COMP 6710: software quality in DLT/Blockchain

Frank McFadden, Alex Lewin

10/14/20

1 Executive Summary

This paper aims to present an overview of software quality Assurance (SQA) within the context of Distributed Ledger Technology (DLT) enabled applications. Through this paper, we address the definitions and fundamental architecture of DLT and blockchain and discuss several candidate approaches for addressing software quality as they pertain to a DLT ecosystem. Further, we will focus on some of the DLT-related project details that most affect software quality efforts.

To conclude, we discuss the challenges of upholding software quality Standards in a rapidly emerging field, especially with respect to DLT's position in existing industries. We conclude that DLT is an emerging field of technology which invites a new set of challenges, distinguishing the industry from the traditional software landscape.

1.1 What is DLT?

Distributed Ledger Technologies refer to applications that implement **decentralized**, **transactional** databases to exchange information and prove authenticity.

- **Decentralization** refers to a major departure from traditional database architecture. Traditionally, databases are typically **centralized**, where all data is centralized in a single location. This creates virtual libraries of data, which pose security issues analogous to physical libraries. The centralized database model is sometimes referred to as the **honeypot model**, as it organizes assets into a single, attackable interface for cyber criminals. On the other hand, **decentralized** databases spread information across a network of machines that work together to maintain security.
- **Transactional** databases refer to a special method of storing data. Traditional databases represent information by descriptively characterizing the properties of given entities.

For example, a database on a banking website could describe the content of an account by stating the following:

$$Customer\ X \rightarrow balance = \$100 \qquad \text{Traditional Example}$$

. This format represents information by a series of transactions. A bank implementing a

transactional database would represent accounts by stating the following:

$$Customer\ X paid\ Customer\ Y \rightarrow \$20 \qquad \text{Transactional Example}$$

This leaves the onus of tracking running balances to the users.

The decentralized and transactional nature of DLT is enabled by the use of **cryptographic signatures**. Whenever a transaction is written to the ledger, the author cryptographically signs the entry. This process proves the authenticity of the entry and prevents tampering. This allows the ledgers to maintain integrity over its contents.

1.2 What is Blockchain?

While DLT has not yet infiltrated the vernacular of the greater public, the term *blockchain* has absolutely garnered the attention of the masses. While blockchain and DLT are closely related, they have notably different meanings. Blockchain exists as a *subcategory* of Distributed Ledger Technologies. DLT refers to any decentralized, transactional database.

Blockchain, on the other hand, refers to a DLT system that organizes transactions into "blocks" and stores the cryptographic hash of the previous block onto the current block, creating a *chain* of *blocks* (ie. *blockchain*). This is an important distinction in order to understand the technology, but for the purposes of this paper, we will use the two terms interchangeably.

---

2 Candidate Approaches for Achieving software quality

Blockchain represents the culmination of many topics in software development - including enterprise software, open source projects, and organizational cooperation. Because of this, when analyzing SQA in the context of blockchain, there are several potential approaches.

These strategies include implementing traditional SQA standards (ISO), applying open-source approaches to SQA (QualOss), and analyzing the maturity of organazations by addressing the software development process (CMM). For the purposes of this paper, we will be focusing on aspects of SQA specific to DLT and blockchain.

---

3 Blockchain-Specific SQA

3.1 Scope

The **scope** of a blockchain refers to its accessibility to the public. More specifically, the scope of a blockchain defines who is allowed to write transactions to the ledger. Today, there are two types of blockchains that dominate the market: **public blockchains** and **private blockchains**. Each has its merits and drawbacks. Both types of blockchains have a wide variety of use cases. In some cases, the decision between a public versus a private blockchain is not completely clear.

For example, the research conducted by Precht, Wunderlich, and Gomez lays out "an approach based on a criteria catalog in order to help companies to choose the correct blockchain implementation" (Precht).

