0.7

dHEDGE v2 Process Quality Review

Score: 76%

Overview

This is a dHEDGE Process Quality Review completed on 21 Sep 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 **76%**, a **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

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, Polygon

Guidance:

Ethereum
Binance Smart Chain
Polygon
Avalanche
Terra

Code and Team

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

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

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

> **Answer:** 100%

They are available at website https://github.com/dhedge/V2-Public/blob/master/publish/matic/versions.json#L338, in mainnet.json, as indicated in the Appendix.

Note: Although it is in a .json file, there is a hyperlink in the dHedge documentation titled "dHedge V2 Contracts" that leads you straight to it, hence the 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 labeling not clear or easy to find
0% Executing addresses could not be found

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 towards the final score.

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

> **Answer:** 100%

Activity is 25 transactions a day on contract 0xca1207647Ff814039530D7d35df0e1Dd2e91Fa84, 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/dhedge/V2-Public

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%

The repository has 781 commits with 3 branches, making this a healthy 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://au.linkedin.com/in/ermin-nurovic-8a580b125

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.dhedge.org/

How to improve this score:

Ensure that 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:** Yes

dHedge's basic software functions are well explained in its documentation.

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

> **Answer:** 100%

A second gitbook details every deployed contract with a clear explanation on each respective contract's function.

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

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

> **Answer:** 57%

Code examples are in the Appendix. As per the SLOC, there is 57% 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:** 60%

There is strong association between the code and the docs, but there is no explicit traceability between the docs and the github repository.

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:** 40%

Code examples are in the Appendix. As per the SLOC, there is 61% 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

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:** 76%

An audit conducted by Iosiro found the code coverage to be "satisfactory", though it recommends enhancing this test suite. A percentage of 76% code coverage was given.

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:** Yes

Scripts/Instructions location: https://github.com/dhedge/V2-Public

14) Report of the results (%)

> ⚠ **Answer:** 0%

There is no report of these test results.

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 found.

16) Stress Testing environment (%)

> ✓ **Answer:** 100%

Testing has been conducted on the Kovan testnet.

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%

Iosiro conducted an audit before the code was deployed, and the results are public.

Certik also conducted audits on dHEDGE.

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:** 40%

There is a bug bounty of up to $50,000 that is not an active 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

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:** 70%

Admin control information was found under "dHEDGE token". Even though they have a governance section, it does not explicitly state the degree of control that is exerted over the contracts, hence the 70%.

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:** 60%

a) Contract upgradeability is intended through the description of user voting power at https://docs.dhedge.org/dht/introduction

~~b) Type of ownership is not indicated.~~

c) Capabilities for changes in contract are also indicated at https://docs.dhedge.org/dht/introduction.

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:** 30%

The admin control information explains their software well, but does not necessarily explain why and how user investments are/stay safe.

Guidance:

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

How to improve this score:

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

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

> ⚠ **Answer:** 0%

There is no documented pause control.

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

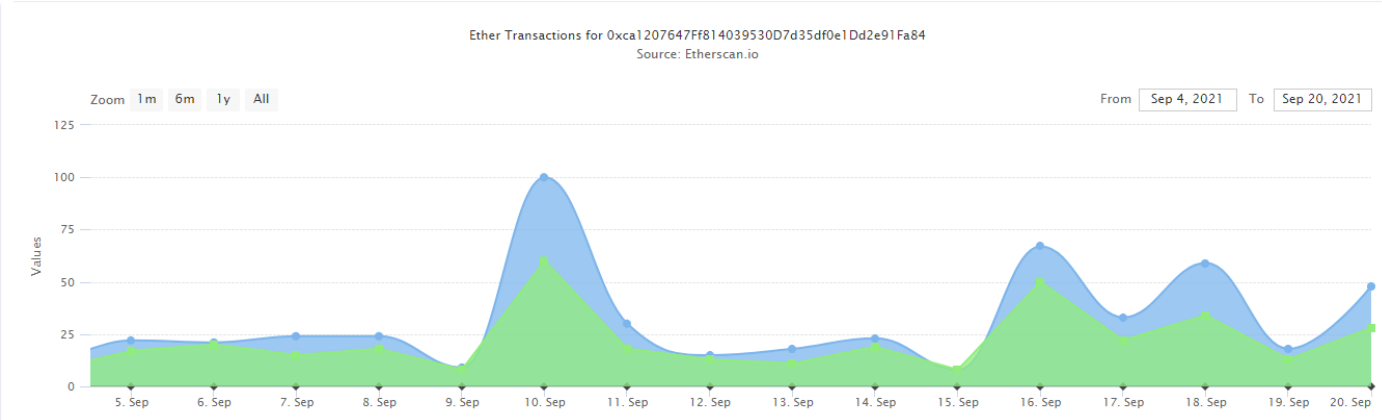
	Total	dHedge v2	
PQ Audit Scoring Matrix (v0.7)	**Points**	**Answer**	**Points**
Total	260		197.15
Code and Team			**76%**
1) Are the executing code addresses readily available? (%)	20	100%	20
2) Is the code actively being used? (%)	5	100%	5
3) Is there a public software repository? (Y/N)	5	Y	5
4) Is there a development history visible? (%)	5	100%	5
5) Is the team public (not anonymous)? (Y/N)	15	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%) c	15	100%	15
9) Are there sufficiently detailed comments for all function	5	57%	2.85
10) Is it possible to trace from software documentation to th	10	60%	6
Testing			
11) Full test suite (Covers all the deployed code) (%)	20	40%	8
12) Code coverage (Covers all the deployed lines of code, or	5	76%	3.8
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	100%	70
18) Is the bug bounty acceptable high? (%)	10	40%	4
Access Controls			
19) Can a user clearly and quickly find the status of the admi	5	70%	3.5
20) Is the information clear and complete	10	60%	6
21) Is the information in non-technical terms	10	30%	3
22) Is there Pause Control documentation including records	10	0%	0
Section Scoring			
Code and Team	50	100%	
Documentation	45	86%	
Testing	50	44%	
Security	80	93%	
Access Controls	35	36%	

