An information system for the evaluation of blockchain smart contracts' effectiveness

Author: Alexander Panayotov

Abstract: Blockchain smart contracts have quickly become a focal point of research and development. Their autonomous, decentralized, transparent and secure nature allows for enforcement of agreement between multiple parties, with no need for trust as a prerequisite and no intermediaries to facilitate the relationship. However, this shift in paradigm, the significant difference with conventional software, and a variety of decentralization specifics has made development of reliable and effective smart contracts extremely difficult. The tendency has been shown by the short, yet turbulent history of smart contracts, full of numerous attacks, exploits, thefts, and failures. In this paper, the authors address this by developing an information system for the automated evaluation of the effectiveness of blockchain smart contracts. The system implements a previously created formal model that is used to calculate the effectiveness of a smart contract, based on specified factors. The information system provides as a result the level of effectiveness in the form of an output metric called Smart Contract Index of Effectiveness, which quantifies the level of potential risk to the effectiveness of a smart contract. System requirements, architecture, technologies and process are discussed. Direction for future development of the information system and further specification of its scope and functionality is also provided.

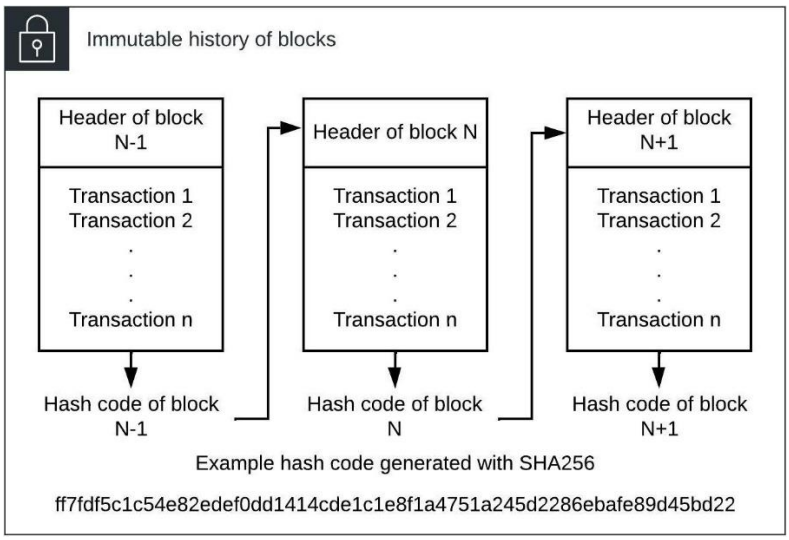
Keywords: information system, blockchain, smart contracts, measuring effectiveness, index of effectiveness

JEL: M15

1. INTRODUCTION

Blockchain is a form of Distributed Ledger Technology (DLT), further specified and modified to facilitate secure communication and transfers of value, based on Peer-to-Peer (P2P) sending and receiving of transactions. In technical terms, a blockchain protocol consists of three parts: a consensus mechanism that makes sure that all participants agree on what happens in the protocol; an immutable history of all past transactions, grouped in units called blocks that are securely linked to one another, in consecutive order, using cryptography; and a P2P network. The immutable history is visually represented in Fig. 1. The first implementation of such a protocol is Bitcoin [1]. Numerous variations have since been created, although only a few are important and widely recognized [2].

Fig. 1. Historical data organized in linked blocks in a blockchain ledger.



1.1. Blockchain importance

Blockchain has rapidly become a point of great research and industrial interest. It promises to revolutionize trade, digital identity, process optimization, privacy, security and more. Yet, until recently, this emerging field of knowledge has largely been intuitive, driven by community support and seed-stage investor funding. Development of decentralization has advanced, however, enough that even transnational governing bodies, such as the European Commission, adopted their own blockchain strategy [3]. Additionally, as blockchain smart contracts have found their way into serious business processes, it becomes necessary to formulate a formal, systematic approach towards developing and integrating effective smart contracts in real-world solutions. That would ensure a transition from experimenting with different use cases towards solving specific business problems using smart contracts, in a reliable and effective fashion.

