Features

The main goal of the plugin is to integrate JIRA with your SonarQube instance and code quality metrics.

This is very important because not all the people access SonarQube to review their code quality data. With this plugin, we are providing all the information in JIRA, so that the users don't need to change their context to SonarQube.

OpenAI integration

Get easy and fast possible resolutions to your sonarqube evidences created in Jira!

We are proud to announce that our connector has a new functionality through the integration of open ai in our plugin.

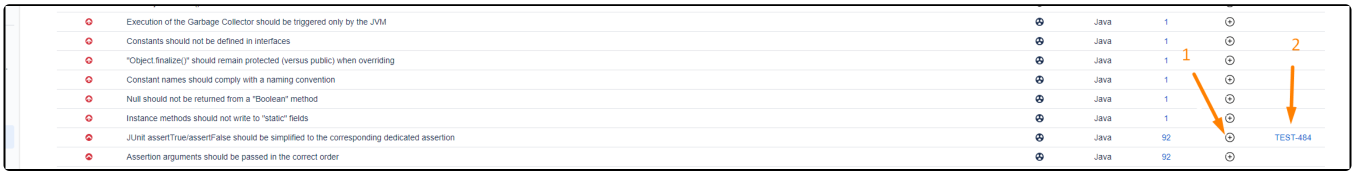
Now every time an issue is created in jira from our connector, in the comments of the newly created issue, one or several possible solutions given by open ai will also be created, to correct the sonar evidence associated with the issue. In this way we help to improve the quality of the code and the maintenance of the applications.

If the Jira issue is associated with several evidences for the same rule, a maximum of 3 comments will be created in the subject with the possible solutions offered by open ai.

These comments are suggestions for solving the problem in code using the OpenAI API. If you don't have a token, you can generate it from here

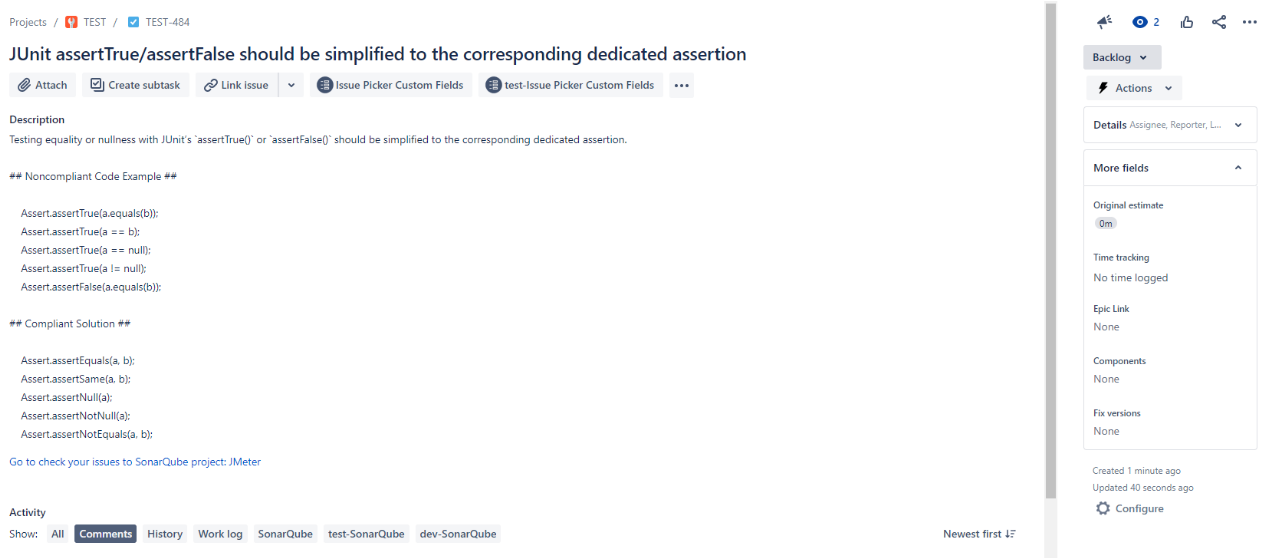
The consultation for the resolution of the evidence is not instantaneous, we would appreciate it if you are a little patient when it comes to obtaining the answers.

