0.7

MDEX Process Quality Review

Score: 41%

Overview

This is a MDEX Process Quality Review completed on August 11th 2021. It was performed using the Process Review process (version 0.7.3) and is documented here. The review was performed by Nic of DeFiSafety. Check out our Telegram.

The final score of the review is **41%**, a **FAIL**. The breakdown of the scoring is in Scoring Appendix. For our purposes, a pass is **70%.**

Summary of the Process

Very simply, the review looks for the following declarations from the developer's site. With these declarations, it is reasonable to trust the smart contracts.

- **Here are my smart contracts on the blockchain**
- **Here is the documentation that explains what my smart contracts do**
- **Here are the tests I ran to verify my smart contract**
- **Here are the audit(s) performed on my code by third party experts**
- **Here are the admin controls and strategies**

Disclaimer

This report is for informational purposes only and does not constitute investment advice of any kind, nor does it constitute an offer to provide investment advisory or other services. Nothing in this report shall be considered a solicitation or offer to buy or sell any security, token, future, option or other financial instrument or to offer or provide any investment advice or service to any person in any jurisdiction. Nothing contained in this report constitutes investment advice or offers any opinion with respect to the suitability of any security, and the views expressed in this report should not be taken as advice to buy, sell or hold any security. The information in this report should not be relied upon for the purpose of investing. In preparing the information contained in this report, we have not taken into account the investment needs, objectives and financial circumstances of any particular investor. This information has no regard to the specific investment objectives, financial situation and particular needs of any specific recipient of this information and investments discussed may not be suitable for all investors.

Any views expressed in this report by us were prepared based upon the information available to us at the time such views were written. The views expressed within this report are limited to DeFiSafety and the author and do not reflect those of any additional or third party and are strictly based upon DeFiSafety, its authors, interpretations and evaluation of relevant data. Changed or additional information could cause such views to change. All information is subject to possible correction. Information may quickly become unreliable for various reasons, including changes in market conditions or economic circumstances.

Chain

This section indicates the blockchain used by this protocol.

> ✓ **Chain:** Binance Smart Chain

Guidance:

Ethereum
Binance Smart Chain
Polygon
Avalanche
Terra

Code and Team

This section looks at the code deployed on the Mainnet that gets reviewed and its corresponding software repository. The document explaining these questions is here. This review will answer the following questions:

1) Are the executing code addresses readily available? (%)
2) Is the code actively being used? (%)
3) Is there a public software repository? (Y/N)
4) Is there a development history visible? (%)
5) Is the team public (not anonymous)? (Y/N)

1) Are the executing code addresses readily available? (%)

> ✓ **Answer:** 100%

They are available at website https://github.com/mdexSwap/contracts, as indicated in the Appendix.

Guidance:

100% Clearly labelled and on website, docs or repo, quick to find
70% Clearly labelled and on website, docs or repo but takes a bit of looking
40% Addresses in mainnet.json, in discord or sub graph, etc

20%	Address found but labeling not clear or easy to find
0%	Executing addresses could not be found

2) Is the code actively being used? (%)

> ✓ **Answer:** 100%

Activity is over 10 transactions a day on contract *MdexFactory.sol*, as indicated in the Appendix.

Guidance:

100%	More than 10 transactions a day
70%	More than 10 transactions a week
40%	More than 10 transactions a month
10%	Less than 10 transactions a month
0%	No activity

3) Is there a public software repository? (Y/N)

> ✓ **Answer:** Yes

GitHub: https://github.com/mdexSwap/contracts.

Is there a public software repository with the code at a minimum, but also normally test and scripts. Even if the repository was created just to hold the files and has just 1 transaction, it gets a **"Yes"**. For teams with private repositories, this answer is **"No"**.

4) Is there a development history visible? (%)

> ! **Answer:** 30%

With 37 commits and 1 branch, this is a semi-healthy software repository.

This metric checks if the software repository demonstrates a strong steady history. This is normally demonstrated by commits, branches and releases in a software repository. A healthy history demonstrates a history of more than a month (at a minimum).

Guidance:

100%	Any one of 100+ commits, 10+branches
70%	Any one of 70+ commits, 7+branches
50%	Any one of 50+ commits, 5+branches

30%	Any one of 30+ commits, 3+branches
0%	Less than 2 branches or less than 30 commits

How to improve this score:

Continue to test and perform other verification activities after deployment, including routine maintenance updating to new releases of testing and deployment tools. A public development history indicates clearly to the public the level of continued investment and activity by the developers on the application. This gives a level of security and faith in the application.

5) Is the team public (not anonymous)? (Y/N)

> ⚠ **Answer:** No

No public team member info was found.

For a **"Yes"** in this question, the real names of some team members must be public on the website or other documentation (LinkedIn, etc). If the team is anonymous, then this question is a **"No"**.