1.2. Smart contract development problems

Smart contracts (SC), in simple terms, are a type of computer code that exists and executes in a blockchain environment (or sometimes simply referred to as blockchain code). Their practical purpose is to be a secure, censorship-resistant and self-executing way of managing assets and processes in a completely transparent and trustworthy way. Execution of this code is entirely deterministic, while being completely decentralized. The process ensures that all participants in a blockchain protocol arrive at a single ‘truth’, which is recorded in the immutable ledger. However, that determinism comes at a cost – significant reduction in protocol efficiency, and many specific conditions, that if not known, or are not clearly and entirely understood, easily lead to compromised effectiveness.

A lot of factors influence smart contract effectiveness and efficiency. Those are usually related to development practices and techniques. Sizable amount of research has gone into analyzing best development practices and creating tools in an attempt to

mitigate different risks [4,5]. Typically, this includes: static and dynamic analysis of smart contract code, visualization of execution flow, classification of bug and other vulnerabilities, testing of code, code formatting, memory optimization, and others. However, most of that research is heavily focused around a specific niche property of smart contracts in a specific context. Jing Liu and Zhentian Liu (2019) conducted a survey and found that most of the literature is focused around security assurance and correctness verification of smart contracts [5]. While those are indeed important factors for reliable smart contracts, it is by far not enough to achieve effectiveness in broader sense. That makes it difficult to get a well-rounded view of writing effective smart contracts, especially considering the avantgarde nature of the technology, the lack of standards and regulation as well as the limited expertise available.

1.3. Model for the evaluation of Smart contract effectiveness

A. Panayotov and P. Ruskov (2022) classify the types of problems related to smart contract creation in five groups: security, privacy and regulation, performance, protocol execution context and technical development [6]. They identify over twenty factors (across those problem groups), which serve as inputs to a specially constructed model that evaluates the effectiveness of smart contracts. The process is the following:

1. All the factors are assigned a Boolean value
2. The now quantified parameters are given as inputs to the model
3. The model is executed
4. The output result is a metric that quantifies the level of risk (and therefore expected effectiveness) to the smart contract. That metric is called “Smart Contract Index of Effectiveness” (SC_E) [6].

2. DESIGNING AN INFORMATION SYSTEM (IS) IMPLEMENTING THE EVALUATION MODEL

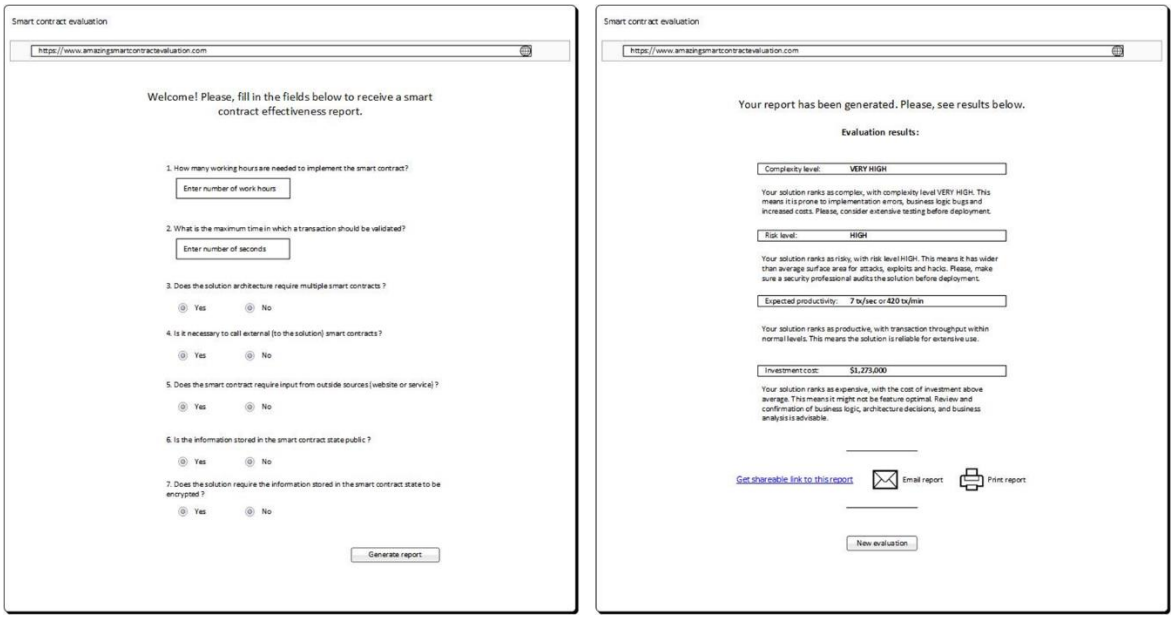
In this paper, the author presents an Information system that implements the above-mentioned evaluation model. The purpose of the system being to help blockchain smart contract developers create better smart contracts by identifying potential weaknesses and keeping in mind common issues and pitfalls before the actual development begins. Such Having in mind the immutable nature of smart contracts, such an approach would significantly accelerate the development process and would reduce the cost of development and problem-fixing later in the lifecycle.

