Repository Report

Snapshot: 19-Jun-2023 15:33

① Date: 19-Jun-2023 20:18

୍ଦ Label: master

Embold Version: 1.9.16.0

Table of Contents

- 1. Executive Summary
- 2. Metrics
- 3. Trend
- 4. Risk
- 5. KPI Distributions
- 6. High-Risk Components
- 7. Worst Components
- 8. Component-Level Design Issues
- 9. Code Issues
- 10. Appendix

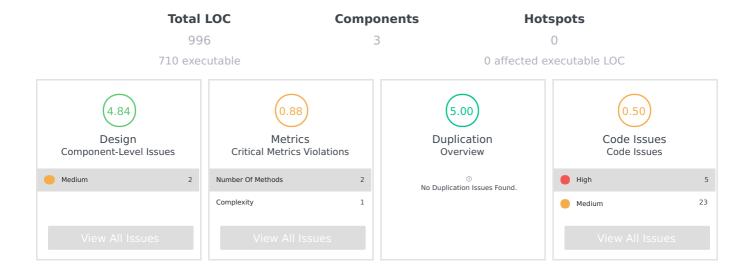
1. Executive Summary

The Embold rating is a numeric representation of the quality of software code, ranging from -5.0 (worst) to +5.0 (best). See the Appendix section for a more in-depth breakdown or read more about ratings online at https://docs.embold.io/embold-score/#embold-rating-system



Kana

19-Jun-2023 15:33





Rating Scale

2. Metrics

The Embold rating is a numeric representation of the quality of software code, ranging from -5.0 (worst) to +5.0 (best). See the Appendix section for a more in-depth breakdown or read more about ratings online at https://docs.embold.io/embold-score/#embold-rating-system



Name	Description	Value
Number Of Methods	The number of methods (NOM) is the total number of methods (or functions) in a component (or class) or file. High NOM indicates a high complexity of the class.	2
Number Of Public Methods	Reflects the number of public methods declared in a class	1
Complexity	Cyclomatic Complexity (CC) is a measure of the program's complexity achieved by measuring the number of linearly independent paths through a program's source code. This measure needs to be applied to sections of source like methods of each class. Presence of IF-ELSE statements or SWITCH statements and FOR loops increases the number of paths in a method. The number of linearly independent paths also means the minimum number of paths that should be tested. The more paths, the higher the number of test cases that need to be implemented. McCabe's method is used to calculate CC.	1

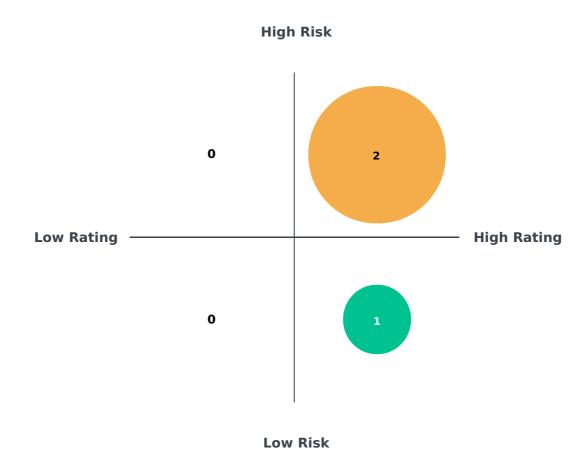
3. Trend



Changes:

Improvements	Deterioration
10101010	10101010
10110110	10110110
10101×	10101×

4. Risk



Category		
Fragile Components (Low Rating & High Risk)	0	
Potential-Risk Components I (High Risk, But High Rating)	2	
Potential-Risk Components II (Low Rating, But Low Risk)	0	
Stable Components (Low Risk & High Rating)	1	

5. KPI Distribution

KPI	Issues
Maintainability	8
Robustness	1
Understandability	15
Analyzability	4

6. High-Risk Components

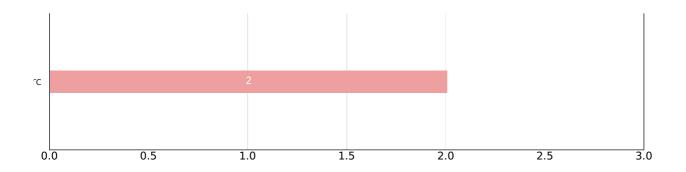
Component	Risk	Commits	Last Commit	Blocker*	Critical*	Major*
KanaAppraiser (Com.mariten.kanatools.Ka naAppraiser)	5.00	2	7 Years Ago	0	0	0
KanaConverter (Com.mariten.kanatools.Ka naConverter)	5.00	2	7 Years Ago	0	0	0
SampleKanaConverter (SampleKanaConverter)	1.00	2	7 Years Ago	0	0	0

^{*}Number of Jira tasks affected by component (Jira plugin required)

7. Worst Components

Component	Overall rating
KanaConverter (com.mariten.kanatools.KanaConverter)	2.52
KanaAppraiser (com.mariten.kanatools.KanaAppraiser)	3.87
SampleKanaConverter (SampleKanaConverter)	4.40

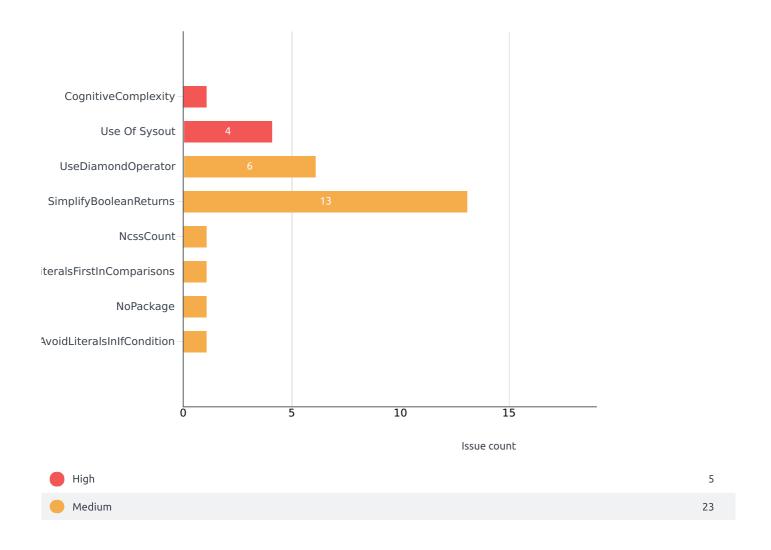
8. Component-Level Design Issues



Name	Description	Criticality	Value
Fat Interface	This anti-pattern applies to modules / classes which have a large interface with too many exposed functions/ methods	Medium	2

Reference Link: https://docs.embold.io/anti-patterns/

9. Code Issues



Reference Link: https://docs.embold.io/code-issues/

10. Appendix

Understanding the Embold Rating

The Embold rating is a numeric representation of the quality of software code, ranging from -5.0 (worst) to +5.0 (best). See the diagram below for further interpretation

- or Greater Indicates good system quality.
- Indicates overall good health, with some localised issues to be cautious of.
- (1.0) (2.0) Indicates some issues, usually localised to specific hotspots.
- (0.0) (1.0) Indicates major issues across multiple hotspots.
- (-5.0) (0.0) Indicates severe, system-wide issues.

Read more about ratings here.

Helpful Resources (Online)

Topic URL

Embold Reference Guide https://docs.embold.io/