

TESWS CODE VIEW REPORT

Overview:

Code Review is a specialized task with the goal of identifying types of weaknesses that exist within a given code base. A large number of bugs can be found and fixed before the code makes it into an official build or into the hands of the test team. Code review is an ongoing process that, ideally, should occur with every code check-in.

Any new code, especially in security sensitive areas, should be code reviewed to discover security vulnerabilities. All vulnerabilities found should not only be placed in a bug database for prioritization and eradication but should be used as input in future code reviews. Over time you can add significantly to the list of bugs you are looking for.

Objective:

1. Identify the type of bugs that are important for your code
2. Generate a list of bugs found in the code that should be prioritized for eradication

Tools:

The following are tools used for static code analysis of TEIW

1. SonarQube

An open-source platform developed by Sonar Source for continuous inspection of code quality to perform automatic reviews with static analysis of code to detect bugs, code smells, and security vulnerabilities.

2. Jenkins

An open source automation server which helps in automating the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery.

3. Gitlab

A web-based Git repository that provides open and private code repositories.

The following are presentation of results of static code analysis by service of each modules
Deliverable 1: TeSWS Billing

New Code

Overall Code

782  Bugs

Reliability 

0  Vulnerabilities

Security 


1  Security Hotspots 

 0.0% Reviewed

Security Review 

2d Debt

722  Code Smells

Maintainability 

 0.0%
Coverage on 619 Lines to cover
Unit Tests

 0.6%
Duplications on 91k Lines
16
Duplicated Blocks

Deliverable 2: TeSWS Frontend 3


New Code

Overall Code

2.7k  Bugs

Reliability 

0  Vulnerabilities

Security 


1  Security Hotspots 

 0.0% Reviewed

Security Review 

1h 43min Debt

53  Code Smells

Maintainability 

 0.0%
Coverage on 881 Lines to cover

-
Unit Tests

 5.5%
Duplications on 29k Lines

73
Duplicated Blocks

Deliverable 3: TeSWS Frontend 3

New Code

Overall Code

593

Bugs

Reliability

D

0

Vulnerabilities

Security

A

5

Security Hotspots

0.0% Reviewed

Security Review

E

10d

Debt

963

Code Smells

Maintainability

A



0.0%

Coverage on 4.9k Lines to cover

-

Unit Tests



9.7%

Duplications on 33k Lines

124


Duplicated Blocks

Deliverable 4: TeSWS Gateway

New Code

Overall Code

0  Bugs

Reliability 

0  Vulnerabilities

Security 


1  Security Hotspots 

 0.0% Reviewed

Security Review 

5min  Debt

1  Code Smells

Maintainability 

 0.0%
Coverage on 24 Lines to cover

-
Unit Tests

 0.0%
Duplications on 25 Lines

0
Duplicated Blocks

Deliverable 5: TeSWS Immigration labour

New Code	Overall Code	
0	Bugs	Reliability A
0	Vulnerabilities	Security A
3	Security Hotspots	0.0% Reviewed Security Review E
3d 3h	Debt	163 Code Smells Maintainability A
0.0%	Coverage on 1.2k Lines to cover	- Unit Tests
7.8%	Duplications on 3k Lines	90 Duplicated Blocks

