0.7

Ampleforth Process Quality Review

Score: 87%

Overview

This is a Ampleforth Protocol Process Quality Review completed on July 26th 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 previous version of the review (0.5) is here.

The final score of the review is 87%, a strong pass. 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:** Ethereum

Guidance:

Ethereum
Binance Smart Chain
Polygon
Avalanche

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/ampleforth/uFragments, 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 *AdminUpgradeabilityProxy.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/ampleforth/uFragments.

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:** 100%

With 252 commits and 12 branches, this is a very 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

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

> **Answer:** Yes

Location: https://www.linkedin.com/company/ampleforth/.

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://drive.google.com/file/d/1I-NmSnQ6E7wY1nyouuf-GuDdJWNCnJWl/view.

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

> **Answer:** Yes

Ampleforth covers their basic software functions at https://drive.google.com/file/d/1I-NmSnQ6E7wY1nyouuf-GuDdJWNCnJWl/view.

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

Answer: 50%

Ampleforth covers the most major of their software functions at https://drive.google.com/file/d/1I-NmSnQ6E7wY1nyouuf-GuDdJWNCnJWl/view. In addition, the code commenting connects the software with the white paper. However neither are actually software documentation. Therefore a 50% score is given.

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: 60%

Code examples are in the Appendix. As per the SLOC, there is 62% commenting to code (CtC).

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

Note: The CtC was calculated using the "contracts" folders from the uFragments, market-oracle, Forth, and cross-chain-ample repositories. As we only evaluate the core code, files like interface, mocks, and any external third-party files were left out of our calculation.

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: 40%

There is a lack of traceability in the Ampleforth technical documentation as no functions are listed in the whitepaper. Commenting is quite good though so 40% and fills the need sfor docs a bit. Therefore 40%.

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: 100%

Code examples are in the Appendix. As per the SLOC, there is 472% testing to code (TtC).

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

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

> **Answer:** 99%

Ampleforth has a 96% codecov for their uFragments repository, 100% codecov for their oracle repository, and 100% codecov for their ampl-balancer repository.

(96%+100%+100%)/3 = 99%

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

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

> **Answer:** Yes

Scrips/Instructions location: Scripts can be found here, and instructions here.

14) Report of the results (%)

> **Answer:** 0%

All codecov reports in the Ampleforth GitHub are private and have their access denied to users.

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:** 100%

Ampleforth has had a Certik Formal Verification test done here.

16) Stress Testing environment (%)

> **Answer:** 100%

Evidence of Ampleforth's test-net smart contract usage can be found here.

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:** 100%

Ampleforth has had multiple audits pre and post-deployment of their multiple smart contracts. The full list of audit reports can be found here, and include reports from QuantStamp, Certik, Slowmist and Trail of Bits.

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, question

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:** 70%

Ampleforth has a Bug Bounty program with Immunefi that is live and rewards participating users with up to 100k in rewards.

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:** 100%

Ampleforth's community governance section can easily be found in their documentation here.

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:** 90%

a) Ampleforth clearly outlines the immutability of the governance contract and the upgradeability of the policy contracts.

b) The defined voting roles are clearly outline in governance documentation.

c) The scope of available upgrade propositions are all outline in governance documentation.

Sources: https://medium.com/ampleforth/state-of-discretion-and-governance-in-ampleforth-492963d84545, https://medium.com/ampleforth/the-state-of-discretion-and-governance-forth-ec2f710d2635, https://medium.com/ampleforth/ampl-forth-realizing-the-full-ecosystem-e2d88ca01691,

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%

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

> ✓ **Answer:** 90%

All descriptions pertaining governance information in the Ampleforth documentation are all detailed in user-friendly language.

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

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

> ⚠ **Answer:** 40%

According to the Ampleforth roadmap and GitHub commits, the pause function was removed in 2020.

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	Ampleforth	
PQ Audit Scoring Matrix (v0.7)	**Points**	**Answer**	**Points**
Total	**260**		227.45
Code and Team			**87%**
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	Y	15
Code Documentation			
6) Is there a whitepaper? (Y/N)	5	Y	5
7) Are the basic software functions documented? (Y/N)	10	Y	10
8) Does the software function documentation fully (100%) cover the deployed contracts? (%)	15	50%	7.5
9) Are there sufficiently detailed comments for all functions within the deployed contract code (%)	5	60%	3
10) Is it possible to trace from software documentation to the implementation in code (%)	10	40%	4
Testing			
11) Full test suite (Covers all the deployed code) (%)	20	100%	20
12) Code coverage (Covers all the deployed lines of code, or explains misses) (%)	5	99%	4.95
13) Scripts and instructions to run the tests? (Y/N)	5	Y	5
14) Report of the results (%)	10	0%	0
15) Formal Verification test done (%)	5	100%	5
16) Stress Testing environment (%)	5	100%	5
Security			
17) Did 3rd Party audits take place? (%)	70	100%	70
18) Is the bug bounty acceptable high? (%)	10	70%	7
Access Controls			
19) Can a user clearly and quickly find the status of the admin controls	5	100%	5
20) Is the information clear and complete	10	90%	9
21) Is the information in non-technical terms	10	90%	9
22) Is there Pause Control documentation including records of tests	10	80%	8
Section Scoring			
Code and Team	50	100%	
Documentation	45	66%	
Testing	50	80%	
Security	80	96%	
Access Controls	35	89%	