Documentation

This section looks at the software documentation. The document explaining these questions is here.

Required questions are;

6) Is there a whitepaper? (Y/N)
7) Are the basic software functions documented? (Y/N)
8) Does the software function documentation fully (100%) cover the deployed contracts? (%)
9) Are there sufficiently detailed comments for all functions within the deployed contract code (%)
10) Is it possible to trace from software documentation to the implementation in
code (%)

6) Is there a whitepaper? (Y/N)

> ✓ **Answer:** Yes

Location: https://mdexdoc.gitbook.io/doc/.

7) Are the basic software functions documented? (Y/N)

> ⚠ **Answer:** No

No software functions are documented in the MDEX documentation. As of the time of this review, the API section of their documentation is empty.

How to improve this score:

Write the document based on the deployed code. For guidance, refer to the SecurEth System Description Document.

8) Does the software function documentation fully (100%) cover the deployed contracts? (%)

> **Answer:** No

No software functions are documented in the MDEX documentation. As of the time of this review, the API section of their documentation is empty.

Guidance:

100%   All contracts and functions documented
80%   Only the major functions documented
79-1%   Estimate of the level of software documentation
0%   No software documentation

How to improve this score:

This score can be improved by adding content to the software functions document such that it comprehensively covers the requirements. For guidance, refer to the SecurEth System Description Document. Using tools that aid traceability detection will help.

9) Are there sufficiently detailed comments for all functions within the deployed contract code (%)

> **Answer:** 42%

Code examples are in the Appendix. As per the SLOC, there is 42% commenting to code (CtC).

The Comments to Code (CtC) ratio is the primary metric for this score.

Guidance:

100%   CtC > 100   Useful comments consistently on all code
90-70%   CtC > 70 Useful comment on most code
60-20%   CtC > 20 Some useful commenting
0%   CtC < 20 No useful commenting

How to improve this score

This score can improve by adding comments to the deployed code such that it comprehensively covers the code. For guidance, refer to the SecurEth Software Requirements.

10) Is it possible to trace from software documentation to the implementation in code (%)

> ⚠ **Answer:** 0%

No software functions are documented in the MDEX documentation. As of the time of this review, the API section of their documentation is empty. Therefore, we cannot check the traceability between the documentation and the source code.

Guidance:

100% Clear explicit traceability between code and documentation at a requirement level for all code
60% Clear association between code and documents via non explicit traceability
40% Documentation lists all the functions and describes their functions
0% No connection between documentation and code

How to improve this score:

This score can improve by adding traceability from documentation to code such that it is clear where each outlined function is coded in the source code. For reference, check the SecurEth guidelines on traceability.

Testing

This section looks at the software testing available. It is explained in this document. This section answers the following questions;

11) Full test suite (Covers all the deployed code) (%)
12) Code coverage (Covers all the deployed lines of code, or explains misses) (%)
13) Scripts and instructions to run the tests (Y/N)
14) Report of the results (%)
15) Formal Verification test done (%)
16) Stress Testing environment (%)

11) Is there a Full test suite? (%)

> ⚠ **Answer:** 0%

The testing suite inside of the MDEX's core contracts' GitHub repository is empty. We therefore cannot evaluate the TtC.

Guidance:

100%	TtC > 120%	Both unit and system test visible
80%	TtC > 80%	Both unit and system test visible
40%	TtC < 80%	Some tests visible
0%		No tests obvious

How to improve this score:

This score can improved by adding tests to fully cover the code. Document what is covered by traceability or test results in the software repository.

12) Code coverage (Covers all the deployed lines of code, or explains misses) (%)

⚠ **Answer:** 0%

There is no evidence of any MDEX code coverage in their documentation or GitHub repositories.

Guidance:

100%	Documented full coverage
99-51%	Value of test coverage from documented results
50%	No indication of code coverage but clearly there is a reasonably complete set of tests
30%	Some tests evident but not complete
0%	No test for coverage seen

How to improve this score:

This score can improved by adding tests that achieve full code coverage. A clear report and scripts in the software repository will guarantee a high score.

13) Scripts and instructions to run the tests (Y/N)

⚠ **Answer:** No

There are no scripts or instructions to run the tests in MDEX's GitHub repository.

How to improve this score:

Add the scripts to the repository and ensure they work. Ask an outsider to create the environment and run the tests. Improve the scripts and docs based on their feedback.

14) Report of the results (%)

> ⚠ **Answer:** 0%

No test result report was found in the MDEX GitHub repository.

Guidance:

100%	Detailed test report as described below
70%	GitHub code coverage report visible
0%	No test report evident

How to improve this score

Add a report with the results. The test scripts should generate the report or elements of it.

15) Formal Verification test done (%)

> ⚠ **Answer:** 0%

No evidence of a MDEX Formal Verification test was found in any of their documentation or in further web research.