3.1.1 Public Blockchains

Public blockchains allow any anonymous user to write transactions to the ledger. Typically, the costs associated with running the network of a public blockchain are payed for by individual transaction fees. These fees are incurred by the author of the change and are written to the ledger as another transaction. An advantage of public blockchains is the idea of "complete accessibility". This concept is very important to blockchain/DLT activists.

With regards to software quality, public blockchains are preferable due to the mechanisms that thwart bad actors. Given that they are publicly accessible, the networks must be designed to operate properly within the context of bad actors.

Drawbacks mainly involve performance speeds. Due to the massive potential size of public blockchains, the networks often incur delays for transactions. With Bitcoin, the largest public blockchain, transactions can be pending for hours or even days. For some mission-critical industries, this delay may not be acceptable.

3.1.2 Private Blockchains

Private blockchains (also called permissioned blockchains) restrict access to approved parties. This is usually done by issuing "invitation tokens" to wallets entering the network. The cost of running the network is typically incurred by an organization or corporation. Private blockchains are being implemented in a variety of industries, such as supply chain, identity, and digital receipts.

Advantages of private blockchains include faster transaction speeds - due to fewer nodes on the network - and more granular control over the protocols. Examples of private blockchains include Hyperledger (maintained by the Linux foundation) and some implementations of Ethereum.

3.2 Verification Protocol

Verification protocol refers to the process of confirming the authenticity of new nodes on the network. The verification protocol answers the following question: "How do I know that new nodes can be trusted?"

This process involves a very complex technical implementation, facilitated by cryptographic signatures. Further, there are several existing verification protocol paradigms that aim to achieve a high level of security while maintaining a high efficiency.

3.2.1 Single Verifier

The "single verifier" approach is the simplest procedure to understand. It involves naming a single-trusted verifier node that is responsible for affirming all other nodes. An advantage of this strategy is a high confidence of security. Drawbacks include reduced scalability, due to the bottleneck of one node taking on the entire burden of verification.

3.2.2 M-of-N

M-of-N refers to the use of a complex voting system to confirm new nodes onto the network. The term "M-of-N" refers to the cryptographic algorithm used to implement this system. Advantages include better scalability and a higher degree of decentralization. Drawbacks mainly involve the potential for a bad actor to be selected as the verifier node. This could potentially be a vector for attack on the network.

3.2.3 Ad-Hoc

An Ad-Hoc approach refers the idea of having a changing verification protocol that depends on the current state/context of the network. One possible implementation could be to use a Single Verifier approach by default and to transition to a M-of-N approach in the event of a surge in traffic.

3.3 Consensus Protocol

The consensus protocol refers to the process that the network undergoes to approve information written to the ledger. It answers the following question: "How do I know that new transactions can be written to the ledger and that new blocks can be added to the chain?" Existing solutions include "Proof of Work" and "Proof of Stake".

Considerations for which strategy to adopt mainly involve a comparison between security and performance.

3.3.1 Proof of Work

Proof of Work refers to the strategy for obtaining consensus that has been implemented by Bitcoin. Advantages of this strategy include its incredible security. With the computational capabilities of modern computers, the Proof of Work approach deems comprimise completely unfeasible. Basically, this approach forces attackers to acquire greater computational power than the combined capability of the entire network. Given the size of large, public blockchains, this seems impossible. Downsides of this approach involve the intense computational power required for upkeep. The act of powering a node in a Proof of Work network is commonly referred to as "mining".

3.3.2 Proof of Stake

Proof of Stake refers to a completely different strategy of maintaining network integrity. In these networks, groups of nodes with highest "seniority" vote on approving blocks onto the chain. Seniority is generally determined by the number of blocks in a chain (ie. chain length) and by the age of nodes in the network. The philosophy is that bad actors should be introduced onto the network more recently that good actors. Advantages primarily involve performance when compared to Proof of Work, as mining is not required. Drawbacks involve the potential for the Proof of Stake algorithm to pick a bad actor as the "leader" node.

