0.7

benQi Process Quality Review

Score: 50

Overview

This is a benQi Process Quality Review completed on 04/11/2021. It was performed using the Process Review process (version 0.7.3) and is documented here. The review was performed by Nick of DeFiSafety. Check out our Telegram.

The final score of the review is **50%,** 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 blockchains used by this protocol. This report covers all of the blockchains upon which the protocol is deployed.

> **Chain:** Avalanche

Guidance:

Ethereum
Binance Smart Chain
Polygon
Avalanche
Terra
Celo
Arbitrum
Solana

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://docs.benqi.fi/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 significantly more than 10 transactions a day on contract *Comptroller*, 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/Benqi-fi

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:** 0%

At 4 commits and 1 branch, benQi's development history is not as expansive as its TVL necessitates.

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:** Yes

Location: https://www.linkedin.com/company/benqi/people/

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://docs.benqi.fi

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

> ⚠ **Answer:** No

No software functions are detailed.

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%

No software function is present.

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:** 63%

Code examples are in the Appendix. As per the SLOC, there is 63% 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%

There is no documentation of code function aside from identification without elaboration.

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%

No evidence of testing was found.

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 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 testing on benQi.

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

Neither scripts nor instructions were found.

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 report was found.

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 was conducted.

16) Stress Testing environment (%)

> ⚠ **Answer:** 0%

No evidence of stress testing was 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%

One audit was conducted before deployment, though it is unclear if the findings were implemented.

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:** 0%

There is no bug bounty offered by benQi.

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:** 40%

A brief mention of access controls was identified under governance, but there is no explicit mention of them. While the protocol is "decentralising", it is unclear where it is down this path. This is unclear information.

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:** 23%

a) All contracts are clearly labelled as upgradeable (or not) -- 0% -- no contracts are identified as upgradeable.
b) The type of ownership is clearly indicated (OnlyOwner / MultiSig / Defined Roles) -- 3% -- it is unclear who owns the contracts
c) The capabilities for change in the contracts are described -- 20% -- some capabilities for change are identified.

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:** 90%

This information is very clear.

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%

No pause control was documented.

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	benQi	
PQ Audit Scoring Matrix (v0.7)	**Points**	**Answer**	**Points**
Total	**260**		129.45
Code and Team			**50%**
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**	0%	0
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**	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**	63%	3.15
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**	70%	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**	40%	2
20) Is the information clear and complete	**10**	23%	2.3
21) Is the information in non-technical terms	**10**	90%	9
22) Is there Pause Control documentation including records of tests	**10**	0%	0
Section Scoring			
Code and Team	50	90%	
Documentation	45	18%	
Testing	50	0%	
Security	80	79%	