16) Stress Testing environment (%)

> ⚠ **Answer:** 0%

No evidence of MDEX's active testnet smart contract usage was found in their documentation.

Security

This section looks at the 3rd party software audits done. It is explained in this document. This section answers the following questions;

17) Did 3rd Party audits take place? (%)
18) Is the bounty value acceptably high?

17) Did 3rd Party audits take place? (%)

> ✓ **Answer:** 90%

Certik and SlowMist have both published MDEX audit reports after their v1 launch, which was on January 19th 2021.

Fairyproof has published a MDEX audit report before their BSC v2 launch, which was on April 8th 2021.

All audit reports can be found here.

Note: Most fix recommendations were successfully implemented by the MDEX team.

Guidance:

100% Multiple Audits performed before deployment and results public and
implemented or not required
90% Single audit performed before deployment and results public and implemented
or not required
70% Audit(s) performed after deployment and no changes required. Audit report is
public

50% Audit(s) performed after deployment and changes needed but not implemented
20% No audit performed
0% Audit Performed after deployment, existence is public, report is not public and
no improvements deployed OR smart contract address' not found, (where question 1 is 0%)

Deduct 25% if code is in a private repo and no note from auditors that audit is applicable to deployed code

18) Is the bounty value acceptably high (%)

> **Answer:** 50%

MDEX's Bug Bounty Program offers participating users with up to 60,000 MDX for the most critical of finds. As of the time of this review, this is just under 100k USD. However, the bounty rewards rises and falls with the price of the token, so this is subjective.

Guidance:

100% Bounty is 10% TVL or at least $1M AND active program (see below)
90% Bounty is 5% TVL or at least 500k AND active program
80% Bounty is 5% TVL or at least 500k
70% Bounty is 100k or over AND active program
60% Bounty is 100k or over
50% Bounty is 50k or over AND active program
40% Bounty is 50k or over
20% Bug bounty program bounty is less than 50k
0% No bug bounty program offered

An active program means that a third party (such as Immunefi) is actively driving hackers to the site. An inactive program would be static mentions on the docs.

Access Controls

This section covers the documentation of special access controls for a DeFi protocol. The admin access controls are the contracts that allow updating contracts or coefficients in the protocol. Since these contracts can allow the protocol admins to "change the rules", complete disclosure of capabilities is vital for user's transparency. It is explained in this document. The questions this section asks are as follow;

19) Can a user clearly and quickly find the status of the admin controls?
20) Is the information clear and complete?
21) Is the information in non-technical terms that pertain to the investments?
22) Is there Pause Control documentation including records of tests?

19) Can a user clearly and quickly find the status of the access controls (%)

> ⚠ **Answer:** 0%

Although there is a governance section in their litepaper, none of it describes the voting process or what the admins have/have no access to.

Guidance:

100%	Clearly labelled and on website, docs or repo, quick to find
70%	Clearly labelled and on website, docs or repo but takes a bit of looking
40%	Access control docs in multiple places and not well labelled
20%	Access control docs in multiple places and not labelled
0%	Admin Control information could not be found

20) Is the information clear and complete (%)

> ⚠ **Answer:** 0%

None of this information is explicitly stated in the MDEX documenation.

Guidance:

All the contracts are immutable -- 100% OR

a) All contracts are clearly labelled as upgradeable (or not) -- 30% AND
b) The type of ownership is clearly indicated (OnlyOwner / MultiSig / Defined Roles) -- 30% AND
c) The capabilities for change in the contracts are described -- 30%

How to improve this score:

Create a document that covers the items described above. An example is enclosed.

21) Is the information in non-technical terms that pertain to the investments (%)

⚠ **Answer:** 0%

Although there is a governance section in their litepaper, none of it describes the voting process or what the admins have/have no access to.

Guidance:

100% All the contracts are immutable
90% Description relates to investments safety and updates in clear, complete non-software l language
30% Description all in software specific language
0% No admin control information could not be found

How to improve this score:

Create a document that covers the items described above in plain language that investors can understand. An example is enclosed.

22) Is there Pause Control documentation including records of tests (%)

⚠ **Answer:** 0%

There is no evidence of Pause Control or a similar function in the MDEX documentation.

Guidance:

100% All the contracts are immutable or no pause control needed and this is explained OR
100% Pause control(s) are clearly documented and there is records of at least one test within 3 months

80% Pause control(s) explained clearly but no evidence of regular tests
40% Pause controls mentioned with no detail on capability or tests
0% Pause control not documented or explained

How to improve this score**:**

Create a document that covers the items described above in plain language that investors can understand. An example is enclosed.

Appendices

Author Details

The author of this review is Rex of DeFi Safety.