"Create issue from the Sonarqube Connector Panel" :



Execution of the Garbage Collector should be triggered only by the JVM		Java	1	⊕	
Constants should not be defined in interfaces		Java	1	⊕	
"Object.finalize()" should remain protected (versus public) when overriding		Java	1	⊕	
Constant names should comply with a naming convention		Java	1	⊕	
Null should not be returned from a "Boolean" method		Java	1	⊕	
Instance methods should not write to "static" fields		Java	1	⊕	
JUnit assertTrue/assertFalse should be simplified to the corresponding dedicated assertion		Java	92	⊕	TEST-484
Assertion arguments should be passed in the correct order		Java	92	⊕	

Access to the Jira issue created to get suggested resolutions :



JUnit assertTrue/assertFalse should be simplified to the corresponding dedicated assertion

Description

Testing equality or nullness with JUnit's `assertTrue()` or `assertFalse()` should be simplified to the corresponding dedicated assertion.

```
## Noncompliant Code Example ##

  Assert.assertTrue(a.equals(b));
  Assert.assertTrue(a == b);
  Assert.assertTrue(a == null);
  Assert.assertTrue(a != null);
  Assert.assertFalse(a.equals(b));

## Compliant Solution ##

  Assert.assertEquals(a, b);
  Assert.assertSame(a, b);
  Assert.assertNull(a);
  Assert.assertNotNull(a);
  Assert.assertNotEquals(a, b);
```

