0.7

Convex Finance Process Quality Review

Score: 46%

Overview

This is a Process Quality Review of Convex Finance completed on May 26th 2021. It was performed using the Process Review process (version 0.7) and is documented here. The review was performed by Nic of DeFiSafety. Check out our Telegram.

The final score of the review is 46%, 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

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: Ethereum**

Guidance:
Ethereum
Binance

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 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://docs.convexfinance.com/convexfinance/faq/contract-addresses 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 labelling not clear or easy to find
0% Executing addresses could not be found

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

> ✓ Answer: 100%

Activity is 150 transactions a day on contract *Booster.sol*, as indicated in the Appendix.

Percentage Score 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/convex-eth

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

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

> ✓ Answer: 100%

With 112 commits and 9 branches, this is a healthy repository.

This 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 10 commits

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

> ⚠ Answer: No

No public team members were found on their various social media platforms.

For a yes in this question the real names of some team members must be public on the website or other documentation. If the team is anonymous and 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://docs.convexfinance.com/convexfinance/

How to improve this score

Ensure the white paper is available for download from your website or at least the software repository. Ideally update the whitepaper to meet the capabilities of your present application.

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

> ⚠ Answer: No

There is no evident software function documentation.

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: 0%

There is no evident software function documentation.

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 improve by adding content to the requirements 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: 0%

Code examples are in the Appendix. As per the SLOC, there is a 19% 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%

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 requirements to code such that it is clear where each requirement is coded. 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: 72%

With a TtC ratio of 72%, this is a decently full test suite.

This score is guided by the Test to Code ratio (TtC). Generally a good test to code ratio is over 100%. However the reviewers best judgement is the final deciding factor.

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 improve 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: 30%

No indication of code coverage but clearly there is a reasonably complete set of tests

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 improve by adding tests achieving 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

No scripts or test instructions are available in their software repo.

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%

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 formal verification of Convex Finance was found.

16) Stress Testing environment (%)

> ⚠ Answer: 0%

No stress test smart contract addresses were found.

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%

MixBytes did a Convex Finance audit on April 19th 2021. While the audit brought up some concerning aspects, all were fixed.

Convex Finance was released May 17th 2021

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
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, question

18) Is the bounty value acceptably high (%)

> ⚠ Answer: 0%

There is no evident bug bounty program.

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
50% Bounty is 100k or over
40% Bounty is 50k or over
20% Bug bounty program bounty is less than 50k
0% No bug bounty program offered

Active program means a third party actively driving hackers to the site. Inactive program would be static mention 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?
2`) 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 admin controls (%)

> ⚠ Answer: 20%

There is some admin control information present, but this information discusses the voting of the veCRV which is not relevant to Access Controls.

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%

a) No idea what is upgradeable - 0%
b) type of ownership is unclear - 0%
c) type of change not described 0%

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%

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%

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	Convex Finance	
PQ Audit Scoring Matrix (v0.7)	**Points**	**Answer**	**Points**
Total	**260**		119.9
Code and Team			**46%**
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**	100%	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 code (%)	**5**	0%	0
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**	72%	14.4
12) Code coverage (Covers all the deployed lines of code, or explains misses) (%)	**5**	30%	1.5
13) Scripts and instructions to run the tests? (Y/N)	**5**	N	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**	0%	0
Access Controls			
19) Can a user clearly and quickly find the status of the admin controls	**5**	20%	1
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	70%	
Documentation	45	11%	

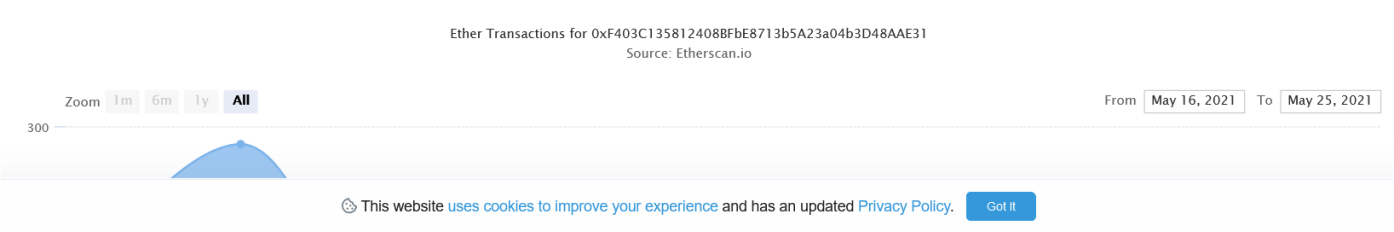
Testing	50	32%	
Security	80	79%	
Access Controls	35	3%	

