Blockchain Applications Testing

Table of Contents

Introduction

Blockchain is a digitized, distributed ledger which is shared, replicated and synchronized among the members of a public or private peer-to-peer computer network comprised of its user machines. The ledger permanently records the history of asset exchanges amongst the members of the network in a linear and chronological order. Every transaction recorded in the ledger has a timestamp and unique cryptographic signature associated with it. Once the information gets stored in the blockchain, it cannot be changed or tampered. All the confirmed and verified transactions are combined into a block and chained to the most current block to form a blockchain.
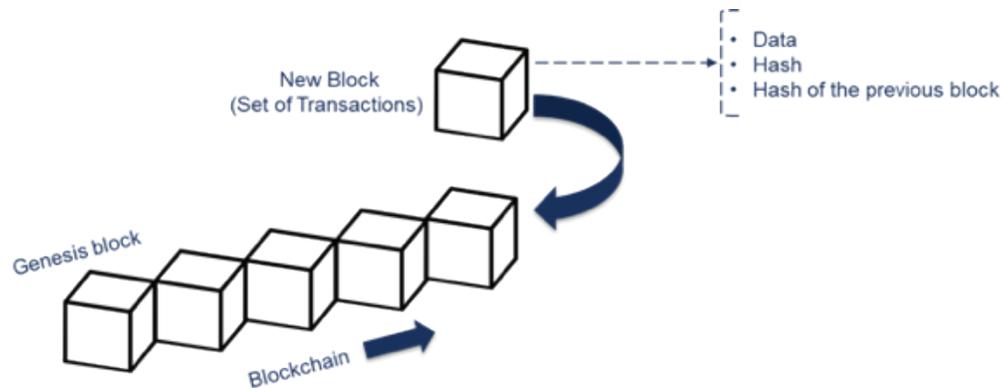
How Blockchain works?