Go to check your issues to SonarQube project: JMeter

Activity

Original estimate
0m

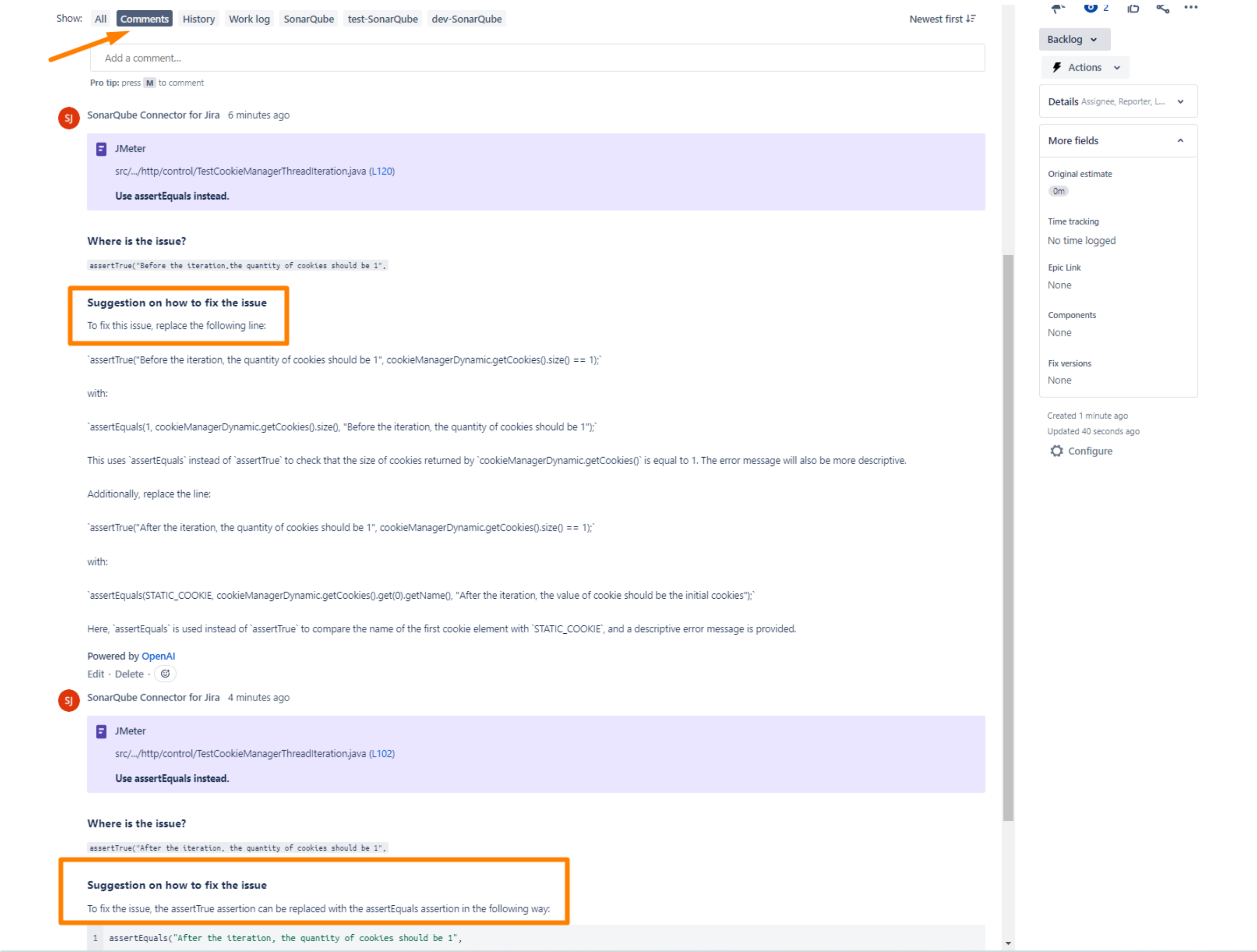
Time tracking
No time logged

Epic Link
None

Components
None

Fix versions
None

Created 1 minute ago
Updated 40 seconds ago


Show: All Comments History Work log SonarQube test-SonarQube dev-SonarQube
Newest first
Add a comment...
Pro tip: press M to comment
Backlog
Actions
Details Assignee, Reporter, L...
More fields
Configure
Edit · Delete ·

Original estimate
0m

Time tracking
No time logged

Epic Link
None

Components
None

Fix versions
None

Created 1 minute ago
Updated 40 seconds ago

SonarQube Connector for Jira 6 minutes ago

JMeter

src/.../http/control/TestCookieManagerThreadIteration.java (L120)

Use assertEquals instead.

Where is the issue?

```
assertTrue("Before the iteration,the quantity of cookies should be 1",
```

Suggestion on how to fix the issue

To fix this issue, replace the following line:

`assertTrue("Before the iteration, the quantity of cookies should be 1", cookieManagerDynamic.getCookies().size() == 1);`

with:

`assertEquals(1, cookieManagerDynamic.getCookies().size(), "Before the iteration, the quantity of cookies should be 1");`

This uses `assertEquals` instead of `assertTrue` to check that the size of cookies returned by `cookieManagerDynamic.getCookies()` is equal to 1. The error message will also be more descriptive.

Additionally, replace the line:

`assertTrue("After the iteration, the quantity of cookies should be 1", cookieManagerDynamic.getCookies().size() == 1);`

with:

`assertEquals(STATIC_COOKIE, cookieManagerDynamic.getCookies().get(0).getName(), "After the iteration, the value of cookie should be the initial cookies");`

Here, `assertEquals` is used instead of `assertTrue` to compare the name of the first cookie element with `STATIC_COOKIE`, and a descriptive error message is provided.

Powered by OpenAI

SonarQube Connector for Jira 4 minutes ago

JMeter

src/.../http/control/TestCookieManagerThreadIteration.java (L102)

Use assertEquals instead.