To achieve this, a low-fidelity prototype was created in order to explore the possibilities for creating the most appropriate information system for smart contract evaluation. The prototype is visually shown in Fig. 2. It mainly included yes/no questions that are manually marked via radio buttons, and a few other fields for providing specific data (numbers). The result was a short high-level report about the smart contract, mainly general statements about level of risk and effectiveness, as well as expected productivity and cost of development. This initial version was shared among peers of the author

(blockchain developers and blockchain experts with relevant expertise, including academic researchers).

The feedback received was positive, with respondents confirming that such a tool would indeed be useful and necessary, and would help developers, researchers and blockchain professionals alike in the creation and analysis of blockchain solutions.

Fig. 2. Low-fidelity prototype of the information system.



Importantly, suggestions for improvement were pointed out. The most important findings include:

- Users found it cumbersome and unnecessarily time-consuming to manually input each parameter (as was the prototype assumption) before getting a report. A much more efficient way of generating the data was necessary.
- It was remarked that the IS would be difficult to use in the case when different adaptation of smart contract business logic is needed. In other words, if a user wants to change just a few of the parameters (to consider an alternative business logic), they would need to fill the form with all parameters all over again, which was considered a disadvantage.
- In addition to the previous comment, it was suggested that it would be very useful if the system allows for memorizing different scenarios (based on different input parameter data) so that they can be easily compared and analyzed, and therefore the most suitable solution can be chosen according to a predefined criteria and use case.
- It was found that the resulting analysis report did not need to give data such as estimation about development cost (in the prototype “Investment cost”) as this was dependent on many factors and was pointed out to dilute the overall

result. The report would need to be highly focused on the expected effectiveness of the smart contract and any potential areas of significant risk.

2.1. Information system requirements

Based on the feedback considerations described above, and after detailed analysis, a framework of requirements and characteristics was formulated that would guarantee the effective and efficient use of the system. In order to address those perceived disadvantages and to achieve its purpose, IS would need to fulfill the following requirements:

a) A template with predefined input parameters should be used as input to the model, instead of manually inputting each parameter. Ideally, an electronic table (e.g., spreadsheet) would be used. This would allow for the quick upload and analysis of the (parameters) dataset.

It would also give flexibility in creating and managing that dataset, as it can be done outside of the system, in a familiar environment, such as MS Excel or Google Sheets. As a result, this would also reduce the load on the IS, since it will be used only for model execution and not for manual inputting and handling of data. This, in turn, would make its structure less complex.

This approach also addresses the other major concern with change in the business logic, and respectively, the input to the model. A user would simply need to modify just the parameters concerned in a spreadsheet and then upload to the system, instead of having to manually input all parameters all over again.