Access Controls	35	38%

Executing Code Appendix

BENQI token (QI)

```
1 0x8729438EB15e2C8B576fCc6AeCdA6A148776C0F5
```

Comptroller

```
1 0x486Af39519B4Dc9a7fCcd318217352830E8AD9b4
```

QiToken

qiAVAX

```
1 0x5C0401e81Bc07Ca70fAD469b451682c0d747Ef1c
```

qiBTC

```
1 0xe194c4c5aC32a3C9ffDb358d9Bfd523a0B6d1568
```

Code Used Appendix

RECEIVED←	0x0c3e75...9e8cb076	FROM TO	0xd46f6f42aa0b59895946035d333143FB1Df3F682 0x486Af39519B4Dc9a7fCcd318217352830E8AD9b4	9 minutes ago Wed, 03 Nov 2021 00:59:25 GMT	0.00 AVAX	✓
RECEIVED←	0x3d4eac...3805f474	FROM TO	0x549D581206d0070b15365cCD15f8FD11c1311F8c 0x486Af39519B4Dc9a7fCcd318217352830E8AD9b4	10 minutes ago Wed, 03 Nov 2021 00:57:56 GMT	0.00 AVAX	✓
RECEIVED←	0x5a8b2e...66a3b3bc	FROM TO	0x549D581206d0070b15365cCD15f8FD11c1311F8c 0x486Af39519B4Dc9a7fCcd318217352830E8AD9b4	11 minutes ago Wed, 03 Nov 2021 00:57:38 GMT	0.00 AVAX	✓
RECEIVED←	0xfd16d3...bf8a2cdf	FROM TO	0x273722eB3012E78d13E17353145F98964e5795Ed 0x486Af39519B4Dc9a7fCcd318217352830E8AD9b4	14 minutes ago Wed, 03 Nov 2021 00:54:01 GMT	0.00 AVAX	✓
RECEIVED←	0xd9ec3d...6d0e8d5e	FROM TO	0x273722eB3012E78d13E17353145F98964e5795Ed 0x486Af39519B4Dc9a7fCcd318217352830E8AD9b4	14 minutes ago Wed, 03 Nov 2021 00:53:56 GMT	0.00 AVAX	✓
RECEIVED←	0x042f6d...28039b1a	FROM TO	0x6013A56FbE216fE0FeD0f4b2741bDC22f1143d3c 0x486Af39519B4Dc9a7fCcd318217352830E8AD9b4	15 minutes ago Wed, 03 Nov 2021 00:53:42 GMT	0.00 AVAX	✓

RECEIVED←	0x1ea5ba...74dcdb30	FROM 0x6013A56FbE216fE0FeD0f4b2741bDC22f1143d3c TO 0x486Af39519B4Dc9a7fCcd318217352830E8AD9b4	16 minutes ago Wed, 03 Nov 2021 00:52:52 GMT	0.00 AVAX	✓
RECEIVED←	0x158fc5...d04af55b	FROM 0x21dF3F5e4FE5C7F8F731fd6B1736ca08ABb3Da32 TO 0x486Af39519B4Dc9a7fCcd318217352830E8AD9b4	18 minutes ago Wed, 03 Nov 2021 00:49:58 GMT	0.00 AVAX	✓
RECEIVED←	0xa81859...473e8ddf	FROM 0x21dF3F5e4FE5C7F8F731fd6B1736ca08ABb3Da32 TO 0x486Af39519B4Dc9a7fCcd318217352830E8AD9b4	19 minutes ago Wed, 03 Nov 2021 00:49:52 GMT	0.00 AVAX	✓
RECEIVED←	0x6fcad6...89baacf6	FROM 0x068D1a6030BB3e8cf185b0433Dd3c48C6D3157b8 TO 0x486Af39519B4Dc9a7fCcd318217352830E8AD9b4	22 minutes ago Wed, 03 Nov 2021 00:46:24 GMT	0.00 AVAX	✓
RECEIVED←	0xf2e4cb...cc8d4404	FROM 0xF3cFa8F7C6FADa90Bb35AcE399d2AC98f742d150 TO 0x486Af39519B4Dc9a7fCcd318217352830E8AD9b4	26 minutes ago Wed, 03 Nov 2021 00:42:31 GMT	0.00 AVAX	✓