Where is the issue?

```
assertTrue("After the iteration, the quantity of cookies should be 1",
```

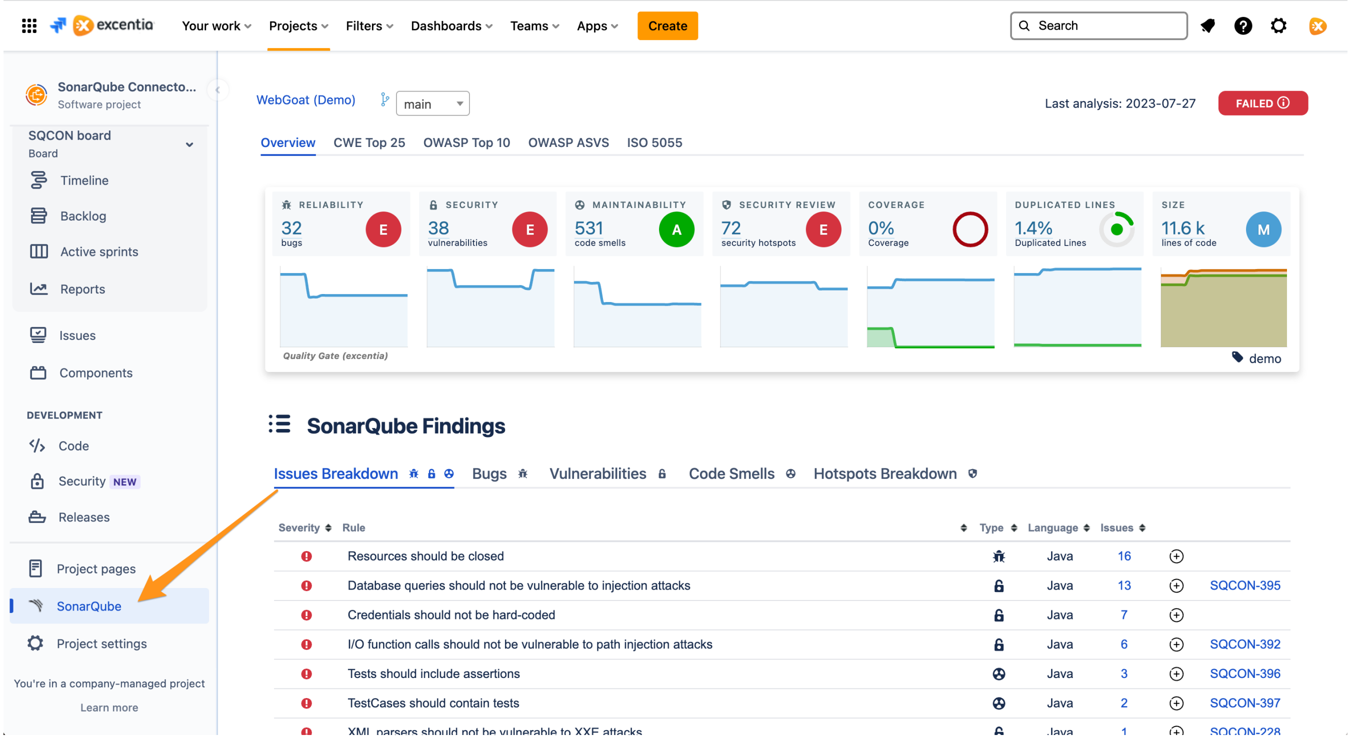
Suggestion on how to fix the issue

To fix the issue, the assertTrue assertion can be replaced with the assertEquals assertion in the following way:

```
assertEquals("After the iteration, the quantity of cookies should be 1",
```

