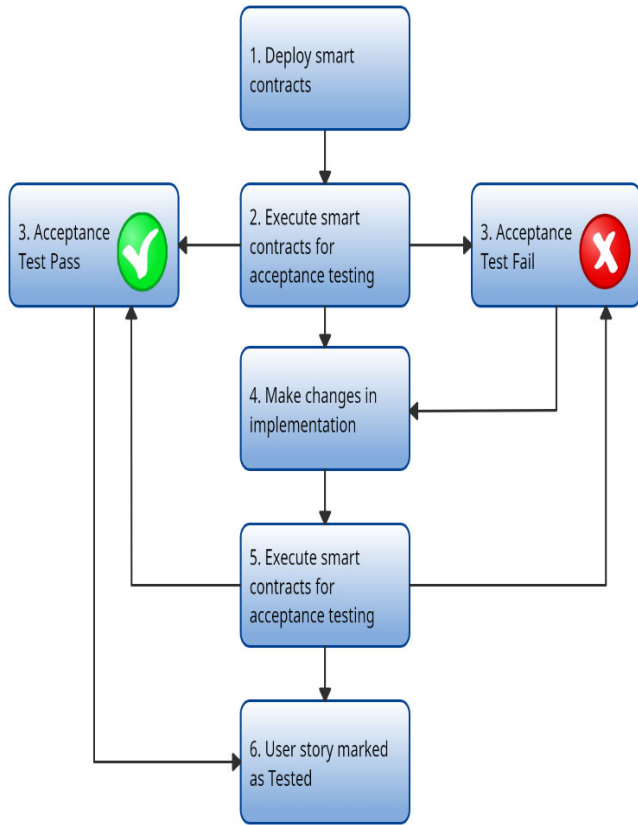


FIGURE 13. Acceptance testing in AgilePlus framework.

TABLE 3. JSON objects for acceptance test 1 in testing layer.

Acceptance Test 1 (Pass State)
<pre>{   "User story ID": "01"   "Requirements": "All the terms and conditions of customer   in agreement and backlog list must be achieved"   "Acceptance test date": "n date"   "Acceptance test status": "Pass" }</pre>

of tested user story changes to “Payment”. All these steps have been stored in blockchain notifying all the members of network. However, the acceptance test is considered as “Fail” if all the terms and conditions have not been successfully satisfied by the development team. In case of acceptance test failure, the development team gets notified which allows them to move implemented user story back to the design and development layer where they fix issues, complete the missing requirements of customer, and implement it again if required. Then, smart contracts execute again for the acceptance testing by moving user story to the testing layer. Therefore in both cases, AgilePlus assigns acceptance test status “Pass/Fail” as shown in Fig. 13, and stored in blockchain

TABLE 4. JSON objects for acceptance test 2 in testing layer.

Acceptance Test 2 (Fail State)
<pre>{   "User story ID": "02"   "Requirements": "All the terms and conditions of customer   in agreement and backlog list must be achieved"   "Acceptance test date": "n date"   "Acceptance test status": "Fail" }</pre>

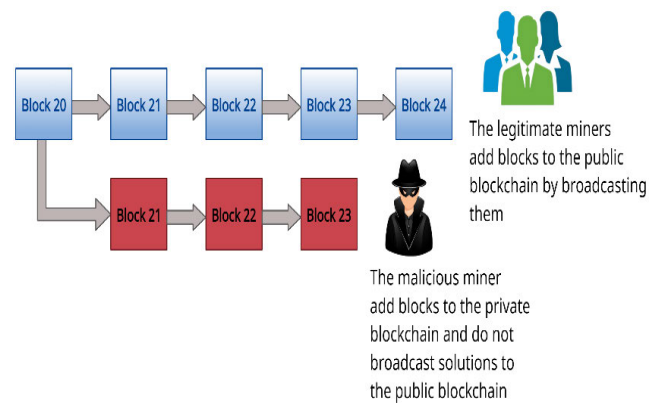


FIGURE 14. 51% Attack.

network notifying all the members. When the development team fails to pass the acceptance test within user story’s due date, the execution of smart contract marks the status of a user story as “Overdue” and automatically assigns penalty to them. The status change and penalty for an overdue user story have been stored in the blockchain network notifying all the members.

JSON objects as shown in Table.3 and Table.4 present the acceptance test results to verify whether all the terms and conditions fixed by customer in the agreement layer have been satisfied.