Deliverable 6: TeSWS Nida

3 Bugs

Reliability

0 Vulnerabilities

Security

5 Security Hotspots

0.0% Reviewed

Security Review

1d 6h Debt

104 Code Smells

Maintainability

0.0%
Coverage on 912 Lines to cover
-
Unit Tests


2.7%
Duplications on 2.1k Lines
8
Duplicated Blocks

Deliverable 7: TeSWS TIC


New Code

Overall Code

1  Bugs

Reliability 

0  Vulnerabilities

Security 


3  Security Hotspots 

 0.0% Reviewed

Security Review 

1d 2h Debt

67  Code Smells

Maintainability 

 0.0%
Coverage on 571 Lines to cover - Unit Tests


 3.8%
Duplications on 1.5k Lines 15 Duplicated Blocks

Deliverable 8: TeSWS TRA

New Code

Overall Code


0  Bugs

Reliability 

0  Vulnerabilities

Security 


12  Security Hotspots 

 0.0% Reviewed

Security Review 

3d 4h  Debt

203  Code Smells

Maintainability 

 0.0%
Coverage on [2.2k](#) Lines to cover

-
Unit Tests

 4.4%
Duplications on [4.8k](#) Lines

24
Duplicated Blocks

Deliverable 9: TeSWS User Management

New Code

Overall Code

0  Bugs

Reliability 

0  Vulnerabilities

Security 


4  Security Hotspots 

 0.0% Reviewed

Security Review 

1d 4h  Debt

82  Code Smells

Maintainability 

 0.0%
Coverage on 1.1k Lines to cover

-
Unit Tests

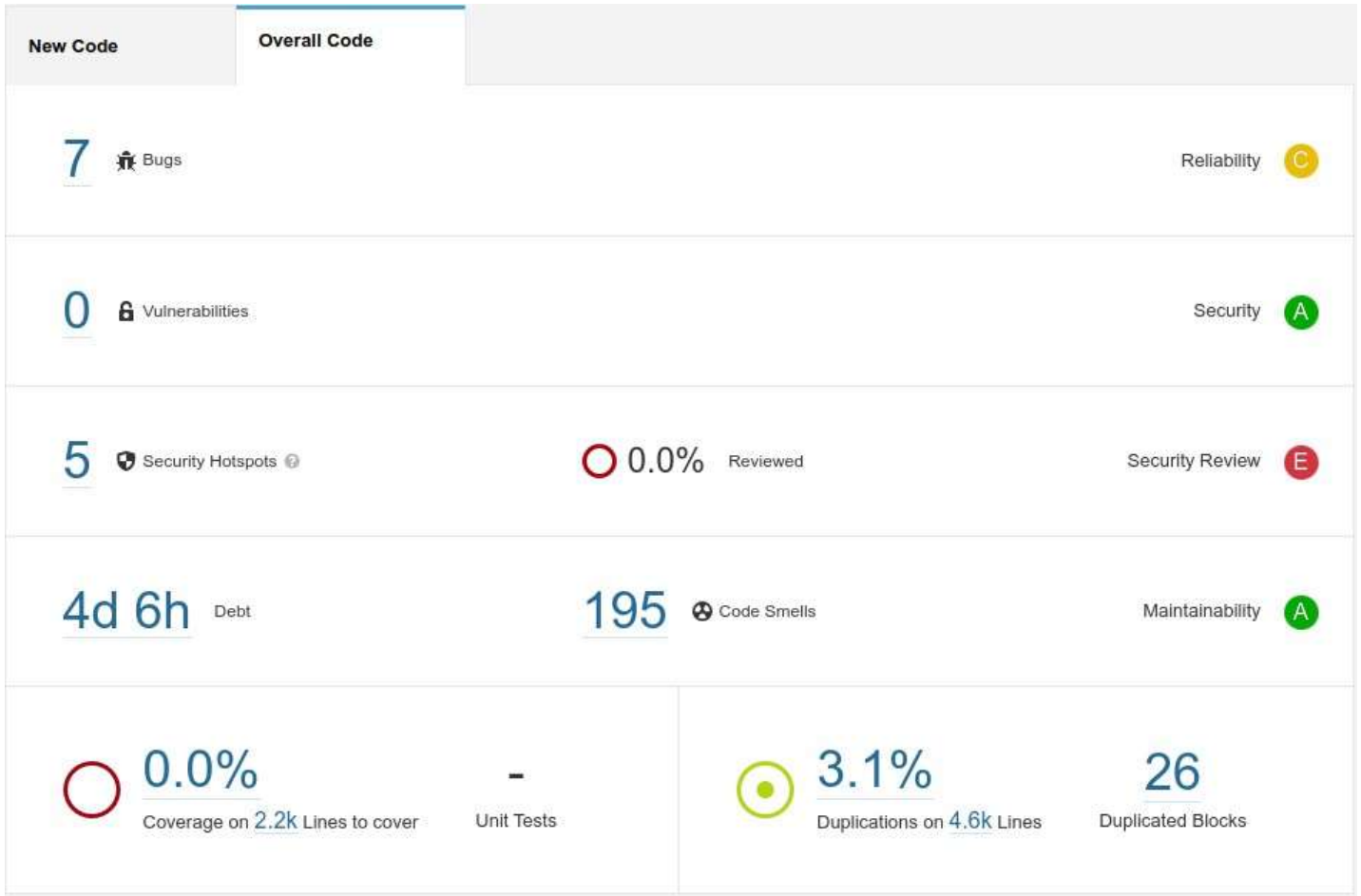
 2.8%
Duplications on 2.4k Lines

12
Duplicated Blocks

Deliverable 10: TeSWS Applicant Registration

New Code	Overall Code	
0 Bugs		Reliability A
0 Vulnerabilities		Security A
2 Security Hotspots	○ 0.0% Reviewed	Security Review E
1d 6h Debt	88 Code Smells	Maintainability A
○ 0.0% Coverage on <u>810</u> Lines to cover	- Unit Tests	○ 2.9% Duplications on <u>2.3k</u> Lines
		17 Duplicated Blocks

Deliverable 11: TeSWS Company Business License



Deliverable 12: TeSWS Messaging Engine

New Code	Overall Code	
7 Bugs		Reliability
2 Vulnerabilities		Security
5 Security Hotspots	0.0% Reviewed	Security Review
4d Debt	215 Code Smells	Maintainability
0.0% Coverage on <u>2k</u> Lines to cover	- Unit Tests	3.5% Duplications on <u>5.5k</u> Lines
		<u>16</u> Duplicated Blocks

Deliverable 13: TeSWS TRA Exemptions

New Code

Overall Code


0  Bugs