Executing Code Appendix

```
{
```

```
      "name": "Sushi LP WBTC-WETH",
      "asset": "0xe62ec2e799305e0d367b0cc3ee2cda135bf89816",
      "assetType": "2",
      "aggregator": "0x1B5E9F94fcB214737bf0cdF7E860Bee65A65500E"
    },
    {
      "name": "Sushi LP WMATIC-WETH",
      "asset": "0xc4e595acdd7d12fec385e5da5d43160e8a0bac0e",
      "assetType": "2",
      "aggregator": "0xAD2D7164CB32c1e54F2C1B9091AcE1bFC12Bf4F3"
    },
    {
      "name": "Sushi LP WETH-AAVE",
      "asset": "0x2813d43463c374a680f235c428fb1d7f08de0b69",
      "assetType": "2",
      "aggregator": "0xd7dD4D2bf5FA90159070d19797b3fFb5A3789483"
    },
    {
      "name": "Sushi LP WETH-CRV",
      "asset": "0x396e655c309676caf0acf4607a868e0cded876db",
      "assetType": "2",
      "aggregator": "0xcc37315085087a0c324d32494eC8E36a1DFE4bFC"
    }
  ],
  "Governance": "0x206CbDa3381e7afdF448621b90f549f89555A588",
  "PoolFactoryProxy": "0xfdc7b8bFe0DD3513Cc669bB8d601Cb83e2F69cB0",
  "PoolLogic": "0xf21848C9852eE543606D3C92962272D70028D3f1",
  "PoolManagerLogic": "0x9A3de79a99adCAA5fd15510B7024a138f6d842A6",
  "AssetHandlerProxy": "0x760FE3179c8491f4b75b21A81F3eE4a5D616A28a",
  "ERC20Guard": "0x3Cf903B98755CBf2174fDBD2e1027e538745A918",
  "UniswapV2RouterGuard": "0x3047511a1b78f63E132E3884c987eA315dd46045",
  "SushiMiniChefV2Guard": "0x47C444b9d91D1F8ceF52f3170218EF89819d6E1b",
  "SushiLPAssetGuard": "0xF291D94C6DE82D67655170dFE90D16204a4b4EBf"
```


Code Used Appendix



Example Code Appendix

```
/// @title Pool Factory
/// @dev A Factory to spawn pools
contract PoolFactory is