6) LAYER 6: PAYMENT LAYER

In this framework, a private ethereum blockchain has been used to protect the system against 51% attacks. Miners or attackers use 51% attack as shown in Fig.14 to control more than 51% network’s hash rate for preventing the confirmation of new transactions, halting payments between all the users of AgilePlus, and causing a double-spending problem. A private blockchain provides an extra layer of security as the listed transactions are private. Moreover, it also restricts the participation of nodes to deploy consensus protocols and view transactions. Therefore, private blockchain plays a vital role in preventing 51% attacks and improving payment security in AgilePlus framework.

When the implemented high priority user story clears the acceptance test, the payment layer has been automatically

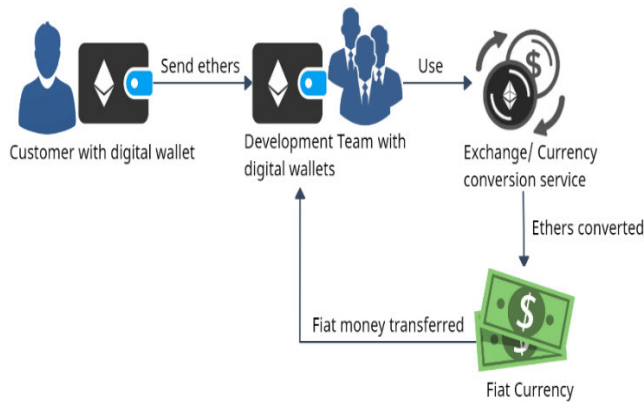


FIGURE 15. Payment and ether cryptocurrency conversion in AgilePlus framework.

TABLE 5. JSON objects for customer’s payment in payment layer.

Payment
<pre>{   "User story ID": "03"   "User story status": "Payment"   "Requirements": "All the terms and conditions of developers in agreement must be achieved"   "Payment due date": "n weeks decided by the development team through consensus" }</pre>

TABLE 6. JSON objects for customer’s penalty (late payment) in payment layer.

Customer’s Penalty (for late payment)
<pre>{   "User story ID": "04"   "Late payment deadline": "n weeks decided by the development team through consensus"   "Payment status": "Late Payment"   "Penalty": "\$n decided by the development team through consensus" }</pre>

enabled. This layer generates a bill for the tested user story and prompts customer to pay for their completed task otherwise, the next user story will not be implemented by the development team. Customer must pay to the development team by sending ethers to their digital wallets in AgilePlus. These ethers can be converted using any currency conversion service into fiat currency such as PKR, AUD, EUR, CAD, and USD, etc. The payment and ether cryptocurrency conversion process in AgilePlus has been shown in Fig.15.

The customer’s payment for the tested user story executes smart contract for the verification of developer’s payment

TABLE 7. JSON objects for customer’s penalty (no payment) in payment layer.

Customer’s Penalty (for non-payment)
<pre>{   "User story ID": "05"   "Late payment deadline": "n weeks decided by the development team through consensus"   "Payment status": "No Payment within n weeks"   "Penalty": "IP address blocked" }</pre>

TABLE 8. JSON objects for developer’s penalty in payment layer.

Developer’s Penalty
<pre>{   "User story ID": "06"   "User story due date": "n weeks decided by the stakeholders through consensus"   "User story status": "Overdue"   "Penalty": "\$n decided by the customer" }</pre>

requirements. If customer successfully satisfies all the payment conditions of development team, a digital payment receipt of customer is generated and the payment is automatically distributed among the team members by transferring ethers into their digital wallets. Then, customer receives an email of the completed task from the development team. All the payment transactions have been registered in blockchain network notifying all the members. This step automatically enables the prioritization layer and changes the project status to “Prioritization”. However, if customer does not satisfy the payment requirements of developers or pays late for the completed task, the execution of smart contract assigns penalty to him. If customer does not pay for the completed user story within n weeks, his IP address automatically gets blocked from the AgilePlus and stored in blockchain notifying all the members. Here “n” is the number of weeks decided by the development team through consensus.

JSON objects as shown in Table.5, 6, 7, and 8 present the contractual details for payment and penalties assigned to the AgilePlus users to ensure trust.

## V. PERFORMANCE

In this section, we have evaluated the performance and effectiveness of our proposed framework, AgilePlus, on real-case scenarios.