3.4 Use Cases

The use case is the single most important factor to consider in the context of SQA in blockchain. There is a massive array of use cases being explored accross virtually every

industry.

When considering software quality with respect to the different use cases of blockchain, SQA efforts are usually correlated with the complexity of the problem/solution. As the use case becomes increasingly complex, the SQA efforts become increasingly complex as well.

3.4.1 Cryptocurrency

The most ubiquitously known use case of DLT is the invention of crypto currency. While the technology required to facilitate crypto currency is incredibly complex, this use case is the most fundamental and generic implementation of a transactional database. The only item written to the ledger is "Wallet X pays Wallet Y". In this use case, the only SQA efforts involve maintaining the integrity and performance of the network.

3.4.2 Zero Knowledge Proofs

Zero Knowledge Proofs are a blockchain-enabled technology that allows entities to cryptographically prove statements without disclosing any information to support the claim. For instance, someone could prove ownership over a vehical without giving any information about the vehical. With regards to software quality, zero knowledge proofs are fundamentally "different from traditional programs." For example, "a Solidity smart contract cannot be simply patched after it has been compiled and added to the blockchain because of the nature of Ethereum and blockchain. Quality assurance of smart contracts is therefore extremely important in the development process" (P. Chapman).

Given the incredible complexity of this technology, strategies to achieve SQA mainly target the software development process. Some recommend the use of Test Driven Development (TDD) to thwart out the potential faults in Zero Knowledge Proofs.

3.4.3 Self-Sovereign Identity

Self-Sovereign Identity (SSI) culminates several areas of blockchain into an incredibly complex and lucridve set of use cases. SSI allows individuals to prove aspects of their identities through the use of verifiable credentials. Through SSI, an individual could prove they are over 21 years old programatically, without needing to show a phisical ID or revealing their birth date. In an SSI ecosystem, individuals could hold pieces of their identity, in the form of credentials, in an SSI-enabled digital wallet.

In order for SSI to be adopted, governments will have to be involved in approving the quality of the software supporting the technology.

---

4 Concluding Thoughts

On October 31st 2008, an author under the pseudonym Satoshi Nakamoto published a white paper that described an implementation for a technology that implemented cryptography to create a completely decentralized currency. Practically overnight, Bitcoin and other cryptocurrencies alike rocked the financial industry and inspired a massive surge of innovative projects that leveraged the same technology.

Today's society has entrusted some of its most mission-critical industries into blockchain-enabled systems, which has placed a multi-billion dollar incentive to develop the products.

In the case of DLT, "Quality Assurance (QA) is at the heart" of its development "as users of today demand seamless user experience, flawless functionality and utmost security and a single glitch can wipe out credibility among end customers" (Sarkar). This nebulous foundation is critical for the future success of the new technology.

Together, blockchain's immaturity and its undeniable value create the perfect conditions for a massive software failure. Given its place in society today, these requirements pressure engineers and organizations to prioritize the SQA concerns that plague new technologies.

---

5 Sources

- P. Chapman, D. Xu, L. Deng and Y. Xiong, "Deviant: A Mutation Testing Tool for Solidity Smart Contracts," https://eds-a-ebscohost-com.spot.lib.auburn.edu/eds/detail/detail?vid=2&sid=8ac14c92-9eab-4f8b-a4ca-222020702953%40pdc-v-sessmgr01&bdata=JnNpdGU9ZWRzLWxpdmUmc2NvcGU9c2l0ZQ%3d
- Sarkar, A., & Mazumder, J. (2017). Quality Assurance in Blockchain. http://qaistc.com/2017/wp-content/uploads/2017/09/stc-2017_quality-assurance-in-blockchain_arpan_jibendu_cognizant.pdf
- Applying Software Quality Criteria to Blockchain Applications: A Criteria Catalog (Precht, Hauke, Wunderlich, Stefan, Marx Gamez, Jorge) http://hdl.handle.net/10125/64511