At the same time, a user can prepare in advance multiple datasets (for different smart contract scenarios), and quickly upload them to the system to get results for easy comparison. This would significantly increase productivity and usefulness, and would make it far easier to analyze results, allowing to easily pinpoint where (and which) factors have the biggest impact on a specific criterion in a use case.

b) For the IS to be meaningful and useful, and to allow for multiple evaluation of different smart contract designs, that would mean it needs to be able to handle many concurrent requests. A cloud service will solve this issue, as it can provide as much resources as are necessary to answer the load in a specific moment in time.

c) IS would need to provide data narrowly focused around the effectiveness of a smart contract, and not nice-to-have, but unnecessary metrics. For that reason, it should provide the user with the overall effectiveness index as a number, and not a general conclusion (i.e., “high effectiveness” or “low effectiveness”), and also a breakdown for the major risks to effectiveness: security, data privacy, and complexity. Other metrics are omitted.

d) Users should not need advanced technical skills, such as programming, in order to prepare the dataset, use the system and interpret the results.

e) The Interface and overall process should be as simple as possible to reduce the possibility of error or misunderstanding.

f) The technological stack used to develop the IS should allow for easy maintenance of the final product and also provide flexibility so that additional modules and functionality can be added as needed, without impacting the already existing solution or changing the listed requirements.

g) Provide the result in a meaningful timeframe, in this case assumed to be under ten seconds.

2.2. Technologies used

Based on the considerations and requirements described above, a decision was made on how to proceed with the development of the information system. After detailed comparison of different technological frameworks and tools, the Google Cloud – specifically Google Apps Script - was chosen as the most suitable [7]. It provides a bundle of all (interconnected) technologies needed to rapidly build a web application.

The main reasons motivating that choice include:

- Fulfils all of the listed requirements
- Free to use (for limited resources)
- Integrates well with most of the existing technologies and online productivity tools
- Google tools and products are familiar to most users
- Allows flexibility to easily add and integrate across the entire Google suite of technologies, which gives freedom to expand and modify the system as required
- Google provides a complete set of all necessary technologies, tools and features, that work very well together, to build small-to-mid-sized applications, all in the same place. This significantly accelerates the development process.

Specifically, the exact technological set used for developing the system is as follows:

- Google Script – a programming language based on JavaScript, used for rapid development of applications, that integrate well with the broader Google ecosystem.
- Google App Script web IDE – the native integrated development environment used for writing Google Script applications and server-side code.
- Google Sheet services – a library that facilitates the communication between Google spreadsheets and the server-side of the application written in Google Script.
- Google Spreadsheet – the familiar electronic table software.

- JavaScript – a well established programming language for writing web applications.
- HTML & CSS – the languages that are used to provide structure and style to a web application.

2.3. Assumptions and constraints

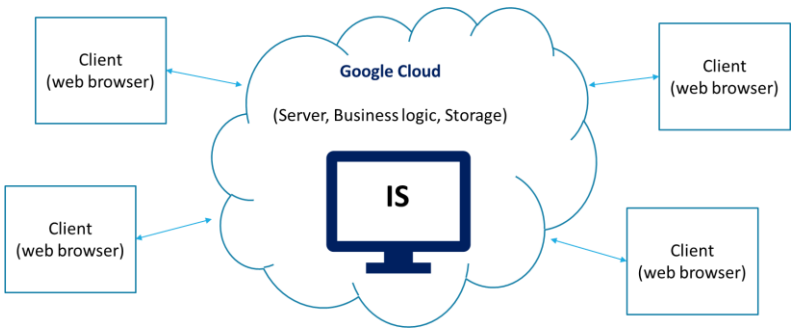
The chosen technologies introduce some constraints on the work and capabilities of the system. The first one is a limit of 2000 requests for any period of 24 hours (payment is required beyond that point). The second limitation is the inability to keep previously generated contract evaluations, as there is no database used, so output data does not persist, and is only available at execution time. Further, the choice of a single vendor introduces the possibility of a single point of failure, i.e., being entirely dependent on the politics and management decision of a single provider, and those can unexpectedly change from time to time. Also, the level of security is not high, due to the framework being optimized for rapid application development, and not high-end solutions. And last but not least, this particular technological stack is not widely popular (compared to more traditional technologies), so its credibility might be an issue with prospective users.

The general assumptions under which the system is designed is that: thorough business analysis was carried out prior to evaluating a smart contract, and so business objectives are set, the purpose of the contract is clearly defined and priorities identified; the input dataset is created (or at the very least – audited) by a blockchain expert.