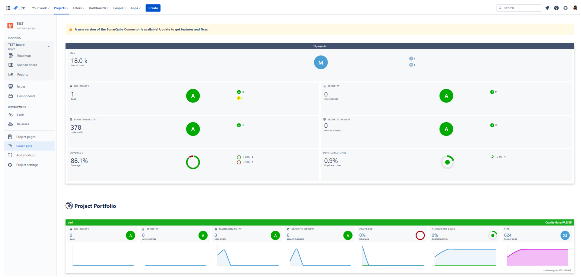
Features at project level

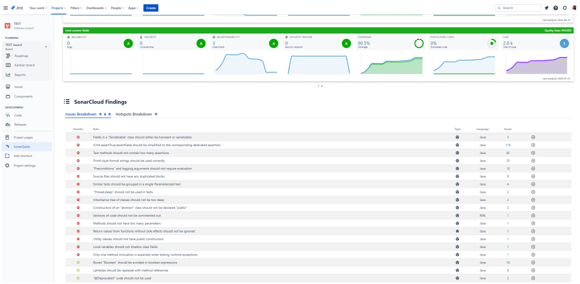
You can browse your project code quality at anytime by **using the project tab panel in your project sidebar**. You will access the full SonarQube dashboard with all your summary and details:



Severity	Rule	Type	Language	Issues		
●	Resources should be closed	🐞	Java	16	⊕	
●	Database queries should not be vulnerable to injection attacks	🔒	Java	13	⊕	SQCON-395
●	Credentials should not be hard-coded	🔒	Java	7	⊕	
●	I/O function calls should not be vulnerable to path injection attacks	🔒	Java	6	⊕	SQCON-392
●	Tests should include assertions	☢	Java	3	⊕	SQCON-396
●	TestCases should contain tests	☢	Java	2	⊕	SQCON-397
●	XML parsers should not be vulnerable to XXE attacks	🔒	Java	1	⊕	SQCON-228

Project Portfolio





Since 1.4 version you can link multiple projects to your Jira project in SonarQube Connector settings.

You just have to provide a comma separated list of project keys instead of only one. Note that if the list if too big then the page will take long to display.

This is very useful if your Jira project manages several software components and you want to have all of them in your SonarQube tab panel.

SonarQube Connector for Jira will aggregate all the metrics into a single view.

This new view provides:

- **A header section** with all the collected metrics and global ratings. This includes the project distribution for each rating (size, reliability, security and maintainability) and the aggregated measures.
- **A project portfolio section** with the information for each project, including last analysis date and quality gate.

How do we compute the global ratings?

If you configure multiple projects then you will get a global rating for the main quality metrics: size, reliability, security and maintainability:

- **Size rating**: this is the sum of all lines of code of all the projects. The rating is computed exactly the same as a individual project. See Size Rating for more information.
- **Reliability, Security and Maintainability rating**: these ratings are computed by giving values from 1 to 5 to each project depending on the rating (A=1 ... E=5). Then we sum all the values and compute the average number. This number is rounded to the nearest value from 1 to 5. This value will be the global rating for your projects.

SonarQube Issues Breakdown

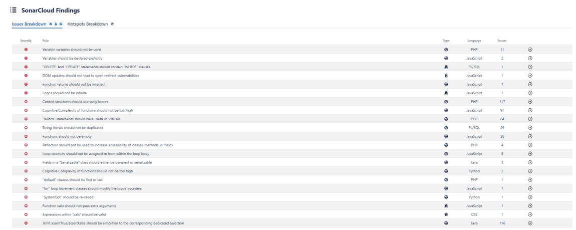
Since 1.5 version you are able to see SonarQube issues directly in your project tab panel.

A new section has been added to your panel so that you can see all issues of the SonarQube resources linked through the connector.

Issues will be summarized if you have more than one resource key in your settings.

This way, as a manager, you can review your top issues inside your Jira project. All the information in the same place.

If you need issue details, we have provided a link to your SonarQube /SonarCloud instance.



