

0.7

GrowthDefi Process Quality Review

Score: 57%

Overview

This is a Process Quality Review of [Growth Defi](#) completed on May 10th 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 57%, 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.

This completed report is copyright (c) DeFiSafety 2021. Permission is given to copy in whole, retaining this copyright label.

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%**

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 |

They are available at website <https://github.com/GrowthDeFi/growthdefi-v1-core> as indicated in the [Appendix](#).

How to improve this score

Make the Ethereum addresses of the smart contract utilized by your application available on either your website or your GitHub (in the README for instance). Ensure the addresses is up to date. This is a very important question wrt to the final score.

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

 Answer: 70%

Activity is 6 transactions a day on contract stkGRO.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/GrowthDeFi>

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 priva, ±Gte repos, this answer is No.

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

 Answer: 100%

With 122 commits and 2 branches they have a robust development history.

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

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

<https://growthdefi.com/#home>

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://growthdefi.com/pdf/litepaper.pdf>

Location: <https://growthdefi.com/>

The link to their Litepaper exists, but does not load anything.

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

 Answer: No

There are no basic software function documentation that is easily visible on their website or GitHub repo.

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 documentation found.

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: 100%

Code examples are in the [Appendix](#). As per the [SLOC](#), there is 102% 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

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

 Answer: 0%

No software documentation was found, and therefore no traceability with their 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 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: 40%

With a **TtC** of 36%, there is not a very good 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: 0%

No code coverage on GitHub or in their [Consensys Dilligence audit](#).

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: Yes

Instructions to run test can be found in their [GitHub](#).

14) Report of the results (%)

 Answer: 0%

There is a testing suite in their [GitHub](#), but no test reports.

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 GrowthDefi can be found on the web.

16) Stress Testing environment (%)

 Answer: 100%

Multiple Kovan addresses with multiple transactions can be found in their [GitHub](#).

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: 70%

An [audit](#) was performed by Consensys Diligence in December 2020 which is after GrowthDefi's launch in August 2020. Audit is public.

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: 70%

Bug Bounty Location: <https://growthdefi.medium.com/gtokens-bug-bounty-up-to-100-000-caa914fb765b>

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?
- 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 admin controls (%)

 Answer: 70%

There was some admin control information found in their [FAQ](#).

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: 30%

The type of ownership is clearly indicated (OnlyOwner / MultiSig / Defined Roles)

Guidance:

All the contracts are immutable -- 100% OR

All contracts are clearly labelled as upgradeable (or not) -- 30% AND


The type of ownership is clearly indicated (OnlyOwner / MultiSig / Defined Roles) -- 30% AND

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%

The information is clear, non-software language.

Guidance:

100% All the contracts are immutable

90% Description relates to investments safety and updates in clear, complete non-software 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](https://twitter.com/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

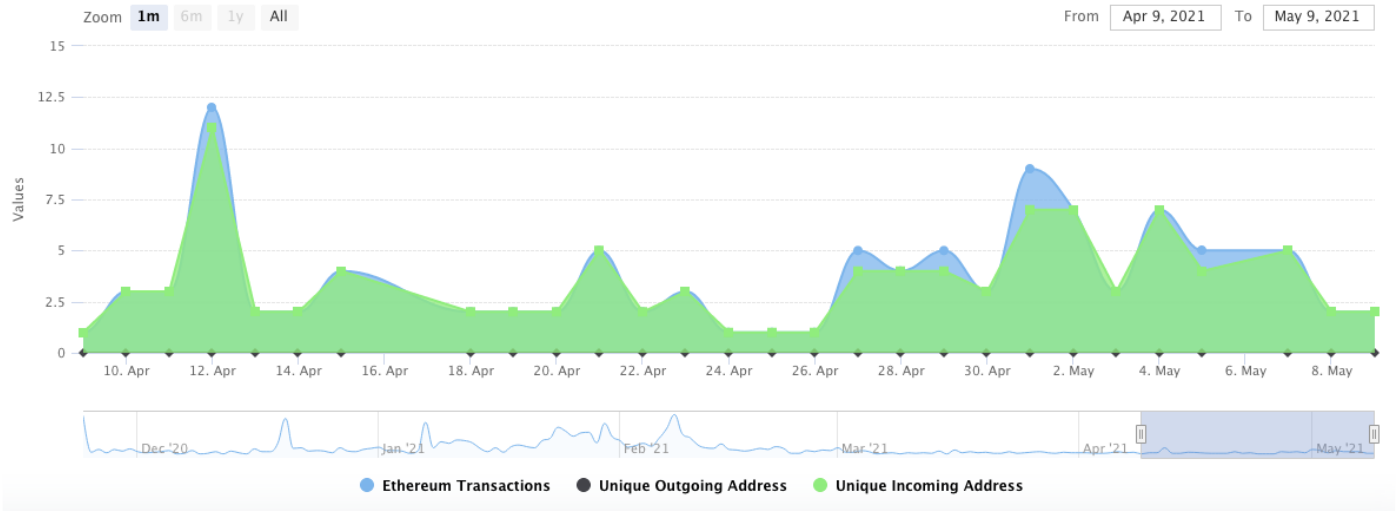
PQ Audit Scoring Matrix (v0.7)	Total	GrowthDefi	
	Points	Answer	Points
Total	260		148
Code and Team			57%
1) Are the executing code addresses readily available? (%)	20	100%	20
2) Is the code actively being used? (%)	5	70%	3.5
3) Is there a public software repository? (Y/N)	5	Y	5
4) Is there a development 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	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	100%	5