Email : rex@defisafety.com Twitter : @defisafety

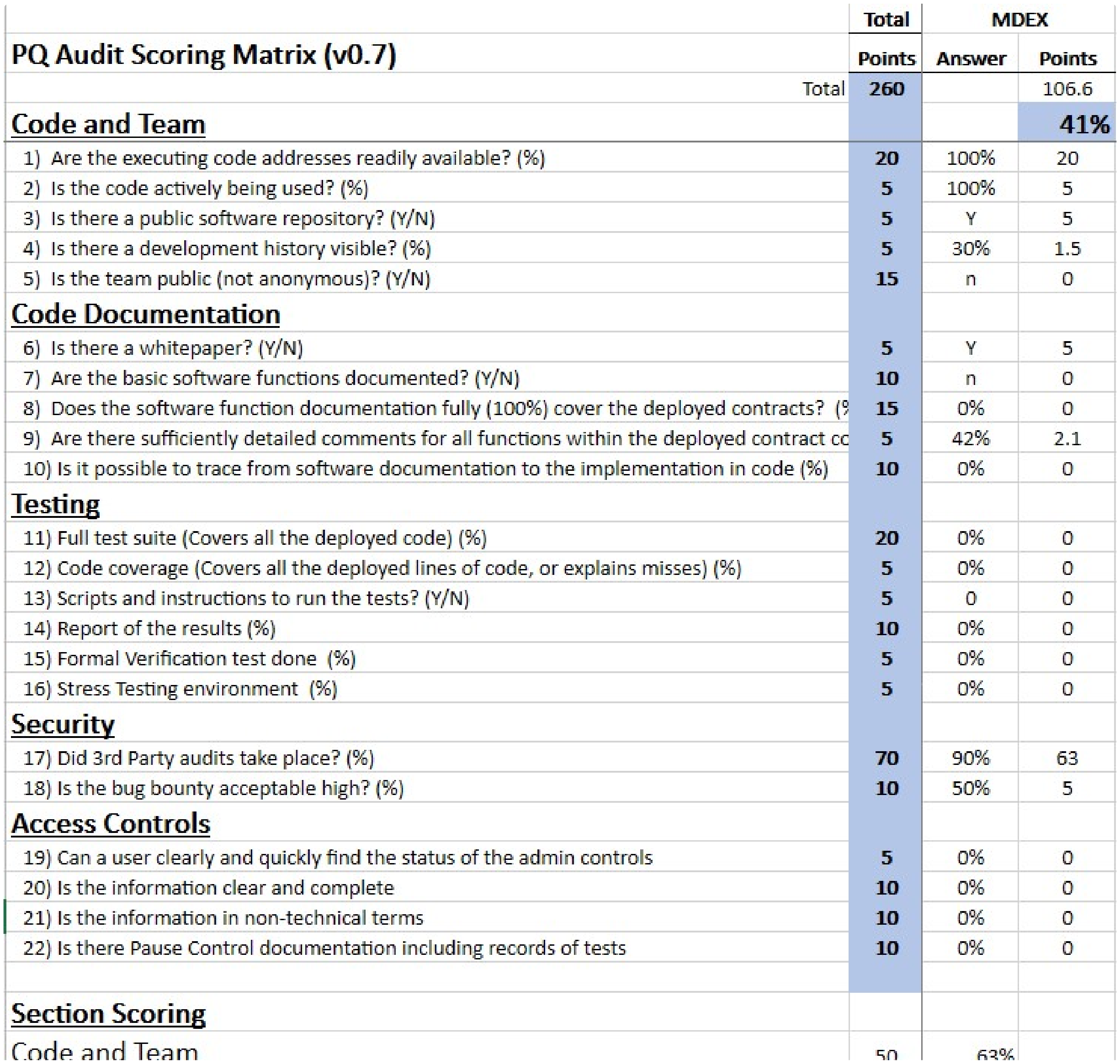
I started with Ethereum just before the DAO and that was a wonderful education. It showed the importance of code quality. The second Parity hack also showed the importance of good process. Here my aviation background offers some value. Aerospace knows how to make reliable code using quality processes.

I was coaxed to go to EthDenver 2018 and there I started SecuEth.org with Bryant and Roman. We created guidelines on good processes for blockchain code development. We got EthFoundation funding to assist in their development.

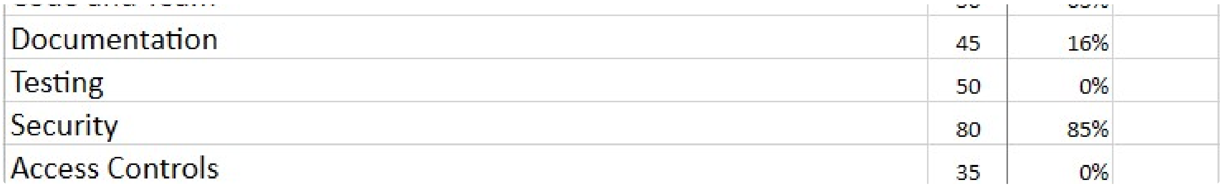
Process Quality Reviews are an extension of the SecurEth guidelines that will further increase the quality processes in Solidity and Vyper development.