SonarCloud Findings

Issues Breakdown | Hotspots Breakdown

Severity	Rule	Type	Language	Issues	
●	Variable variables should not be used	☢	PHP	11	⊕
●	Variables should be declared explicitly	☢	JavaScript	2	⊕
●	"DELETE" and "UPDATE" statements should contain "WHERE" clauses	🐞	PL/SQL	1	⊕
●	DOM updates should not lead to open redirect vulnerabilities	🔒	JavaScript	1	⊕
●	Function returns should not be invariant	☢	JavaScript	1	⊕
●	Loops should not be infinite	🐞	JavaScript	1	⊕
◉	Control structures should use curly braces	☢	PHP	117	⊕
◉	Cognitive Complexity of functions should not be too high	☢	JavaScript	97	⊕
◉	"switch" statements should have "default" clauses	☢	PHP	84	⊕
◉	String literals should not be duplicated	☢	PL/SQL	29	⊕
◉	Functions should not be empty	☢	JavaScript	20	⊕
◉	Reflection should not be used to increase accessibility of classes, methods, or fields	☢	PHP	4	⊕
◉	Loop counters should not be assigned to from within the loop body	☢	JavaScript	3	⊕
◉	Fields in a "Serializable" class should either be transient or serializable	☢	Java	3	⊕
◉	Cognitive Complexity of functions should not be too high	☢	Python	2	⊕
◉	"default" clauses should be first or last	☢	PHP	1	⊕
◉	"for" loop increment clauses should modify the loops' counters	☢	JavaScript	1	⊕
◉	"SystemExit" should be re-raised	☢	Python	1	⊕
◉	Function calls should not pass extra arguments	🐞	JavaScript	1	⊕
◉	Expressions within "calc" should be valid	🐞	CSS	1	⊕
◉	JUnit assertTrue/assertFalse should be simplified to the corresponding dedicated assertion	☢	Java	116	⊕

Create Jira Issues from SonarQube Issues

Since 1.7.1 version you can create SonarQube issues in Jira.

A button has been added in the row of the SonarQube rule to create them.

Issue will be created as follows:

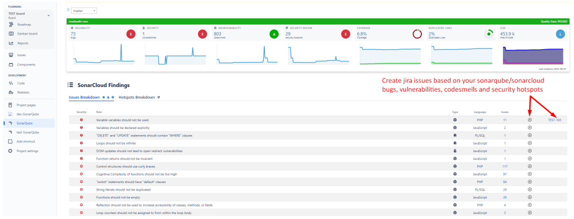
- **Summary**: Name of the SonarQube rule.
- **Description**: A link to SonarQube issues. (If you have more than one project, a link is created only to projects that have issues)

If the SonarQube rules already have an associated issue, you will get a warning saying the issue cannot be created because it already exists.

If a new SonarQube analyis is launched and some issues have already been resolved and new ones have appeared, pressing the button will only create a new issues with the new ones.

When issue is created, a Jira issue link appears in the same row of SonarQube rule. If issue has been deleted or resolved, nothing appears.

> **Note**
>
> Since 2.3 the creation of issues with required fields is supported.
>
> In case the configured issue type has required fields, the create issue dialog will be opened to fill the required fields.



Create jira issues based on your sonarqube/sonarcloud bugs, vulnerabilities, codesmells and security hotspots

SonarCloud Findings

Issues Breakdown | Hotspots Breakdown

Severity	Rule	Type	Language	Issues		
●	Variable variables should not be used	☢	PHP	11	⊕	TEST-165
●	Variables should be declared explicitly	☢	JavaScript	2	⊕	
●	"DELETE" and "UPDATE" statements should contain "WHERE" clauses	🐞	PL/SQL	1	⊕	
●	Loops should not be infinite	🐞	JavaScript	1	⊕	
●	DOM updates should not lead to open redirect vulnerabilities	🔒	JavaScript	1	⊕	
●	Function returns should not be invariant	☢	JavaScript	1	⊕	
◉	Control structures should use curly braces	☢	PHP	117	⊕	
◉	Cognitive Complexity of functions should not be too high	☢	JavaScript	97	⊕	
◉	"switch" statements should have "default" clauses	☢	PHP	84	⊕	
◉	String literals should not be duplicated	☢	PL/SQL	29	⊕	
◉	Functions should not be empty	☢	JavaScript	20	⊕	
◉	Reflection should not be used to increase accessibility of classes, methods, or fields	☢	PHP	4	⊕	
◉	Loop counters should not be assigned to from within the loop body	☢	JavaScript	3	⊕	