Example Code Appendix

```
contract Comptroller is ComptrollerVXStorage, ComptrollerInterface, ComptrollerErrorReporte
    /// @notice Emitted when an admin supports a market
    event MarketListed(QiToken qiToken);

    /// @notice Emitted when an account enters a market
    event MarketEntered(QiToken qiToken, address account);

    /// @notice Emitted when an account exits a market
    event MarketExited(QiToken qiToken, address account);

    /// @notice Emitted when close factor is changed by admin
    event NewCloseFactor(uint oldCloseFactorMantissa, uint newCloseFactorMantissa);

    /// @notice Emitted when a collateral factor is changed by admin
    event NewCollateralFactor(QiToken qiToken, uint oldCollateralFactorMantissa, uint newC

    /// @notice Emitted when liquidation incentive is changed by admin
    event NewLiquidationIncentive(uint oldLiquidationIncentiveMantissa, uint newLiquidatio

    /// @notice Emitted when price oracle is changed
    event NewPriceOracle(PriceOracle oldPriceOracle, PriceOracle newPriceOracle);

    /// @notice Emitted when pause guardian is changed
    event NewPauseGuardian(address oldPauseGuardian, address newPauseGuardian);

    /// @notice Emitted when an action is paused globally
    event ActionPaused(string action, bool pauseState);

    /// @notice Emitted when an action is paused on a market
    event ActionPaused(QiToken qiToken, string action, bool pauseState);

    /// @notice Emitted when a new BENQI or AVAX speed is calculated for a market
    event SpeedUpdated(uint8 tokenType, QiToken indexed qiToken, uint newSpeed);

    /// @notice Emitted when a new BENQI speed is set for a contributor
    event ContributorQiSpeedUpdated(address indexed contributor, uint newSpeed);

    /// @notice Emitted when BENQI or AVAX is distributed to a borrower
    event DistributedBorrowerReward(uint8 indexed tokenType, QiToken indexed qiToken, addr
```

```
40
41      /// @notice Emitted when BENQI or AVAX is distributed to a supplier
42      event DistributedSupplierReward(uint8 indexed tokenType, QiToken indexed qiToken, addr
43
44      /// @notice Emitted when borrow cap for a qiToken is changed
45      event NewBorrowCap(QiToken indexed qiToken, uint newBorrowCap);
46
47      /// @notice Emitted when borrow cap guardian is changed
48      event NewBorrowCapGuardian(address oldBorrowCapGuardian, address newBorrowCapGuardian)
49
50      /// @notice Emitted when BENQI is granted by admin
51      event QiGranted(address recipient, uint amount);
52
53      /// @notice The initial BENQI and AVAX index for a market
54      uint224 public constant initialIndexConstant = 1e36;
55
56      // closeFactorMantissa must be strictly greater than this value
57      uint internal constant closeFactorMinMantissa = 0.05e18; // 0.05
58
59      // closeFactorMantissa must not exceed this value
60      uint internal constant closeFactorMaxMantissa = 0.9e18; // 0.9
61
62      // No collateralFactorMantissa may exceed this value
63      uint internal constant collateralFactorMaxMantissa = 0.9e18; // 0.9
64
65      // reward token type to show BENQI or AVAX
66      uint8 public constant rewardQi = 0;
67      uint8 public constant rewardAvax = 1;
68
69      constructor() public {
70          admin = msg.sender;
71      }
72
73      /*** Assets You Are In ***/
74
75      /**
76       * @notice Returns the assets an account has entered
77       * @param account The address of the account to pull assets for
78       * @return A dynamic list with the assets the account has entered
79       */
80      function getAssetsIn(address account) external view returns (QiToken[] memory) {
81          QiToken[] memory assetsIn = accountAssets[account];
82
83          return assetsIn;
84      }
85
86      /**
87       * @notice Returns whether the given account is entered in the given asset
88       * @param account The address of the account to check
89       * @param qiToken The qiToken to check
90       * @return True if the account is in the asset, otherwise false.
91       */
92      function checkMembership(address account, QiToken qiToken) external view returns (bool
```

```
93          return markets[address(qiToken)].accountMembership[account];
94      }
95
96      /**
97       * @notice Add assets to be included in account liquidity calculation
98       * @param qiTokens The list of addresses of the qiToken markets to be enabled
99       * @return Success indicator for whether each corresponding market was entered
100      */
101     function enterMarkets(address[] memory qiTokens) public returns (uint[] memory) {
102         uint len = qiTokens.length;
103