Let's understand how Blockchain works with a simple example –
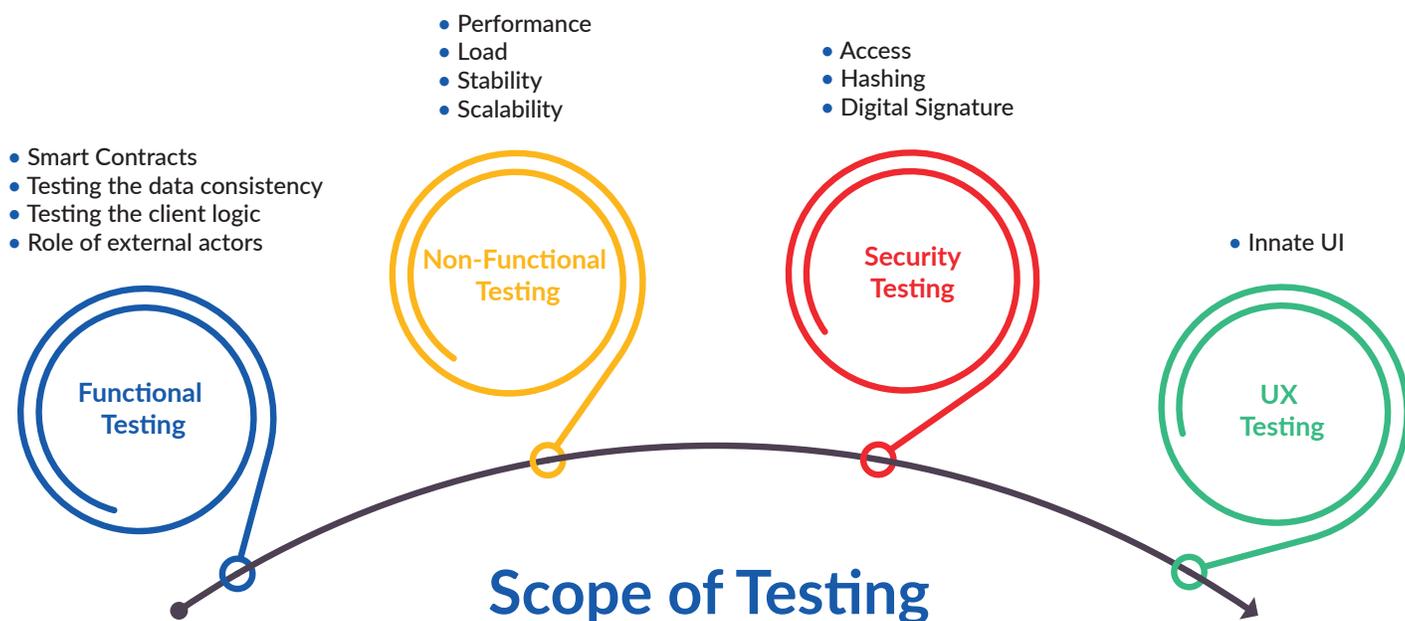**Suppose Mr. X wants to transfer money to his friend Mr. Y. Below is how it works in Blockchain.**
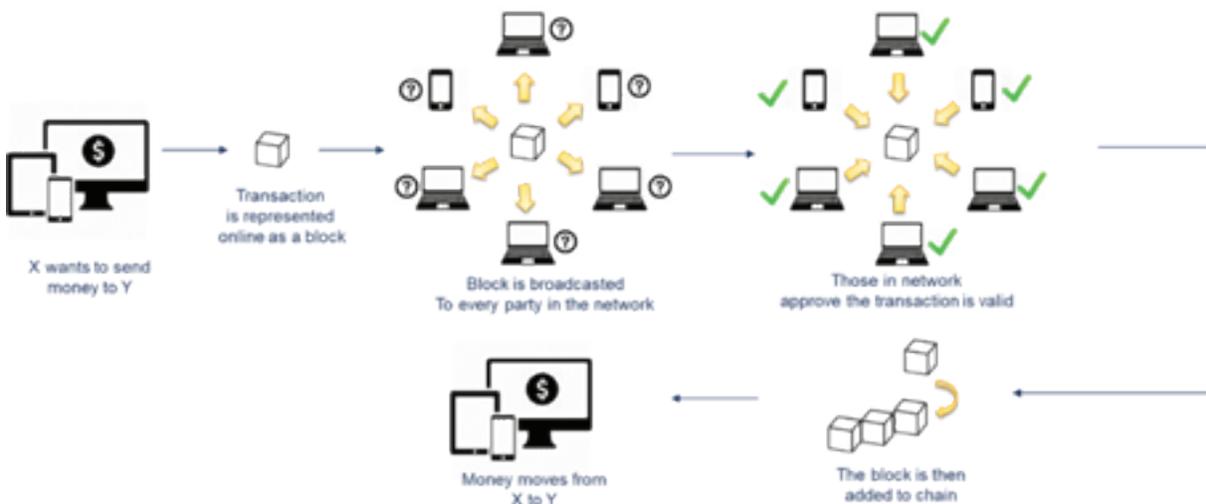
1. X wants to send money to Y.
2. Transaction is represented online.
3. The requested transaction is broadcasted to a P2P network consisted of computers known as nodes.
4. The network of nodes validates the transaction and the user's status using known algorithms.
5. A verified transaction can include cryptocurrency, contracts, records or other information.
6. Once verified, the transaction is linked to other transactions to create a new block of data for the ledger.
7. The new block is then appended to the existing Blockchain, in a way that is permanent and unalterable.
8. The transaction is finally completed, and Y gets the money.

Testing Blockchain-based applications

Testing blockchain-based applications is challenging because there is a significant change in the technology itself. Besides all requisite testing: functional, integration, security, performance and specialized testing; blockchain testing requires smart contract testing, node testing, and advanced level non-functional testing.

- Performance
- Load
- Stability
- Scalability

- Access
- Hashing
- Digital Signature

- Smart Contracts
- Testing the data consistency
- Testing the client logic
- Role of external actors

Non-Functional Testing

Security Testing

- Innate UI

Functional Testing

UX Testing

Scope of Testing

Generic Testing Services:

Functional Testing:
Functional testing of the basic components, the systems and its workings is essential. Testing here is done to check the effectiveness of use-case scenarios and the specific business processes involved.

Integration Testing:
Integration testing is important since integrating a blockchain app with various systems and environment often poses great difficulty. It becomes essential to ensure that the interfaces between the components, the integrations, and the different parts of the system are functioning cohesively. This is essential to ensure performance consistency. The challenge arises when there is a lack of proper information on all interfaces along with their response status.

Security Testing:
Node and shared ledger are the most important parts of the blockchain for securing a blockchain application. The aim is to identify if the application is vulnerable to attacks, assess if the authorization systems are robust, identify if the system protects the data and can ward off malicious attacks, etc. Along with this, it is imperative to test integrity, authentication, confidentiality, and non-repudiation during security testing.

Performance Testing:
For performance testing, a replica of the production environment is required. However, to replicate the level of transaction and the delay in processing the transaction, it is difficult to execute performance testing. The performance of an application and the latency vary with networks as well as transaction size. Performance testing in blockchain includes identifying performance bottlenecks, defining the metrics for tuning the system, and assessing if the application is ready for production.