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	40%	8
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	Y	5
14) Report of the results (%)	10	0%	0
15) Formal Verification test done (%)	5	0%	0
16) Stress Testing environment (%)	5	100%	5
Security			
17) Did 3rd Party audits take place? (%)	70	70%	49
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	70%	3.5
20) Is the information clear and complete	10	30%	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	97%	
Documentation	45	22%	
Testing	50	36%	
Security	80	70%	
Access Controls	35	44%	

Executing Code Appendix

Token	Mainnet Address
gDAI	0x5301988A8EB906a65b57e9BAF4750A3C74e3E635
gUSDC	0x6dfaabaf237174Fb5E2c12e2130613d64E1a4bbe
gETH	0x3eEE7Fe99640c47ABF43Cd2C2B6A80EB785e38cf
gWBTC	0xe567B3174af8eA368ed536998a597147Ec29De8f
gcDAI	0x4085669d375D7EBb225C05F6128e60C19079ee1c
gcUSDC	0x0e93b2D3969A0a6b71CE21Aa5be417cd4cAC38D0
gcETH	0xF510949599b90f78A0B40aae82539D09b9bE9e28
gcWBTC	0x1085045eF3f1564e4dA4C7315C0B7448d82d5D32
stkGRO	0xD93f98b483CC2F9EFE512696DF8F5deCB73F9497
Token	Kovan Address
gDAI	0x8e7D3c9D614a49d54FA1176B8CE7fcdDcE571a6E
gUSDC	0x7AE53D7076c5Df0762A7e85fa24c01408A63c1e8
gETH	0x4104F56839F8FD1FD67297713213DE447C33556E
gWBTC	0xE45d930b67269CeBf207aAB4dCc200463f439634
gcDAI	0x8Cde5552602DB7563f424d105217e098d96fce36
gcUSDC	0x151ac053B6EEEB604c957f2E1F69F797834DB39b
gcETH	0x9Ca66B0165fF06066cd4f39731bBD2797E4E0691
gcWBTC	0xb389dc7A147065c0F0572b8c3340F6F01D427751

Code Used Appendix



Example Code Appendix

```

1 // SPDX-License-Identifier: GPL-3.0-only
2 pragma solidity ^0.6.0;
3
4 import { SafeMath } from "@openzeppelin/contracts/math/SafeMath.sol";
5
6 import { GToken } from "./GToken.sol";
7 import { G } from "./G.sol";
8 import { GC } from "./GC.sol";
9
10 /**
11  * @dev This library implements data structure abstraction for the delegated
12  *       reserve management code in order to circumvent the EVM contract size limit.
13  *       It is therefore a public library shared by all gcToken Type 2 contracts and
14  *       needs to be published alongside them. See GTokenType2.sol for further
15  *       documentation.
16  */
17 library GCDelegatedReserveManager
18 {
19     using SafeMath for uint256;
20     using GCDelegatedReserveManager for GCDelegatedReserveManager.Self;
21
22     uint256 constant MAXIMUM_COLLATERALIZATION_RATIO = 96e16; // 96% of 50% = 48%
23     uint256 constant DEFAULT_COLLATERALIZATION_RATIO = 66e16; // 66% of 50% = 33%
24     uint256 constant DEFAULT_COLLATERALIZATION_MARGIN = 15e16; // 15% of 50% = 7.5%
25
26     struct Self {
27         address reserveToken;
28         address underlyingToken;
29     }
30

```

```

30         address exchange;
31
32         address miningToken;
33         uint256 miningMinGulpAmount;
34         uint256 miningMaxGulpAmount;
35
36         address borrowToken;
37
38         address growthToken;
39         address growthReserveToken;
40         uint256 growthMinGulpAmount;
41         uint256 growthMaxGulpAmount;
42
43         uint256 collateralizationRatio;
44         uint256 collateralizationMargin;
45     }
46
47     /**
48      * @dev Initializes the data structure. This method is exposed publicly.
49      *      Note that the underlying borrowing token must match the growth
50      *      reserve token given that funds borrowed will be reinvested in
51      *      the provided growth token (gToken).
52      * @param _reserveToken The ERC-20 token address of the reserve token (cToken).
53      * @param _miningToken The ERC-20 token address to be collected from
54      *                      liquidity mining (COMP).
55      * @param _borrowToken The ERC-20 token address of the borrow token (cToken).
56      * @param _growthToken The ERC-20 token address of the growth token (gToken).
57      */
58     function init(Self storage _self, address _reserveToken, address _miningToken, add
59     {
60         address _underlyingToken = GC.getUnderlyingToken(_reserveToken);
61         address _borrowUnderlyingToken = GC.getUnderlyingToken(_borrowToken);
62         address _growthReserveToken = GToken(_growthToken).reserveToken();
63         assert(_borrowUnderlyingToken == _growthReserveToken);
64
65         _self.reserveToken = _reserveToken;
66         _self.underlyingToken = _underlyingToken;
67
68         _self.exchange = address(0);
69
70         _self.miningToken = _miningToken;
71         _self.miningMinGulpAmount = 0;
72         _self.miningMaxGulpAmount = 0;
73
74         _self.borrowToken = _borrowToken;
75
76         _self.growthToken = _growthToken;
77         _self.growthReserveToken = _growthReserveToken;
78         _self.growthMinGulpAmount = 0;
79         _self.growthMaxGulpAmount = 0;
80
81         _self.collateralizationRatio = DEFAULT_COLLATERALIZATION_RATIO;
82         _self.collateralizationMargin = DEFAULT_COLLATERALIZATION_MARGIN;