### A. PERFORMANCE ASSESSMENT

In order to test the efficiency of AgilePlus, we have used Spyder IDE Version 4.2.5 for implementing the AgilePlus blockchain network in python. For performance testing,

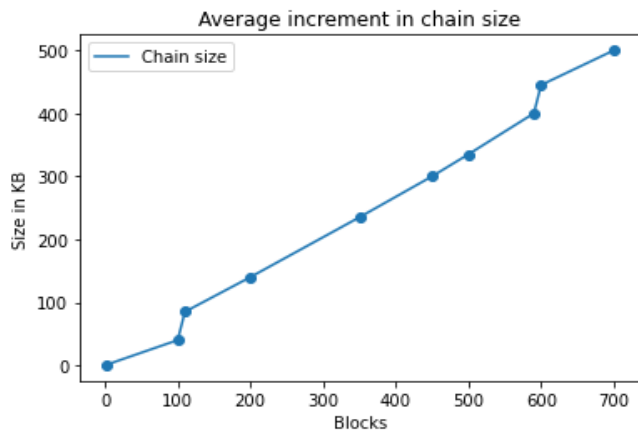


FIGURE 16. The average increase in file size.

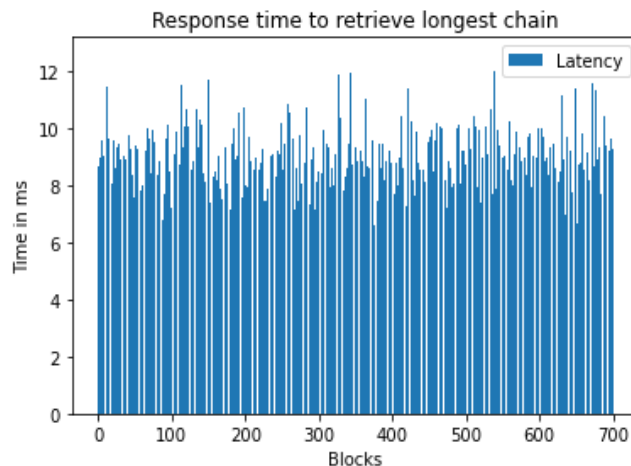


FIGURE 17. Latency for retrieving the longest chain.

Postman tool’s Version 8.5.1 has been used to interact with APIs. Moreover, we have also used matplotlib library in python for the data visualization of chain size and latency.

**B. EVALUATION METRICS**

The metrics used for evaluating AgilePlus framework include block size evaluation and latency. These are briefly explained as follows:

1) BLOCK SIZE EVALUATION

The size of block refers to the amount of data stored in a block. A single block consists of transaction data in chain. The amount of data or block size has been examined to analyze the average increase in size of file/ blockchain.

2) LATENCY

Latency is defined as the delay that occurs when the component of a system waits for another system component to respond to an action. In terms of blockchain network, it refers to the time between the submission of transaction to a network and the first acceptance confirmation by the network.

**C. PERFORMANCE EVALUATION**

This section presents the experimental evaluation of AgilePlus on real-case scenarios in terms of performance. We have implemented the AgilePlus blockchain network where entities transfer data and make transactions in real-time. HTTP requests of “GET” and “POST” have been set up by using the postman tool in order to interact with APIs. We have used the six functions as use cases to validate our proposed framework:

1. get\_chain
2. connect\_node
3. mine\_block
4. add\_transaction
5. is\_chain\_valid
6. replace\_chain

A random transaction has been used to send HTTP requests for 700 repetitions in postman. For 700 blocks, the size of chain having a range between 0.448 KB to 500 KB, increases by adding new blocks with transactions in the blockchain. Fig. 16 shows that the increase in number of blocks increases the blockchain size. The average increase in size is 248 KB. Fig.16 also shows that the newly mined block having a large number of transactions affect the size of files by increasing it from 100 to 110 blocks and 590 to 600 blocks in a linear fashion. The number of transactions in a block is not defined at the development time for this particular use case which results in a slight increase in the size of newly mined blocks.

We have examined all the current chains in blockchain network of AgilePlus and replaced them with the network’s longest existing chain against that specific timestamp by following all the steps required for processing the HTTP request. From this process, we have analyzed the time taken for executing the replace\_chain request of HTTP in order to evaluate the latency for achieving the current longest chain. Fig. 17 shows random delays in the time taken for executing the HTTP requests. The reason behind these random increments and decrements in time for executing a request is the blockchain’s decentralized nature which involves multiple nodes and does not allow any central system to control the data. Furthermore, these multiple servers in the AgilePlus blockchain network use their system capacities which depend on their local machine’s response time against multiple requests, speed of their machines, and internet bandwidth. Such factors cause latency to execute the HTTP requests on different servers.