DeFiSafety is my full time gig and we are working on funding vehicles for a permanent staff.

Scoring Appendix

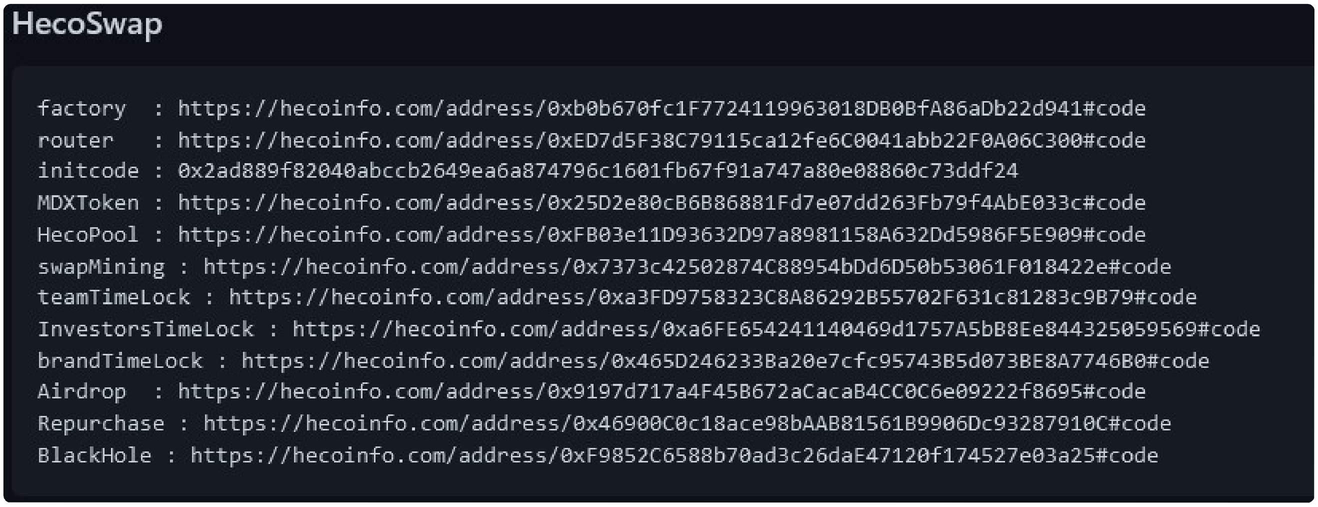


	Total	MDEX	
PQ Audit Scoring Matrix (v0.7)	Points	Answer	Points
Total	260		106.6
Code and Team			41%
1) Are the executing code addresses readily available? (%)	20	100%	20
2) Is the code actively being used? (%)	5	100%	5
3) Is there a public software repository? (Y/N)	5	Y	5
4) Is there a development history visible? (%)	5	30%	1.5
5) Is the team public (not anonymous)? (Y/N)	15	n	0
Code Documentation			
6) Is there a whitepaper? (Y/N)	5	Y	5
7) Are the basic software functions documented? (Y/N)	10	n	0
8) Does the software function documentation fully (100%) cover the deployed contracts? (%	15	0%	0
9) Are there sufficiently detailed comments for all functions within the deployed contract cc	5	42%	2.1
10) Is it possible to trace from software documentation to the implementation in code (%)	10	0%	0
Testing			
11) Full test suite (Covers all the deployed code) (%)	20	0%	0
12) Code coverage (Covers all the deployed lines of code, or explains misses) (%)	5	0%	0
13) Scripts and instructions to run the tests? (Y/N)	5	0	0
14) Report of the results (%)	10	0%	0
15) Formal Verification test done (%)	5	0%	0
16) Stress Testing environment (%)	5	0%	0
Security			
17) Did 3rd Party audits take place? (%)	70	90%	63
18) Is the bug bounty acceptable high? (%)	10	50%	5
Access Controls			
19) Can a user clearly and quickly find the status of the admin controls	5	0%	0
20) Is the information clear and complete	10	0%	0
21) Is the information in non-technical terms	10	0%	0
22) Is there Pause Control documentation including records of tests	10	0%	0
Section Scoring			
Code and Team	50	63%	