Executing Code Appendix

The official mainnet addresses are:

- ERC-20 Token: 0xD46bA6D942050d489DBd938a2C909A5d5039A161
- Supply Policy: 0x1B228a749077b8e307C5856cE62Ef35d96Dca2ea
- Orchestrator: 0x6fb00a180781e75f87e2b690af0196baa77c7e7c
- Market Oracle: 0x99c9775e076fdf99388c029550155032ba2d8914
- CPI Oracle: 0xa759f960dd59a1ad32c995ecabe802a0c35f244f

Code Used Appendix

Advanced Latest 25 internal transaction

Parent Txn Hash	Block	Age
0x6d3b6258d574fd81bff...	12899045	19 hrs 8 mins ago
0x6d3b6258d574fd81bff...	12899045	19 hrs 8 mins ago

0x6d3b6258d574fd81bff...	12899045	19 hrs 8 mins ago
0x6d3b6258d574fd81bff...	12899045	19 hrs 8 mins ago
0x6d3b6258d574fd81bff...	12899045	19 hrs 8 mins ago
0x6d3b6258d574fd81bff...	12899045	19 hrs 8 mins ago
0x6d3b6258d574fd81bff...	12899045	19 hrs 8 mins ago
0xc9261badbe97fe4926...	12899042	19 hrs 8 mins ago
0xc9261badbe97fe4926...	12899042	19 hrs 8 mins ago
0xc9261badbe97fe4926...	12899042	19 hrs 8 mins ago
0xc9261badbe97fe4926...	12899042	19 hrs 8 mins ago
0xc9261badbe97fe4926...	12899042	19 hrs 8 mins ago
0xc9261badbe97fe4926...	12899042	19 hrs 8 mins ago
0xc9261badbe97fe4926...	12899042	19 hrs 8 mins ago

Example Code Appendix

```
/**
 * @title Orchestrator
 * @notice The orchestrator is the main entry point for rebase operations. It coordinates
 * actions with external consumers.
 */
contract Orchestrator is Ownable {
    struct Transaction {
        bool enabled;
        address destination;
        bytes data;
    }

    // Stable ordering is not guaranteed.
    Transaction[] public transactions;

    UFragmentsPolicy public policy;

    /**
     * @param policy_ Address of the UFragments policy.
     */
    constructor(address policy_) public {
        Ownable.initialize(msg.sender);
        policy = UFragmentsPolicy(policy_);
    }

```

```
    /**
     * @notice Main entry point to initiate a rebase operation.
     *         The Orchestrator calls rebase on the policy and notifies downstream applica
     *         Contracts are guarded from calling, to avoid flash loan attacks on liquidity
     *         providers.
     *         If a transaction in the transaction list fails, Orchestrator will stop execu
     *         and revert to prevent a gas underprice attack.
     */
    function rebase() external {
        require(msg.sender == tx.origin); // solhint-disable-line avoid-tx-origin

        policy.rebase();

        for (uint256 i = 0; i < transactions.length; i++) {
            Transaction storage t = transactions[i];
            if (t.enabled) {
                (bool result, ) = t.destination.call(t.data);
                if (!result) {
                    revert("Transaction Failed");
                }
            }
        }
    }

    /**
     * @notice Adds a transaction that gets called for a downstream receiver of rebases
     * @param destination Address of contract destination
     * @param data Transaction data payload
     */
    function addTransaction(address destination, bytes memory data) external onlyOwner {
        transactions.push(Transaction({enabled: true, destination: destination, data: data
    }

    /**
     * @param index Index of transaction to remove.
     *              Transaction ordering may have changed since adding.
     */
    function removeTransaction(uint256 index) external onlyOwner {
        require(index < transactions.length, "index out of bounds");

        if (index < transactions.length - 1) {
            transactions[index] = transactions[transactions.length - 1];
        }

        transactions.pop();
    }

    /**
     * @param index Index of transaction. Transaction ordering may have changed since addi
     * @param enabled True for enabled, false for disabled.
     */
    function setTransactionEnabled(uint256 index, bool enabled) external onlyOwner {
        require(index < transactions.length, "index must be in range of stored tx list");
```

```
79          transactions[index].enabled = enabled;
80      }
81
82      /**
83       * @return Number of transactions, both enabled and disabled, in transactions list.
84       */
85      function transactionsSize() external view returns (uint256) {
86          return transactions.length;
87      }
88  }
```

SLOC Appendix

Solidity Contracts

Language	Files	Lines	Blanks	Comments	Code	Complex
Solidity	17	3332	507	1078	1747	177

Comments to Code 1078/1747 = 62%

Javascript Tests

Language	Files	Lines	Blanks	Comments	Code	Complex
JavaScript	16	5815	616	76	5123	15
TypeScript	12	3607	406	72	3129	19

Tests to Code 8252/1747 = 472%