**D. COMPARISON OF AGILEPLUS FRAMEWORK WITH RELATED WORK**

We have also discussed some parameters that are present in our proposed framework and these parameters have been used for comparison with the related work in this domain. The

**TABLE 9.** Comparison between AgilePlus framework and related work.

Ref No	Related Work	Blockchain-based	Scalability	Communication	Coordination	Testing
[20], [21]	Winbook Framework	N	N	Y	Y	N
[22]	Impact of Social Media in DASD	N	N	Y	Y	N
[23]	ADCC Framework	N	Y	Y	Y	Y
[24]	Enterprise Social Networking Tool (Slack)	N	N	Y	Y	N
[25]	C&C Framework	N	N	Y	Y	N
[26]	Blockchain-oriented Requirements Engineering: A Framework	Y	N	N	N	N
[27]	ChainSoft Platform	Y	N	N	N	Y
[28]	Blockchain applications for agile methodologies	Y	N	N	N	Y
This Article	AgilePlus Framework	Y	Y	Y	Y	Y

comparison between AgilePlus framework and related work has been shown in Table 9.

### 1) BLOCKCHAIN-BASED FRAMEWORK

AgilePlus embeds private ethereum blockchain to ensure security, transparency, trust, and traceability.

**Security:** The integration of private ethereum blockchain in AgilePlus is useful for the following reasons:

- It prevents 51% attacks
- It allows secure payment and distribution between virtual agile teams.
- A private key is required for all the stakeholders to log in to the specific project in order to restrict the participation of unauthorized nodes.

**Transparency:** The software development process is transparent in AgilePlus as data is always available at any time to all the dispersed stakeholders and they get notified with each update.

**Trust:** AgilePlus creates trust by executing smart contracts to achieve customers and developers satisfaction. The

execution of smart contracts verify whether all the terms and conditions of the customers and developers have been fulfilled.

**Traceability:** AgilePlus tracks the work progress of distributed agile teams by storing each step in blockchain and notifying all the members in network.

### 2) SCALABILITY

IPFS has been used as off-chain storage in AgilePlus to solve the scalability problem of blockchain for storing the records of customers, developers, and their communication. This decrease in data load of blockchain results in the faster performance of AgilePlus transactions.

### 3) COMMUNICATION

AgilePlus uses DApp for providing a user-friendly interface to its users. The stakeholders can use five communication modes in all six layers which include posts, comments, group chat, messages, and video conference to ensure effective communication.



#### 4) COORDINATION

Our proposed framework involves consensus in agreement and requirement elicitation layers for coordination between customers and developers.

#### 5) TESTING

AgilePlus execute smart contracts for acceptance testing to verify whether all the terms and conditions of customer have been successfully fulfilled by the development team.

### VI. DISCUSSION

This section formally discusses the experimental results of our scalable, secure and transparent framework, AgilePlus, which is efficient in terms of performance. The proposed framework is useful in the successful development of agile-based projects, providing its users with a collaborative and distributed environment. AgilePlus addresses and overcomes the major challenges of trust, traceability, security, transparency, communication, and coordination in DASD by using blockchain technology due to its decentralized nature, consensus mechanism, distributed storage, and strong security properties. However, the lack of such fundamental factors in distributed environments may result in overdue or project failures, payment clashes, project deal cancellations, customer dissatisfaction, and lack of trust.

The performance evaluation results of our proposed framework indicate that blockchain technology in AgilePlus plays a vital to eliminate all these barriers for the successful development of software projects in DASD. AgilePlus integrates a private ethereum blockchain and provides the users with six layers each having a separate virtual wall to follow the agile development life cycle. We consider that integration of blockchain technology in AgilePlus framework is quite beneficial for the following reasons:

- The decentralized nature of blockchain supports DASD.
- Blockchain provides a user-friendly interface for six virtual walls through DApps to ensure effective communication and collaboration between distributed users across the world.
- Blockchain uses consensus mechanism to improve coordination in DASD.
- The ethereum blockchain in AgilePlus is useful for achieving automatic acceptance testing, verification of developer's payment requirements, and payment distribution to the digital wallets of development team by executing smart contracts.
- Blockchain technology facilitates the distributed users with digital wallets and cryptocurrency for payments such as ether (ETH).
- Customer satisfaction is achieved as blockchain ensures transparency.
- Blockchain technology ensures security as it provides secure payment transactions. Furthermore, AgilePlus uses a private ethereum blockchain to restrict the participation of nodes for preventing 51% attacks.