Documentation	45	16%	
Testing	50	0%	
Security	80	85%	
Access Controls	35	0%	

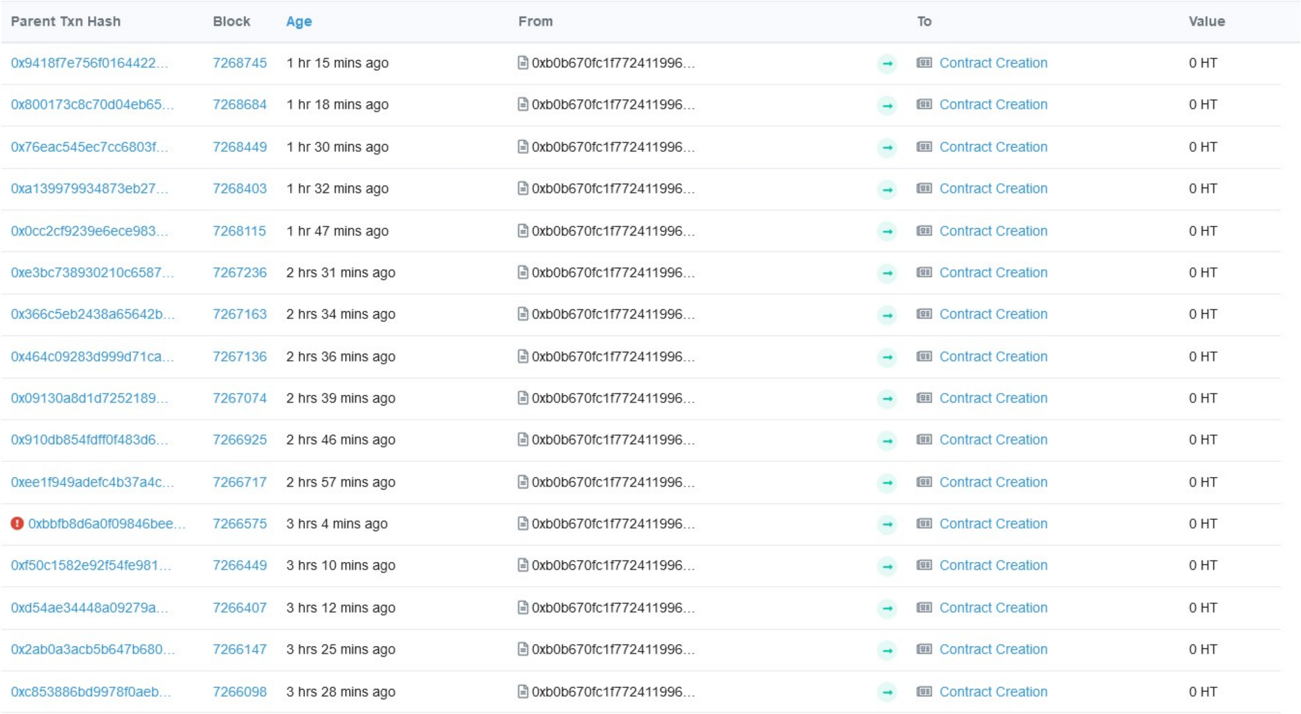
Executing Code Appendix



HecoSwap

```
factory  : https://hecoinfo.com/address/0xb0b670fc1F7724119963018DB0BfA86aDb22d941#code
router   : https://hecoinfo.com/address/0xED7d5F38C79115ca12fe6C0041abb22F0A06C300#code
initcode : 0x2ad889f82040abccb2649ea6a874796c1601fb67f91a747a80e08860c73ddf24
MDXToken : https://hecoinfo.com/address/0x25D2e80cB6B86881Fd7e07dd263Fb79f4AbE033c#code
HecoPool : https://hecoinfo.com/address/0xFB03e11D93632D97a8981158A632Dd5986F5E909#code
swapMining : https://hecoinfo.com/address/0x7373c42502874C88954bDd6D50b53061F018422e#code
teamTimeLock : https://hecoinfo.com/address/0xa3FD9758323C8A86292B55702F631c81283c9B79#code
InvestorsTimeLock : https://hecoinfo.com/address/0xa6FE654241140469d1757A5bB8Ee844325059569#code
brandTimeLock : https://hecoinfo.com/address/0x465D246233Ba20e7cfc95743B5d073BE8A7746B0#code
Airdrop  : https://hecoinfo.com/address/0x9197d717a4F45B672aCacaB4CC0C6e09222f8695#code
Repurchase : https://hecoinfo.com/address/0x46900C0c18ace98bAAB81561B9906Dc93287910C#code
BlackHole : https://hecoinfo.com/address/0xF9852C6588b70ad3c26daE47120f174527e03a25#code
```