Reliability 

0  Vulnerabilities

Security 

4  Security Hotspots 

 0.0% Reviewed

Security Review 

1d 5h  Debt

115  Code Smells

Maintainability 

 0.0%
Coverage on 1.4k Lines to cover

-
Unit Tests

 2.7%
Duplications on 2.9k Lines

14
Duplicated Blocks

Conclusion

These details are designed to provide the developers with proof that the stated weaknesses exist as well as to provide examples that the developers can use to find and fix similar areas of the code. As mentioned before, the Secure Code Review does not claim to find every issue; as such the development team should use the information in these findings as an opportunity to improve the entire code base.

Issue Types

1. **Bugs** - A coding mistake that can lead to an error or unexpected behavior at runtime.
2. **Vulnerability** - A point in your code that's open to attack.
3. **Code Smell** - A maintainability issue that makes your code confusing and difficult to maintain.

Each issue has one of five severities:

1. BLOCKER

Bug with a high probability to impact the behavior of the application in production: memory leak, unclosed JDBC connection, The code **MUST** be fixed immediately.

2. CRITICAL

Either a bug with a low probability to impact the behavior of the application in production or an issue which represents a security flaw: empty catch block, SQL injection, ... The code **MUST** be immediately reviewed.

3. MAJOR

Quality flaw which can highly impact the developer productivity: uncovered piece of code, duplicated blocks, unused parameters, ...

4. MINOR

Quality flaw which can slightly impact the developer productivity: lines should not be too long, "switch" statements should have at least 3 cases, ...

5. INFO

Neither a bug nor a quality flaw, just a finding.

Recommendations:

1. SonarLint: can help developers by letting you perform local analyses to check your code before pushing it back to the GitLab. Every developer should install in local machine IDE to help to analyse code before push to GitLab.

This feature is available to users of: Visual Studio, IntelliJ, VS Code and Eclipse.