```

```

83
84         GC.safeEnter(_reserveToken);
85     }
86
87     /**
88     * @dev Sets the contract address for asset conversion delegation.
89     *      This library converts the miningToken into the underlyingToken
90     *      and use the assets to back the reserveToken. See GExchange.sol
91     *      for further documentation. This method is exposed publicly.
92     * @param _exchange The address of the contract that implements the
93     *                  GExchange interface.
94     */
95     function setExchange(Self storage _self, address _exchange) public
96     {
97         _self.exchange = _exchange;
98     }
99
100    /**
101    * @dev Sets the range for converting liquidity mining assets. This
102    *      method is exposed publicly.
103    * @param _miningMinGulpAmount The minimum amount, funds will only be
104    *                              converted once the minimum is accumulated.
105    * @param _miningMaxGulpAmount The maximum amount, funds beyond this
106    *                              limit will not be converted and are left
107    *                              for future rounds of conversion.
108    */
109    function setMiningGulpRange(Self storage _self, uint256 _miningMinGulpAmount, uint256 _miningMaxGulpAmount) public
110    {
111        require(_miningMinGulpAmount <= _miningMaxGulpAmount, "invalid range");
112        _self.miningMinGulpAmount = _miningMinGulpAmount;
113        _self.miningMaxGulpAmount = _miningMaxGulpAmount;
114    }
115
116    /**
117    * @dev Sets the range for converting growth profits. This
118    *      method is exposed publicly.
119    * @param _growthMinGulpAmount The minimum amount, funds will only be
120    *                              converted once the minimum is accumulated.
121    * @param _growthMaxGulpAmount The maximum amount, funds beyond this
122    *                              limit will not be converted and are left
123    *                              for future rounds of conversion.
124    */
125    function setGrowthGulpRange(Self storage _self, uint256 _growthMinGulpAmount, uint256 _growthMaxGulpAmount) public
126    {
127        require(_growthMinGulpAmount <= _growthMaxGulpAmount, "invalid range");
128        _self.growthMinGulpAmount = _growthMinGulpAmount;
129        _self.growthMaxGulpAmount = _growthMaxGulpAmount;
130    }
131
132    /**
133    * @dev Sets the collateralization ratio and margin. These values are
134    *      percentual and relative to the maximum collateralization ratio
135    *      provided by the underlying asset. This method is exposed publicly.

```

```

135     * provided by the underlying asset. This method is exposed publicly.
136
137     * @param _collateralizationRatio The target collateralization ratio,
138     * between lend and borrow, that the
139     * reserve will try to maintain.
140     * @param _collateralizationMargin The deviation from the target ratio
141     * that should be accepted.
142     */
143     function setCollateralizationRatio(Self storage _self, uint256 _collateralizationRatio,
144     {
145         require(_collateralizationMargin <= _collateralizationRatio && _collateralizationMargin > 0);
146         _self.collateralizationRatio = _collateralizationRatio;
147         _self.collateralizationMargin = _collateralizationMargin;
148     }
149
150     /**
151     * @dev Performs the reserve adjustment actions leaving a liquidity room,
152     * if necessary. It will attempt to incorporate the liquidity mining
153     * assets into the reserve, the profits from the underlying growth
154     * investment and adjust the collateralization targeting the
155     * configured ratio. This method is exposed publicly.
156     * @param _roomAmount The underlying token amount to be available after the
157     * operation. This is relevant for withdrawals, once the
158     * room amount is withdrawn the reserve should reflect
159     * the configured collateralization ratio.
160     * @return _success A boolean indicating whether or not both actions succeeded.
161     */

```

SLOC Appendix

Solidity Contracts

Language	Files	Lines	Blanks	Comments	Code	Complexity
Solidity	30	4413	397	2031	1985	162

Comments to Code 2031/1985 = 102%

Javascript Tests

Language	Files	Lines	Blanks	Comments	Code	Complexity
JavaScript	9	1018	209	80	729	

Tests to Code 729/1985 = 36%