Code Used Appendix



Parent Txn Hash	Block	Age	From		To	Value
0x9418f7e756f0164422...	7268745	1 hr 15 mins ago	0xb0b670fc1f772411996...	→	Contract Creation	0 HT
0x800173c8c70d04eb65...	7268684	1 hr 18 mins ago	0xb0b670fc1f772411996...	→	Contract Creation	0 HT
0x76eac545ec7cc6803f...	7268449	1 hr 30 mins ago	0xb0b670fc1f772411996...	→	Contract Creation	0 HT
0xa139979934873eb27...	7268403	1 hr 32 mins ago	0xb0b670fc1f772411996...	→	Contract Creation	0 HT
0x0cc2cf9239e6ece983...	7268115	1 hr 47 mins ago	0xb0b670fc1f772411996...	→	Contract Creation	0 HT
0xe3bc738930210c6587...	7267236	2 hrs 31 mins ago	0xb0b670fc1f772411996...	→	Contract Creation	0 HT
0x366c5eb2438a65642b...	7267163	2 hrs 34 mins ago	0xb0b670fc1f772411996...	→	Contract Creation	0 HT
0x464c09283d999d71ca...	7267136	2 hrs 36 mins ago	0xb0b670fc1f772411996...	→	Contract Creation	0 HT
0x09130a8d1d7252189...	7267074	2 hrs 39 mins ago	0xb0b670fc1f772411996...	→	Contract Creation	0 HT
0x910db854fdff0f483d6...	7266925	2 hrs 46 mins ago	0xb0b670fc1f772411996...	→	Contract Creation	0 HT
0xee1f949adefc4b37a4c...	7266717	2 hrs 57 mins ago	0xb0b670fc1f772411996...	→	Contract Creation	0 HT
❗ 0xbbfb8d6a0f09846bee...	7266575	3 hrs 4 mins ago	0xb0b670fc1f772411996...	→	Contract Creation	0 HT
0xf50c1582e92f54fe981...	7266449	3 hrs 10 mins ago	0xb0b670fc1f772411996...	→	Contract Creation	0 HT
0xd54ae34448a09279a...	7266407	3 hrs 12 mins ago	0xb0b670fc1f772411996...	→	Contract Creation	0 HT
0x2ab0a3acb5b647b680...	7266147	3 hrs 25 mins ago	0xb0b670fc1f772411996...	→	Contract Creation	0 HT
0xc853886bd9978f0aeb...	7266098	3 hrs 28 mins ago	0xb0b670fc1f772411996...	→	Contract Creation	0 HT