2.4. Architecture

The information system was designed with the classic client-server model in mind, where clients (web browsers or other devices) request data from a server (the web application) as a result of a specific process, executed at the server environment. This can be represented visually in Fig. 3. The clients in this particular case are web browsers. This was chosen for simplicity purposes and also because users are well used to working in a cloud (browser) environment.

Fig. 3. High-level architectural view of the information system.



As a result, to make the system as easy and as friendly to use as possible, all the user-related work is designed to happen in a single browser (connected to the Google cloud).

2.5. Process

In order to use the system to receive an evaluation of a smart contract, several steps need to be executed in order:

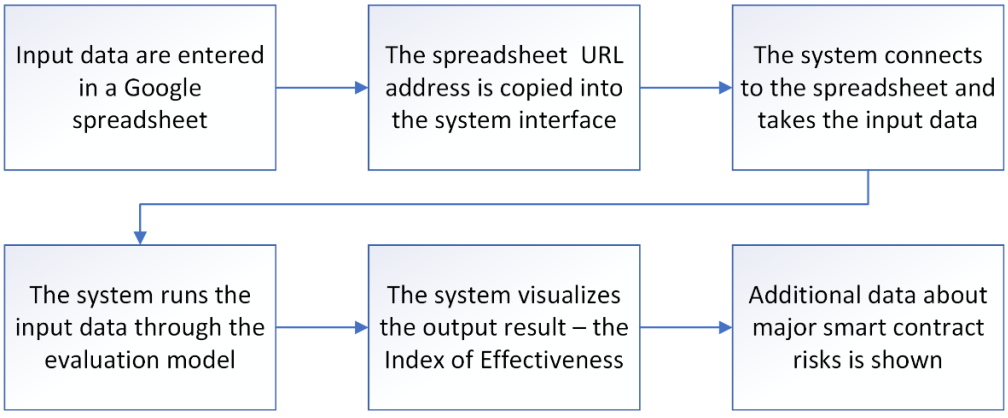
1) The dataset with input parameters should be prepared.
2) The dataset should be provided to the IS.
3) The IS runs the data through the evaluation model and provides the output result.

The entire lifecycle process is visually represented in Fig. 4. The user is actively involved in the first two steps, which essentially means: evaluating the factors that impact effectiveness, assigning them a Boolean value, and structuring that information in a spreadsheet file. Providing that file to the IS was designed to be effortless – as simple as copying the file's URL into the web application, which is a trivial task.

2.6. Practical advantages

The way this information system is designed provides many advantages for a range of blockchain professionals.

Fig. 4. The process of using the information system for smart contract evaluation.



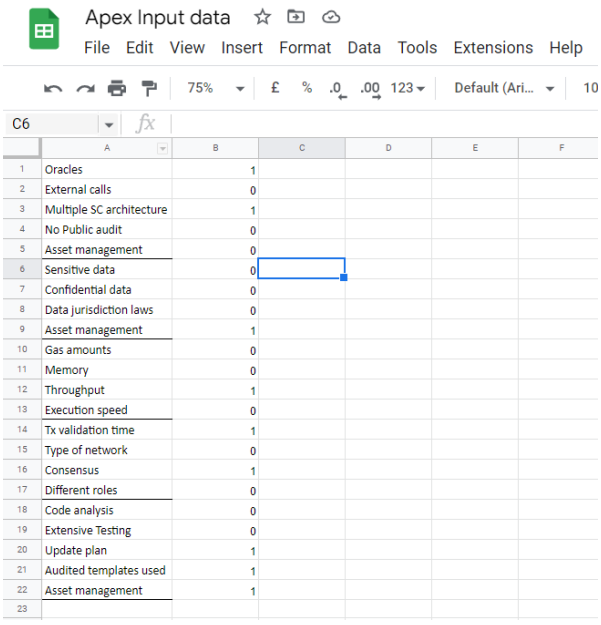
It automates the evaluation of smart contracts, turning it into a useful tool in the hands of experts, which unlocks their productivity. The immediate practical advantage to using the system is eliminating subjective judgement and human error. Providing objective data about what can be expected in terms of effectiveness and how different factors impact that characteristic allows for rational and founded decisions to guide the development process, and escaping the somewhat intuitive nature of blockchain, since no standards are formed (as of yet). Another advantage is enabling architects and developers