```

```
    PausableUpgradeable,
  5 ProxyFactory,
  6 IHasDaoInfo,
  7 IHasFeeInfo,
  8 IHasAssetInfo,
  9 IHasGuardInfo,
 10 IHasPausable
 11 {
 12   using SafeMathUpgradeable for uint256;
 13   using AddressHelper for address;
 14
 15   event FundCreated(
 16     address fundAddress,
 17     bool isPoolPrivate,
 18     string fundName,
 19     string managerName,
 20     address manager,
 21     uint256 time,
 22     uint256 managerFeeNumerator,
 23     uint256 managerFeeDenominator
 24   );
 25
 26   event DAOAddressSet(address daoAddress);
 27
 28   event GovernanceAddressSet(address governanceAddress);
 29
 30   event DaoFeeSet(uint256 numerator, uint256 denominator);
 31
 32   event ExitFeeSet(uint256 numerator, uint256 denominator);
 33
 34   event ExitCooldownSet(uint256 cooldown);
 35
 36   event MaximumSupportedAssetCountSet(uint256 count);
 37
 38   event LogUpgrade(address indexed manager, address indexed pool);
 39
 40   event SetPoolManagerFee(uint256 numerator, uint256 denominator);
 41
 42   event SetMaximumManagerFee(uint256 numerator, uint256 denominator);
 43
 44   event SetMaximumManagerFeeNumeratorChange(uint256 amount);
 45
 46   event SetAssetHandler(address assetHandler);
 47
 48   event SetPoolStorageVersion(uint256 poolStorageVersion);
 49
 50   event SetManagerFeeNumeratorChangeDelay(uint256 delay);
 51
 52   address[] public deployedFunds;
 53
 54   address public override daoAddress;
 55   address public governanceAddress;
 56
```

```
57
    address internal _assetHandler;
58  uint256 internal _daoFeeNumerator;
59  uint256 internal _daoFeeDenominator;
60
61  mapping(address => bool) public isPool;
62
63  uint256 private _MAXIMUM_MANAGER_FEE_NUMERATOR;
64  uint256 private _MANAGER_FEE_DENOMINATOR;
65
66  uint256 internal _exitCooldown;
67
68  uint256 internal _maximumSupportedAssetCount;
69
70  mapping(address => uint256) public poolVersion;
71  uint256 public poolStorageVersion;
72
73  uint256 public override maximumManagerFeeNumeratorChange;
74  uint256 public override managerFeeNumeratorChangeDelay;
75
76  /// @notice Initialize the factory
77  /// @param _poolLogic The pool logic address
78  /// @param _managerLogic The manager logic address
79  /// @param assetHandler The address of the asset handler
80  /// @param _daoAddress The address of the DAO
81  /// @param _governanceAddress The address of the governance contract
82  function initialize(
83    address _poolLogic,
84    address _managerLogic,
85    address assetHandler,
86    address _daoAddress,
87    address _governanceAddress
88  ) external initializer {
89    __ProxyFactory_init(_poolLogic, _managerLogic);
90    __Pausable_init();
91
92    _setAssetHandler(assetHandler);
93
94    _setDAOAddress(_daoAddress);
95
96    _setGovernanceAddress(_governanceAddress);
97
98    _setMaximumManagerFee(5000, 10000);
99
100   _setDaoFee(10, 100); // 10%
101   _setExitCooldown(1 days);
102   setManagerFeeNumeratorChangeDelay(4 weeks);
103   setMaximumManagerFeeNumeratorChange(1000);
104
105   _setMaximumSupportedAssetCount(10);
106
107   _setPoolStorageVersion(230); // V2.3.0;
108 }
```

```
109
110   /// @notice Function to create a new fund
111   /// @param _privatePool A boolean indicating whether the fund is private or not
112   /// @param _manager A manager address
113   /// @param _managerName The name of the manager
114   /// @param _fundName The name of the fund
115   /// @param _fundSymbol The symbol of the fund
116   /// @param _managerFeeNumerator The numerator of the manager fee
117   /// @param _supportedAssets An array of supported assets
118   /// @return fund Address of the fund
119   function createFund(
120     bool _privatePool,
121     address _manager,
122     string memory _managerName,
123     string memory _fundName,
124     string memory _fundSymbol,
125     uint256 _managerFeeNumerator,
126     IHasSupportedAsset.Asset[] memory _supportedAssets
127   ) external returns (address fund) {
128     require(!paused(), "contracts paused");
129     require(_supportedAssets.length <= _maximumSupportedAssetCount, "maximum assets reache
130     require(_managerFeeNumerator <= _MAXIMUM_MANAGER_FEE_NUMERATOR, "invalid manager fee")
131
132     bytes memory poolLogicData =
133       abi.encodeWithSignature(
134         "initialize(address,bool,string,string)",
135         address(this),
136         _privatePool,
137         _fundName,
138         _fundSymbol
139       );
140
141     fund = deploy(poolLogicData, 2);
142
143     bytes memory managerLogicData =
144       abi.encodeWithSignature(
145         "initialize(address,address,string,address,uint256,(address,bool)[])",
146         address(this),
147         _manager,
148         _managerName,
149         fund,
150         _managerFeeNumerator,
151         _supportedAssets
152       );
153
154     address managerLogic = deploy(managerLogicData, 1);
155     // Ignore return value as want it to continue regardless
156     IPoolLogic(fund).setPoolManagerLogic(managerLogic);
157
158     deployedFunds.push(fund);
159     isPool[fund] = true;
160
161     poolVersion[fund] = poolStorageVersion;
```

```
162
163     emit FundCreated(
164       fund,
165       _privatePool,
166       _fundName,
167       _managerName,
168       _manager,
169       block.timestamp,
170       _managerFeeNumerator,
171       _MANAGER_FEE_DENOMINATOR
172     );
173   }
174
```

SLOC Appendix

Solidity Contracts

Language	Files	Lines	Blanks	Comments	Code	Complex
Solidity	23	4609	778	1388	2443	258

Comments to Code 1388/2443 = 57%

Javascript Tests

Language	Files	Lines	Blanks	Comments	Code	Complex
JavaScript	3	1935	330	100	1505	0

Tests to Code 1505/2443 = 61%