- Blockchain technology is useful for tracking the work progress of scattered agile teams by recording their performance in blockchain.

The significant findings show that the integration of blockchain technology in AgilePlus successfully overcomes the issues of trust, security, transparency, traceability, communication, and coordination. These characteristics of blockchain avoid project deal cancellations, payment clashes between customers and developers, customer dissatisfaction, and project overdue or failure. Moreover, the utilization of IPFS as off-chain secondary storage greatly reduces the data load on blockchain which results in the faster performance of AgilePlus transactions.

Our experimental results prove that blockchain technology is the major strength of this research which greatly enhances the software development process in DASD. However, the previous studies have not proposed an efficient, transparent, secure, and scalable framework by using blockchain technology to solve the major problems of trust, traceability, security, and transparency in DASD. These existing researches lacked blockchain technology and only solved the issues of coordination and communication in DASD. Therefore, our proposed framework, AgilePlus, outperforms all the previous frameworks and researches for DASD by incorporating blockchain technology.

The limitation of incorporating ethereum blockchain in AgilePlus framework involves high energy consumption as mining of block requires a lot of power. Moreover, data modification is hard, once it is stored in the blockchain. All the performance evaluation results show that the emerging technology of blockchain has a great potential to transform the future of DASD and can drive major changes in offshore software companies.

### VII. CONCLUSION AND FUTURE WORK

The current state-of-the-art tools and frameworks for DASD are inefficient as they lack security, traceability, transparency, and trust. Thus, there is a great need for an efficient framework to solve all these issues. In this paper, we have proposed a novel blockchain embedded framework, AgilePlus, to overcome the major challenges of traceability, transparency, trust, security, communication, and coordination in DASD. AgilePlus uses DApp to provide its users with a user-friendly interface and stores the transaction records in blockchain for tracking the work progress of virtual agile teams. The proposed framework executes the smart contracts for acceptance testing, secure payment, and verification of developer's payment requirements along with the payment distribution between team members by transferring ethers into their digital wallets. The execution of smart contracts automatically assign penalties to its customers for late or non-payments and also to the developers for overdue tasks. We have used a private ethereum blockchain in AgilePlus to provide security against 51% attacks. Moreover, we have also used IPFS as off-chain storage to solve the scalability issue of

blockchain. Finally, our experimental results have proved that integration of blockchain technology in AgilePlus successfully overcomes all the above-mentioned major challenges which greatly enhances DASD.

For future work, we have planned to provide users with the option to give ratings and reviews for AgilePlus at the end of project. We have also planned to integrate a translator in AgilePlus to support different languages. Furthermore, our future work also involves the execution of smart contracts for solving task distribution issues between development teams in AgilePlus framework.