to easily analyze different scenarios and chose the direction of development, which is best suited to the defined business aim and use case. Such clarity significantly helps the entire process, and reduces the possibility of scope creep, constantly changing direction, and overall uncertainty about a project. This in turn reduces the amount of resources required for task completion, and allows for better preparation. A significant advantage the system provides is being aware of where the biggest risks to effectiveness lie in a specific smart contract implementation. This way, developers can be cautious and consider potential protective measures before coding even begins. And last, but not least, business blockchain professionals can greatly benefit as well. The tool can significantly accelerate the process of finding the best solution to a problem, and can help communicate to the client what can be expected from a specific implementation, and how different decisions about the solution impact its effectiveness. A trade-off can then be easily identified, that satisfies both the client and the vendor.

3. IMPLEMENTATION OF IS AND RESULTS DISCUSSION

The system was developed according to the design specification and requirements laid out above and was tested with real data. The results from using it are discussed below.

Fig. 5. Preparing the input dataset in a Google spreadsheet file.

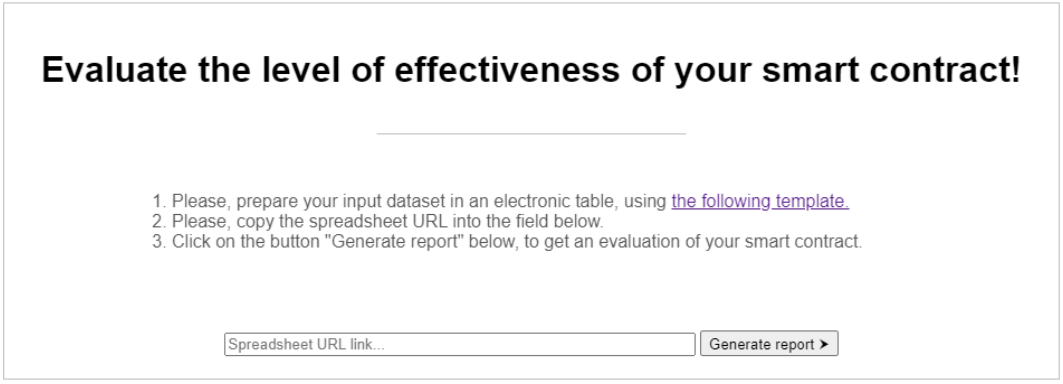


3.1. User Interface

The implementation consists of: 2 simple web pages - a data receiving page, and a results page; a spreadsheet template used to create the input dataset. An example with real data is visually presented below using screenshots of the system at each step. A user creates a new spreadsheet file and fills the input data as an array of pairs, as shown on Fig. 5. The URL of the file is then copied into the text field of the receiving page and the

generate report button is clicked, as shown on Fig. 6. The system then runs the data through the model and provides the output effectiveness metric on the results page, as shown on Fig. 7.

Fig. 6. The data receiving web page.