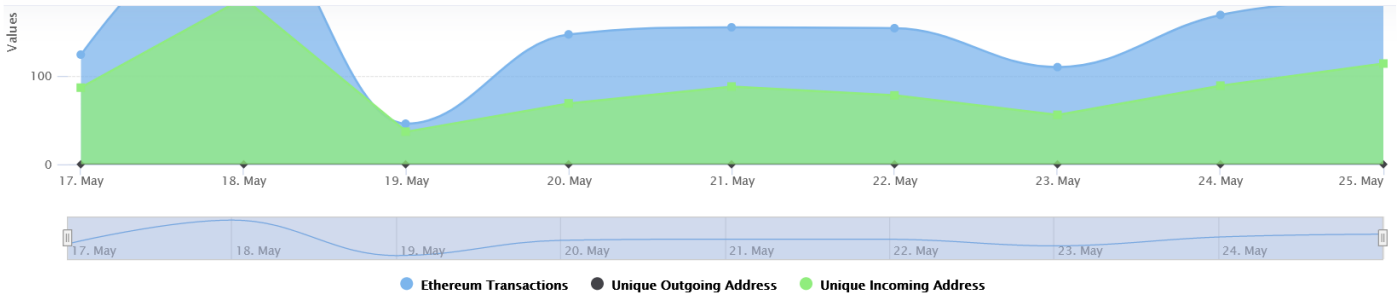
Executing Code Appendix

System Contracts

- Booster(main deposit contract): 0xF403C135812408BFbE8713b5A23a04b3D48A
- Voter Proxy(whitelist contract): 0x989AEb4d175e16225E39E87d0D97A3360524A
- CVX: 0x4e3FBD56CD56c3e72c1403e103b45Db9da5B9D2B
- cvxCRV: 0x62B9c7356A2Dc64a1969e19C23e4f579F9810Aa7
- CRV Depositor: 0x8014595F2AB54cD7c604B00E9fb932176fDc86Ae
- Reward Factory: 0xEdCCB35798fae4925718A43cc608aE136208aa8D
- Token Factory: 0x3c995e43E6ddD551E226F4c5544C77BfeD147aB9
- Stash Factory: 0x877288c4e6EbA4f635bA7428706447353B47De75
- CVX Rewards: 0xCF50b810E57Ac33B91dCF525C6ddd9881B139332
- cvxCRV Rewards: 0x3Fe65692bfCD0e6CF84cB1E7d24108E434A7587e
- Pool Manager: 0x3b2D30734491AD77880d31e8aFe2A4d8aC135a9C
- Arbitrator Vault: 0x25E12482a25CF36EC70fDA2A09C1ED077Fc21616
- Convex MasterChef: 0x5F465e9fcfFc217c5849906216581a657cd60605
- Vested Escrow: 0xe98984aD858075813AdA4261aF47e68A64E28fCC
- Airdrop Factory: 0xa1Bc2Cf69D474b39B91665e24E7f2606Ed142991
- Airdrop: 0x2E088A0A19dda628B4304301d1EA70b114e4AcCd
- Claim Zap: 0xAb9F4BB0aDD2CFbb168da95C590205419cD71f9B
- CVX/ETH SLP: 0x05767d9EF41dC40689678fFca0608878fb3dE906
- cvxCRV/CRV SLP: 0x33F6DDAEa2a8a54062E021873bCaEE006CdF4007
- Convex Multisig: 0xa3C5A1e09150B75ff251c1a7815A07182c3de2FB
- Convex Deployer: 0x947B7742C403f20e5FaCcDAc5E092C943E7D0277
- Treasury Vault: 0x1389388d01708118b497f59521f6943Be2541bb7

Code Used Appendix





Example Code Appendix

```
// SPDX-License-Identifier: MIT
pragma solidity 0.6.12;

import "./Interfaces.sol";
import '@openzeppelin/contracts/math/SafeMath.sol';
import '@openzeppelin/contracts/token/ERC20/IERC20.sol';
import '@openzeppelin/contracts/utils/Address.sol';
import '@openzeppelin/contracts/token/ERC20/SafeERC20.sol';


contract Booster{
    using SafeERC20 for IERC20;
    using Address for address;
    using SafeMath for uint256;

    address public constant crv = address(0xD533a949740bb3306d119CC777fa900bA034cd52);
    address public constant registry = address(0x0000000022D53366457F9d5E68Ec105046FC4383)
    uint256 public constant distributionAddressId = 4;
    address public constant voteOwnership = address(0xE478de485ad2fe566d49342Cbd03E49ed7DB
    address public constant voteParameter = address(0xBCfF8B0b9419b9A88c44546519b1e909cF33

    uint256 public lockIncentive = 1000; //incentive to crv stakers
    uint256 public stakerIncentive = 450; //incentive to native token stakers
    uint256 public earmarkIncentive = 50; //incentive to users who spend gas to make calls
    uint256 public platformFee = 0; //possible fee to build treasury
    uint256 public constant MaxFees = 2000;
    uint256 public constant FEE_DENOMINATOR = 10000;

    address public owner;
    address public feeManager;
    address public poolManager;
    address public immutable staker;
    address public immutable minter;
    address public rewardFactory;
    address public stashFactory;
    address public tokenFactory;
    address public rewardArbitrator;
    address public voteDelegate;
    address public treasury;
    address public stakerRewards; //cvx rewards

```

```
    address public lockRewards; //cvxCrv rewards(crv)
42  address public lockFees; //cvxCrv vecrv fees
43  address public feeDistro;
44  address public feeToken;
45
46  bool public isShutdown;
47
48  struct PoolInfo {
49      address lptoken;
50      address token;
51      address gauge;
52      address crvRewards;
53      address stash;
54      bool shutdown;
55  }
56
57  //index(pid) -> pool
58  PoolInfo[] public poolInfo;
59  mapping(address => bool) public gaugeMap;
60
```

SLOC Appendix

Solidity Contracts

Language	Files	Lines	Blanks	Comments	Code	Complex
Solidity	31	4196	703	554	2939	434

Comments to Code 554/2939 = 19%

Javascript Tests

Language	Files	Lines	Blanks	Comments	Code	Complex
JavaScript	18	2957	515	327	2115	44

Tests to Code 2115/2939 = 72%