API Testing:
Consider a transaction using API for communication. The transaction must be validated against specific rules to generate an update order which the blockchain then distributes. The API then receives a confirmation that the blockchain has been updated. Testers must validate the interaction of applications in and out of the blockchain ecosystem at every process step to validate whether API requests and responses are formatted and handled correctly.

Specialized Testing:

Smart Contract Testing:

Smart contract testing is specialized testing. Smart contracts lie at the core of the blockchain validation process. Testing of smart contracts calls for simulating all possible expected and unexpected conditions for all possible contracts. Testing looks at business logic combinations and appropriate execution of all the transactions in the context of a dynamically changing and expanding the network.

Node Testing:

Blockchain functions in a peer-to-peer distributed network through network nodes using a specific protocol for authentication. This makes it essential to achieve a consensus across all nodes on the order in which the transactions are added to the network. Node testing for the consistency of transactions is needed. This calls for the testing of the consensus protocol to determine that all the transactions get stored in the proper sequence. This would have to be the case under normal conditions and also under conditions when nodes fail simultaneously or when they do not participate in the network for some time. These tests help ensure that the nodes in the network sync with other validating peers and the integrity of the network and shared ledger are maintained throughout.

Along with all this, testing for block size, chain size, transmission of data, and testing of cryptographical data are also essential to blockchain applications. Given the sheer number of nodes and the various combinations and transactions that need to be validated, test automation may well prove critical to the success of blockchain applications. While these factors affect the implementation of blockchain technology, the challenges are not limited to these as testing would depend on how blockchain has been implemented for your application.
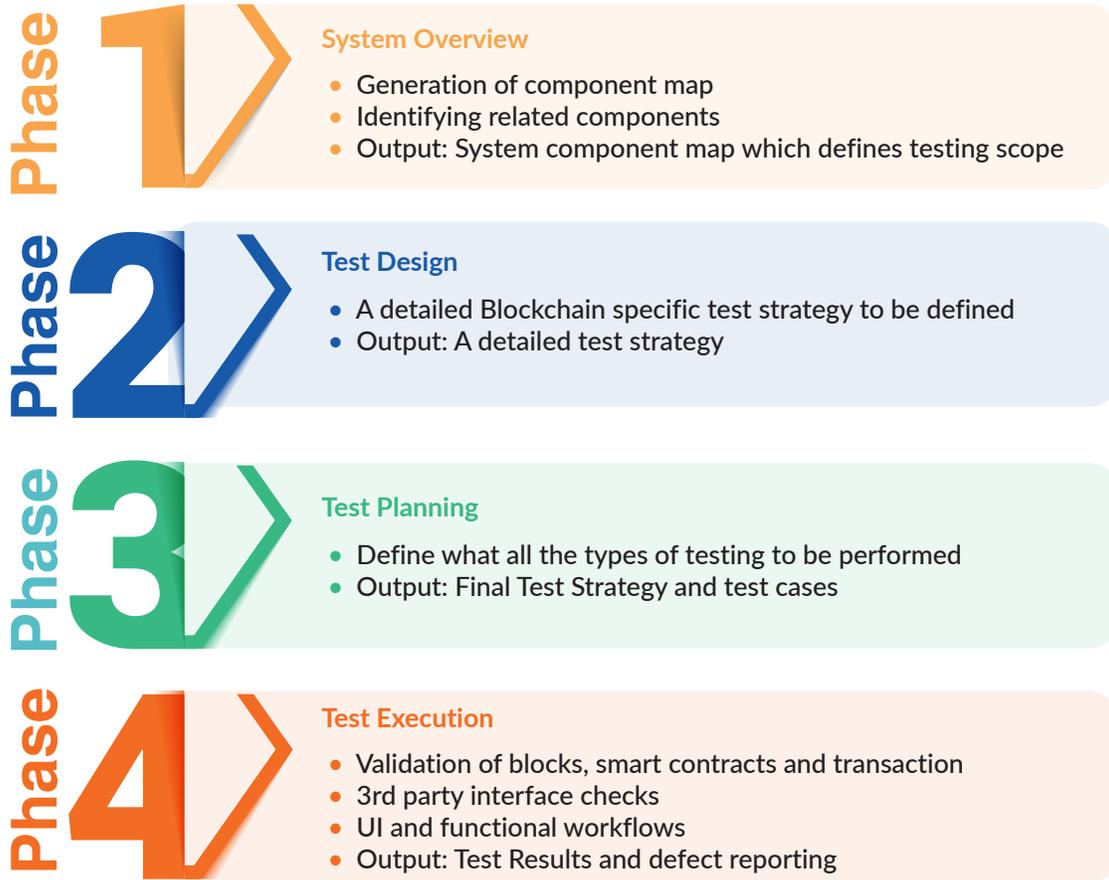
Testing Phases and Lifecycle:

Phases:

Testing Phases	Methodology and Tools
Unit Testing	TDD with Mock Stubs
System Testing	Verifying contracts, Blocks and updating through scripts
Integration Testing	Blockchain Environment setup and local testing
Functional / UI Testing	Automated tests for Front-end using Selenium / Appiumv

Testing Lifecycle:

Phase 1

System Overview
- Generation of component map
- Identifying related components
- Output: System component map which defines testing scope

Phase 2

Test Design
- A detailed Blockchain specific test strategy to be defined
- Output: A detailed test strategy

Phase 3

Test Planning
- Define what all the types of testing to be performed
- Output: Final Test Strategy and test cases

Phase 4

Test Execution
- Validation of blocks, smart contracts and transaction
- 3rd party interface checks
- UI and functional workflows
- Output: Test Results and defect reporting

Tools for Blockchain Testing:

Exonum:
It is a framework that allows you to test the operation of the whole service. It allows you to test transaction execution and APIs in the synchronous environment. Exonum has a fundamental security property to check the rare cases of compromised blockchain nodes.
Reference: https://exonum.com/

Ganache:
It's the most-used library for testing Ethereum contracts locally. It's a personal blockchain   for Ethereum development you can use to deploy contracts, develop your applications, and run tests.
Reference: https://truffleframework.com/

Ethereum:
It is a development framework with inbuilt libraries for testing.
Reference: https://www.ethereum.org/

Truffle:
It is a development framework with the ability to write automated tests for your contracts in both JavaScript and Solidity. A world-class development environment, testing framework and asset pipeline for blockchains using the Ethereum Virtual Machine (EVM), aiming to make life as a developer easier.
Reference: https://truffleframework.com/

Corda:

It is an open-source distributed ledger platform. It has a built-in testing feature to help with:

- Writing contract tests
- Integration testing
- Writing flow tests
- Load testing

Reference: https://www.corda.net/

Populus:

This framework provides some powerful utilities for testing blockchain contracts. Testing in Populus is powered by Python framework.
Reference: https://populus.readthedocs.io/en/latest/testing.html

Conclusion

Blockchain is an emerging technology and like any new technology, how well and how comprehensively we can test will play a key role in its success and adoption.

Author:

Naresh Padmanabhan (Paddy) is the Practice Manager for Multi-Channel and Blockchain testing at Hexaware Technologies. He has over 13 years of experience in various stages of software development life cycle and Paddy is a certified Blockchain expert from Blockchain council. He also holds a master's degree in Computer Applications from a reputed Indian University.

About Hexaware

Hexaware is the fastest growing next-generation provider of IT, BPO and consulting services. Our focus lies on taking a leadership position in helping our clients attain customer intimacy as their competitive advantage. Our digital offerings have helped our clients achieve operational excellence and customer delight by 'Powering Man Machine Collaboration.' We are now on a journey of metamorphosing the experiences of our customer's customers by leveraging our industry-leading delivery and execution model, built around the strategy— 'Automate Everything, Cloudify Everything, Transform Customer Experiences.'

We serve customers in Banking, Financial Services, Capital Markets, Healthcare, Insurance, Manufacturing, Retail, Education, Telecom, Professional Services (Tax, Audit, Accounting and Legal), Travel, Transportation and Logistics. We deliver highly evolved services in Rapid Application prototyping, development and deployment; Build, Migrate and Run cloud solutions; Automation-based Application support; Enterprise Solutions for digitizing the back-office; Customer Experience Transformation; Business Intelligence & Analytics; Digital Assurance (Testing); Infrastructure Management Services; and Business Process Services.

Hexaware services customers in over two dozen languages, from every major time zone and every major regulatory zone. Our goal is to be the first IT services company in the world to have a 50% digital workforce.

- -

NA Headquarters
Metro 101, Suite 600,101 Wood
Avenue South, Iselin,
New Jersey - 08830
Tel: +001-609-409-6950
Fax: +001-609-409-6910

India Headquarters
152, Sector – 3
Millennium Business Park
'A' Block, TTC Industrial Area
Mahape, Navi Mumbai – 400 710
Tel : +91-22-67919595
Fax : +91-22-67919500

EU Headquarters
Level 19, 40 Bank Street,
Canary Wharf,
London - E14 5NR
Tel: +44-020-77154100
Fax: +44-020-77154101

APAC Headquarters
180 Cecil Street,
#11-02, Bangkok Bank Building,
Singapore - 069546
Tel : +65-63253020
Fax : +65-6222728

www.hexaware.com | marketing@hexaware.com