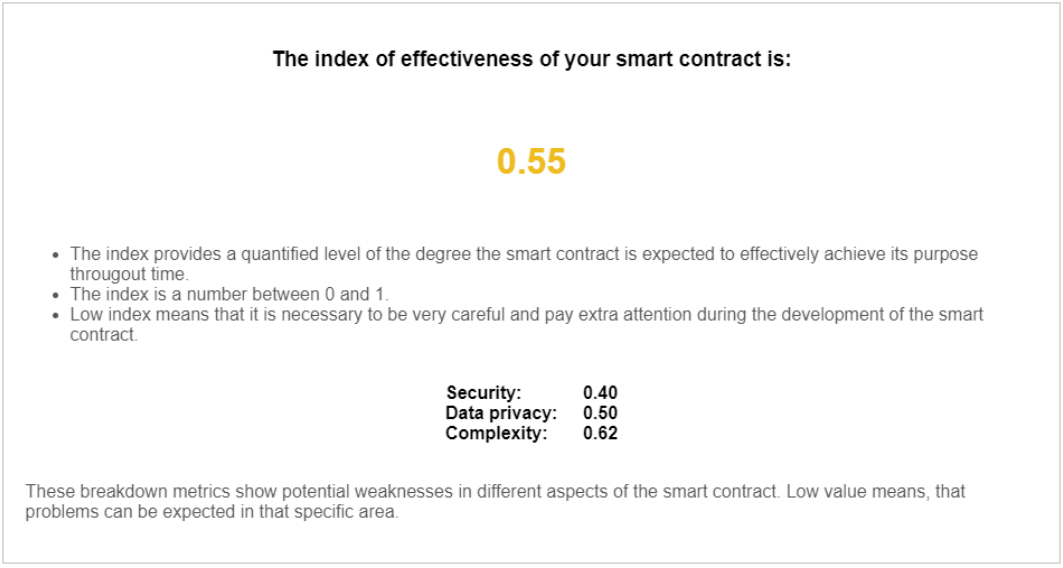


3.2. Lessons from developing the system

The created system performed as expected and achieved its purpose. The process of using it is easy and simple, the technologies allowed for quick and flexible development and the solution operates for free (within the requests volume constraint).

Downsides that were discovered include: not intuitive and difficult to read script documentation, a need for a more elegant graphic design suitable for multiple display sizes, and a better (and more user-friendly) way of presenting the output data - graphics may be better to perceive effectiveness and risk levels, instead of numbers and text.

Fig. 7. The contract effectiveness evaluation results web page



4. CONCLUSION AND FUTURE DEVELOPMENT

In the paper, the author has proposed an information system automating the evaluation of the effectiveness of blockchain smart contracts, using a model the author had previously created. The importance of blockchain and the advantages of automating smart contract evaluation were considered. Further, the design of the proposed system was explained in detail. The prototype and the feedback that informed system design was presented. Requirements that guide IS creation were given. The technological stack used to implement the solution was listed and explained. The architecture and the lifecycle process of using the application were shown. Advantages to this automated solution were discussed. The system was implemented and screenshots of the user interface were presented. Lessons learned were shared. Finally, direction for future system development and analysis is also provided.

More research is needed to expand and further advance the usefulness of the information system. Areas of improvement include: developing the system using more sophisticated framework to remove the constraints of the current one; creating an application programming interface (API) to allow the solution to be programmatically used remotely and turn it into a software-as-a-service (SaaS); collect data from multiple contract evaluations and analyzing it to: increase accuracy, investigate possible application in different subfields of blockchain development, advancing the model itself; implementing functionality for memorizing results for particular input datasets; automatic analysis for the best solution among different evaluations according to a specified priority criterion; and more.

References

1. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system. [online] Available at: https://bitcoin.org/bitcoin.pdf [Accessed January, 8th, 2023].

2. 5 Major Types of Blockchain Protocols | Analytics Steps, [online] Available at: https://www.analyticssteps.com/blogs/5-major-types-blockchain-protocols [Accessed January, 8th, 2023].

3. Blockchain Strategy | Shaping Europe's digital future (europa.eu), [online] Available at: https://digital-strategy.ec.europa.eu/en/policies/blockchain-strategy [Accessed January, 8th, 2023].

4. Mavridou, A., Laszka, A., Stachtiari, E., Dubey, A. (2019). VeriSolid: Correct-by-Design Smart Contracts for Ethereum. In: Goldberg, I., Moore, T. (eds) Financial Cryptography and Data security. FC 2019. Lecture Notes in Computer Science(), vol 11598. Springer, Cham. https://doi.org/10.1007/978-3-030-32101-7_27

5. Liu, Jing & Liu, Zhentian. (2019). A Survey on Security Verification of Blockchain Smart Contracts. IEEE Access. 7. 77894-77904. 10.1109/ACCESS.2019.2921624

6. A. Panayotov and P. Ruskov, "Measuring the effectiveness of blockchain smart contracts," 2022 International Conference Automatics and Informatics (ICAI), 2022, pp. 73-77, doi: 10.1109/ICAI55857.2022.9960013.

7. Apps Script – Google Apps Script, [online] Available at: https://www.google.com/script/start/ [Accessed January, 8th, 2023].