104         uint[] memory results = new uint[](len);
105         for (uint i = 0; i < len; i++) {
106             QiToken qiToken = QiToken(qiTokens[i]);
107
108             results[i] = uint(addToMarketInternal(qiToken, msg.sender));
109         }
110
111         return results;
112     }
113
114     /**
115      * @notice Add the market to the borrower's "assets in" for liquidity calculations
116      * @param qiToken The market to enter
117      * @param borrower The address of the account to modify
118      * @return Success indicator for whether the market was entered
119      */
120     function addToMarketInternal(QiToken qiToken, address borrower) internal returns (Erro
121         Market storage marketToJoin = markets[address(qiToken)];
122
123         if (!marketToJoin.isListed) {
124             // market is not listed, cannot join
125             return Error.MARKET_NOT_LISTED;
126         }
127
128         if (marketToJoin.accountMembership[borrower] == true) {
129             // already joined
130             return Error.NO_ERROR;
131         }
132
133         // survived the gauntlet, add to list
134         // NOTE: we store these somewhat redundantly as a significant optimization
135         //  this avoids having to iterate through the list for the most common use cases
136         //  that is, only when we need to perform liquidity checks
137         //  and not whenever we want to check if an account is in a particular market
138         marketToJoin.accountMembership[borrower] = true;
139         accountAssets[borrower].push(qiToken);
140
141         emit MarketEntered(qiToken, borrower);
142
143         return Error.NO_ERROR;
144     }
145
```

```
146
147     /**
148      * @notice Removes asset from sender's account liquidity calculation
149      * @dev Sender must not have an outstanding borrow balance in the asset,
150      *  or be providing necessary collateral for an outstanding borrow.
151      * @param qiTokenAddress The address of the asset to be removed
152      * @return Whether or not the account successfully exited the market
153      */
154     function exitMarket(address qiTokenAddress) external returns (uint) {
155         QiToken qiToken = QiToken(qiTokenAddress);
156         /* Get sender tokensHeld and amountOwed underlying from the qiToken */
157         (uint oErr, uint tokensHeld, uint amountOwed, ) = qiToken.getAccountSnapshot(msg.se
158         require(oErr == 0, "exitMarket: getAccountSnapshot failed"); // semi-opaque error c
159
160         /* Fail if the sender has a borrow balance */
161         if (amountOwed != 0) {
162             return fail(Error.NONZERO_BORROW_BALANCE, FailureInfo.EXIT_MARKET_BALANCE_OWED)
163         }
164
165         /* Fail if the sender is not permitted to redeem all of their tokens */
166         uint allowed = redeemAllowedInternal(qiTokenAddress, msg.sender, tokensHeld);
167         if (allowed != 0) {
168             return failOpaque(Error.REJECTION, FailureInfo.EXIT_MARKET_REJECTION, allowed)
169         }
170
171         Market storage marketToExit = markets[address(qiToken)];
172
173         /* Return true if the sender is not already ‘in’ the market */
174         if (!marketToExit.accountMembership[msg.sender]) {
175             return uint(Error.NO_ERROR);
176         }
177
178         /* Set qiToken account membership to false */
179         delete marketToExit.accountMembership[msg.sender];
180
181         /* Delete qiToken from the account’s list of assets */
182         // load into memory for faster iteration
183         QiToken[] memory userAssetList = accountAssets[msg.sender];
184         uint len = userAssetList.length;
185         uint assetIndex = len;
186         for (uint i = 0; i < len; i++) {
187             if (userAssetList[i] == qiToken) {
188                 assetIndex = i;
189                 break;
190             }
191         }
192
193         // We *must* have found the asset in the list or our redundant data structure is b
194         assert(assetIndex < len);
195
196         // copy last item in list to location of item to be removed, reduce length by 1
197         QiToken[] storage storedList = accountAssets[msg.sender];
198         storedList[assetIndex] = storedList[storedList.length - 1];
```

```
198         storedList.length--;
199
200         emit MarketExited(qiToken, msg.sender);
201
202         return uint(Error.NO_ERROR);
203     }
```

SLOC Appendix

Solidity Contracts

Language	Files	Lines	Blanks	Comments	Code	Complex
Solidity	19	3457	541	1132	1784	301

Comments to Code 1132/1784 = 63%

Javascript Tests

Language	Files	Lines	Blanks	Comments	Code	Complex
JavaScript	0	0	0	0	0	0

Tests to Code = N/A