## REFERENCES

- [1] M. D. Kahya and Ç. Seneler, "Geographical distance challenges in distributed agile software development: Case study of a global company," in *Proc. 3rd Int. Conf. Comput. Sci. Eng. (UBMK)*, Sep. 2018, pp. 78–83.
- [2] W. Aslam and F. Ijaz, "A quantitative framework for task allocation in distributed agile software development," *IEEE Access*, vol. 6, pp. 15380–15390, 2018.
- [3] S. Alsaqqa, S. Sawalha, and H. Abdel-Nabi, "Agile software development: Methodologies and trends," *Int. J. Interact. Mobile Technol.*, vol. 14, no. 11, p. 246, Jul. 2020.
- [4] A. M. Gheorghe, I. D. Gheorghe, and I. L. Iatan, "Agile software development," *Inform. Economica*, vol. 24, no. 2, pp. 90–100, 2020.
- [5] D. Kunda, M. Mulenga, M. Sinyinda, and V. Chama, "Challenges of agile development and implementation in a developing country: A Zambia case study," *J. Comput. Sci.*, vol. 14, no. 5, pp. 585–600, May 2018.
- [6] P. T. Robinson, "Communication network in an agile distributed software development team," in *Proc. ACM/IEEE 14th Int. Conf. Global Softw. Eng. (ICGSE)*, May 2019, pp. 100–104.
- [7] M. Singh, N. Chauhan, and R. Popli, "A framework for transitioning of traditional software development method to distributed agile software development," in *Proc. Int. Conf. Issues Challenges Intell. Comput. Techn. (ICICT)*, vol. 1, Sep. 2019, pp. 1–4.
- [8] K. B. Awar, M. S. I. Sameem, and Y. Hafeez, "A model for applying agile practices in distributed environment: A case of local software industry," in *Proc. Int. Conf. Commun., Comput. Digit. Syst. (C-CODE)*, Mar. 2017, pp. 228–232.
- [9] Y. I. Alzoubi and A. Q. Gill, "An empirical investigation of geographically distributed agile development: The agile enterprise architecture is a communication enabler," *IEEE Access*, vol. 8, pp. 80269–80289, 2020.
- [10] G. Ahammad, U. Iqbal Janjua, T. M. Madni, M. F. Cheema, and A. R. Shahid, "An empirical study to investigate the impact of communication issues in GSD in Pakistan's IT industry," *IEEE Access*, vol. 7, pp. 171648–171672, 2019.
- [11] M. A. B. Khalid, A. Farooq, and W. Mahmood, "Communication challenges for distributed teams," *Int. J. Eng. Manuf.*, vol. 11, no. 1, pp. 19–28, Feb. 2021.
- [12] S. Maro, J.-P. Steghofer, E. Knauss, J. Horkoff, R. Kasauli, R. Wohlrab, J. L. Korsgaard, F. Wartenberg, N. J. Strom, and R. Alexandersson, "Managing traceability information models: Not such a simple task after all?" *IEEE Softw.*, vol. 38, no. 5, pp. 101–109, Sep. 2021.
- [13] K. Salah, M. Rehman, N. Nizamuddin, and A. Al-Fuqaha, "Blockchain for AI: Review and open research challenges," *IEEE Access*, vol. 7, pp. 10127–10149, 2019.
- [14] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proc. IEEE Int. Congr. Big Data*, Jun. 2017, pp. 557–564.
- [15] M. Niranjanamurthy, B. N. Nithya, and S. Jagannatha, "Analysis of blockchain technology: Pros, cons and SWOT," *Cluster Comput.*, vol. 22, no. S6, pp. 14743–14757, Nov. 2019.
- [16] O. R. Kabi and V. N. Franqueira, "Blockchain-based distributed marketplace," in *Proc. Int. Conf. Bus. Inf. Syst.* Cham, Switzerland: Springer, Jul. 2018, pp. 197–210.
- [17] I. Acharjamayum, R. Patgiri, and D. Devi, "Blockchain: A tale of peer to peer security," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2018, pp. 609–617.
- [18] P. D. Kate and P. S. Powar, "Review of blockchain architecture implementation in various fields of technologies," *Int. J.*, vol. 6, no. 3, pp. 23–26, 2021.
- [19] A. Khelifi, O. Aziz, M. S. Farooq, A. Abid, and F. Bukhari, "Social and economic contribution of 5G and blockchain with green computing: Taxonomy, challenges, and opportunities," *IEEE Access*, vol. 9, pp. 69082–69099, 2021.
- [20] N. Kukreja and B. Boehm, "Process implications of social networking-based requirements negotiation tools," in *Proc. Int. Conf. Softw. Syst. Process (ICSSP)*, Jun. 2012, pp. 68–72.
- [21] N. Kukreja, "Winbook: A social networking based framework for collaborative requirements elicitation and WinWin negotiations," in *Proc. 34th Int. Conf. Softw. Eng. (ICSE)*, Jun. 2012, pp. 1610–1612.
- [22] G. Lampropoulos and K. Siakas, "Communication in distributed agile software development: Impact of social media-social networking," in *Proc. SQM*, vol. 26, 2018, p. 43.
- [23] M. Younas, D. N. A. Jawawi, A. K. Mahmood, M. N. Ahmad, M. U. Sarwar, and M. Y. Idris, "Agile software development using cloud computing: A case study," *IEEE Access*, vol. 8, pp. 4475–4484, 2020.
- [24] V. Stray, N. B. Moe, and M. Noroozi, "Slack me if you can! Using enterprise social networking tools in virtual agile teams," in *Proc. ACM/IEEE 14th Int. Conf. Global Softw. Eng. (ICGSE)*, May 2019, pp. 111–121.
- [25] R. Qureshi, M. Basher, and A. A. Alzahrani, "Novel framework to improve communication and coordination among distributed agile teams," *Int. J. Inf. Eng. Electron. Bus.*, vol. 10, no. 4, pp. 16–24, Jul. 2018.
- [26] S. Demi, "Blockchain-oriented requirements engineering: A framework," in *Proc. IEEE 28th Int. Requirements Eng. Conf. (RE)*, Aug. 2020, pp. 428–433.
- [27] M. Król, S. Reñé, O. Ascigil, and I. Psaras, "ChainSoft: Collaborative software development using smart contracts," in *Proc. 1st Workshop Cryptocurrencies Blockchains Distrib. Syst.*, Jun. 2018, pp. 1–6.
- [28] V. Lenarduzzi, M. I. Lunesu, M. Marchesi, and R. Tonelli, "Blockchain applications for agile methodologies," in *Proc. 19th Int. Conf. Agile Softw. Develop., Companion*, May 2018, pp. 1–3.
- [29] S. Ferretti and G. D'Angelo, "On the ethereum blockchain structure: A complex networks theory perspective," *Concurrency Comput., Pract. Exp.*, vol. 32, no. 12, pp. 1–11, Jun. 2020.
- [30] G. A. Oliva, A. E. Hassan, and Z. M. J. Jiang, "An exploratory study of smart contracts in the Ethereum blockchain platform," *Empirical Softw. Eng.*, vol. 25, pp. 1864–1904, Mar. 2020.
- [31] K. Wu, Y. Ma, G. Huang, and X. Liu, "A first look at blockchain-based decentralized applications," *Softw., Pract. Exp.*, vol. 51, no. 10, pp. 2033–2050, Oct. 2021.
- [32] R. Kumar and R. Tripathi, "Blockchain-based framework for data storage in peer-to-peer scheme using interplanetary file system," in *Handbook of Research on Blockchain Technology*. New York, NY, USA: Academic, 2020, pp. 35–59.
- [33] M. U. Javed, M. Rehman, N. Javaid, A. Aldegheshem, N. Alrajeh, and M. Tahir, "Blockchain-based secure data storage for distributed vehicular networks," *Appl. Sci.*, vol. 10, no. 6, p. 2011, Mar. 2020.
- [34] M. S. Farooq, M. Khan, and A. Abid, "A framework to make charity collection transparent and auditable using blockchain technology," *Comput. Electr. Eng.*, vol. 83, May 2020, Art. no. 106588.