Example Code Appendix

```
contract CoinChef is Ownable {
    using SafeMath for uint256;
    using SafeERC20 for IERC20;

    using EnumerableSet for EnumerableSet.AddressSet;
    EnumerableSet.AddressSet private _sushiLP;

    // Info of each user.
    struct UserInfo {
        uint256 amount;     // How many LP tokens the user has provided.
        uint256 rewardDebt; // Reward debt.
        uint256 sushiRewardDebt; //sushi Reward debt.
    }

    // Info of each pool.
    struct PoolInfo {
        IERC20 lpToken;           // Address of LP token contract.
        uint256 allocPoint;       // How many allocation points assigned to this pool. MDX
        uint256 lastRewardBlock;  // Last block number that MDXs distribution occurs.
        uint256 accMdxPerShare; // Accumulated MDXs per share, times 1e12.
        uint256 totalAmount;    // Total amount of current pool deposit.
        uint256 accSushiPerShare; //Accumulated SuSHIs per share
    }

    // The MDX TOKEN!
    IMdx public mdx;
    // MDX tokens created per block.
    uint256 public constant mdxPerBlock = 100 ** 1e18;
    // Info of each pool.
    PoolInfo[] public poolInfo;
    // Info of each user that stakes LP tokens.
    mapping(uint256 => mapping(address => UserInfo)) public userInfo;
    // Corresponding to the pid of the sushi pool
    mapping(uint256 => uint256) public poolCorrespond;
    // Total allocation points. Must be the sum of all allocation points in all pools.
    uint256 public totalAllocPoint = 0;
    // The block number when MDX mining starts.
    uint256 public startBlock;
    // The block number when MDX mining end;
    uint256 public endBlock;
    // SUSHI MasterChef 0xc2EdaD668740f1aA35E4D8f227fB8E17dcA888Cd
    address public constant sushiChef = 0xc2EdaD668740f1aA35E4D8f227fB8E17dcA888Cd;
    // SUSHI Token 0x6B3595068778DD592e39A122f4f5a5cF09C90fE2
    address public constant sushiToken = 0x6B3595068778DD592e39A122f4f5a5cF09C90fE2;

    event Deposit(address indexed user, uint256 indexed pid, uint256 amount);
    event Withdraw(address indexed user, uint256 indexed pid, uint256 amount);
    event EmergencyWithdraw(address indexed user, uint256 indexed pid, uint256 amount);

    constructor(
        IMdx _mdx,
        uint256 _startBlock
    ) public {
        mdx = _mdx;
        startBlock = _startBlock;
        endBlock = _startBlock.add(200000);
    }

    function poolLength() public view returns (uint256) {
        return poolInfo.length;
    }

    function addSushiLP(address _addLP) public onlyOwner returns (bool) {
        require(_addLP != address(0), "LP is the zero address");
        IERC20(_addLP).approve(sushiChef, uint256(- 1));
        return EnumerableSet.add(_sushiLP, _addLP);
    }

    function isSushiLP(address _LP) public view returns (bool) {
        return EnumerableSet.contains(_sushiLP, _LP);
    }

    function getSushiLPLength() public view returns (uint256) {
        return EnumerableSet.length(_sushiLP);
    }

    function getSushiLPAddress(uint256 _pid) public view returns (address){
        require(_pid <= getSushiLPLength() - 1, "not find this SushiLP");
        return EnumerableSet.at(_sushiLP, _pid);
    }

    // Add a new lp to the pool. Can only be called by the owner.
    // XXX DO NOT add the same LP token more than once. Rewards will be messed up if you d
    function add(uint256 _allocPoint, IERC20 _lpToken, bool _withUpdate) public onlyOwner
        require(address(_lpToken) != address(0), "lpToken is the zero address");
        require(block.number < endBlock, "All token mining completed");
        if (_withUpdate) {
            massUpdatePools();
        }
        uint256 lastRewardBlock = block.number > startBlock ? block.number : startBlock;
        totalAllocPoint = totalAllocPoint.add(_allocPoint);
        poolInfo.push(PoolInfo({
        lpToken : _lpToken,
        allocPoint : _allocPoint,
        lastRewardBlock : lastRewardBlock,
        accMdxPerShare : 0,
        totalAmount : 0,
        accSushiPerShare : 0
        }));
    }

    // Update the given pool's MDX allocation point. Can only be called by the owner.
    function set(uint256 _pid, uint256 _allocPoint, bool _withUpdate) public onlyOwner {
        if (_withUpdate) {
            massUpdatePools();
        }
        totalAllocPoint = totalAllocPoint.sub(poolInfo[_pid].allocPoint).add(_allocPoint);
        poolInfo[_pid].allocPoint = _allocPoint;
    }

    // The current pool corresponds to the pid of the sushi pool
    function setPoolCorr(uint256 _pid, uint256 _sid) public onlyOwner {
        require(_pid <= poolLength() - 1, "not find this pool");
        poolCorrespond[_pid] = _sid;
    }

    // Update reward variables for all pools. Be careful of gas spending!
    function massUpdatePools() public {
        uint256 length = poolInfo.length;
        for (uint256 pid = 0; pid < length; ++pid) {
            updatePool(pid);
        }
    }

    // Update reward variables of the given pool to be up-to-date.
    function updatePool(uint256 _pid) public {
        PoolInfo storage pool = poolInfo[_pid];
        uint256 number = block.number > endBlock ? endBlock : block.number;
        if (number <= pool.lastRewardBlock) {
            return;
        }
        uint256 lpSupply;
        if (isSushiLP(address(pool.lpToken))) {
            if (pool.totalAmount == 0) {
                pool.lastRewardBlock = number;
                return;
            }
            lpSupply = pool.totalAmount;
        } else {
            lpSupply = pool.lpToken.balanceOf(address(this));
            if (lpSupply == 0) {
                pool.lastRewardBlock = number;
                return;
            }
        }

        uint256 multiplier = number.sub(pool.lastRewardBlock);
        uint256 mdxReward = multiplier.mul(mdxPerBlock).mul(pool.allocPoint).div(totalAllo
        bool minRet = mdx.mint(address(this), mdxReward);
        if (minRet) {
            pool.accMdxPerShare = pool.accMdxPerShare.add(mdxReward.mul(1e12).div(lpSupply
        }
        pool.lastRewardBlock = number;
    }

    // View function to see pending MDXs on frontend.
    function pending(uint256 _pid, address _user) external view returns (uint256, uint256)
        PoolInfo storage pool = poolInfo[_pid];
        if (isSushiLP(address(pool.lpToken))) {
            (uint256 mdxAmount, uint256 sushiAmount) = pendingMdxAndSushi(_pid, _user);
            return (mdxAmount, sushiAmount);
        } else {
            uint256 mdxAmount = pendingMdx(_pid, _user);
            return (mdxAmount, 0);
        }
    }

```

SLOC Appendix

Solidity Contracts

Language	Files	Lines	Blanks	Comments	Code	Complex
Solidity	21	12240	1827	3070	7343	911

Comments to Code 3070/7343 = 42%

Javascript Tests

Language	Files	Lines	Blanks	Comments	Code	Complex
JavaScript	N/A	N/A	N/A	N/A	N/A	N/A

Tests to Code = N/A