**MUHAMMAD SHOAB FAROOQ** is currently working as a Professor in computer science with the University of Management and Technology, Lahore. He is also an Affiliate Member of George Mason University, USA. He possesses more than 26 years of teaching experience in the field of computer science. He has published many peer-reviewed international journals and conference papers. His research interests include theory of programming languages, blockchain IoT, internet of vehicles, machine learning, distributed systems, and education.



**ZAREEN KALIM** was born in Lahore, Pakistan. She received the B.S. and M.S. degrees in software engineering from the University of Management and Technology, Pakistan. Her research interests include agile software development, requirements engineering, and blockchain.



**SAIM RASHEED** was born in Pakistan. He received the Ph.D. degree in the area of brain–computer interaction from the University of Milan, Italy, in 2011. He has almost 20 years of experience in academia and research. He has been actively participating and contributing in many international conferences and journals. His current research interests include human–computer interaction, gamification, and natural language processing.



**JUNAID NASIR QURESHI** received the M.S. degree in computer science from the University of Central Punjab, Lahore. He is currently pursuing the Ph.D. degree in computer science with the University of Management and Technology, Lahore, Pakistan. He is currently working as a Senior Lecturer with the Department of Computer Science, Bahria University Lahore Campus (BULC). He have 12 years of professional experience in education and industry. His research inter-

ests include the Internet of Things (IoT), blockchain, distributed systems, human–computer interaction, software engineering, agile project development, and education.



**ADNAN ABID** (Member, IEEE) received the B.S. degree from the National University of Computer and Emerging Science, Pakistan, in 2001, the M.S. degree in information technology from the National University of Science and Technology, Pakistan, in 2007, and the Ph.D. degree in computer science from the Politecnico di Milano, Italy, in 2012. He spent one year at EPFL, Switzerland, to complete his M.S. thesis. He is currently working as a Professor with the Department of

Computer Science, University of Management and Technology, Pakistan. He has almost 40 publications in different international journals and conferences. His research interests include computer science education, information retrieval, and data management. He has served as a reviewer in many international conferences.

...