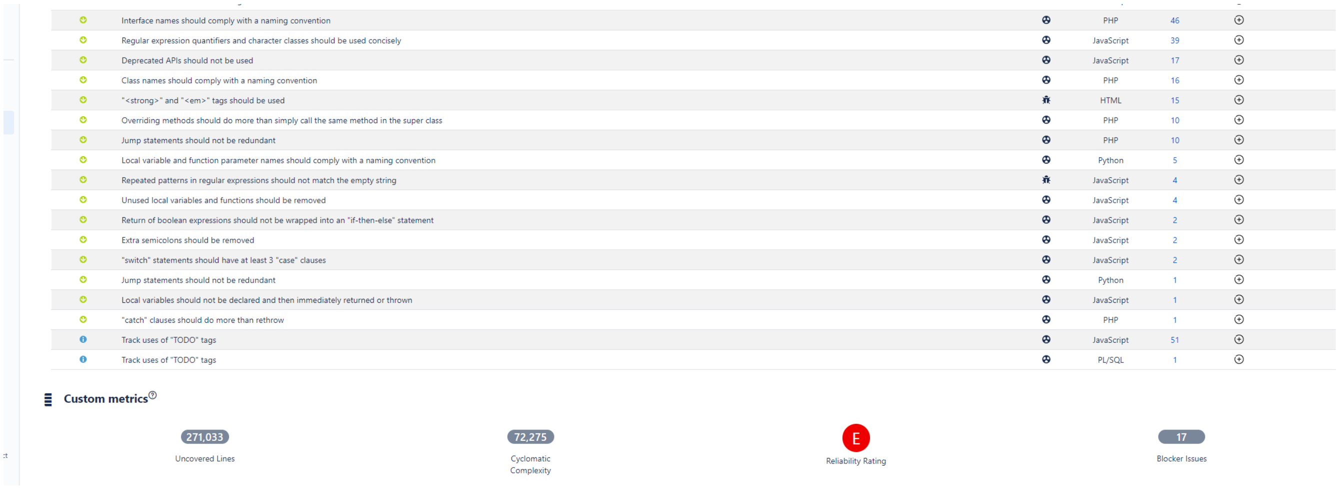
Custom Metrics

Since 1.8 version you can add a list of custom metric keys to display in the dashboard, you can specify 1 or a list of metric keys separated by commas. For example: blocker_violations, major_violations, classes.

==================================

Single project

==================================



Interface names should comply with a naming convention	PHP	46	⊕
Regular expression quantifiers and character classes should be used concisely	JavaScript	39	⊕
Deprecated APIs should not be used	JavaScript	17	⊕
Class names should comply with a naming convention	PHP	16	⊕
"<strong>" and "<em>" tags should be used	HTML	15	⊕
Overriding methods should do more than simply call the same method in the super class	PHP	10	⊕
Jump statements should not be redundant	PHP	10	⊕
Local variable and function parameter names should comply with a naming convention	Python	5	⊕
Repeated patterns in regular expressions should not match the empty string	JavaScript	4	⊕
Unused local variables and functions should be removed	JavaScript	4	⊕
Return of boolean expressions should not be wrapped into an "if-then-else" statement	JavaScript	2	⊕
Extra semicolons should be removed	JavaScript	2	⊕
"switch" statements should have at least 3 "case" clauses	JavaScript	2	⊕
Jump statements should not be redundant	Python	1	⊕
Local variables should not be declared and then immediately returned or thrown	JavaScript	1	⊕
"catch" clauses should do more than rethrow	PHP	1	⊕
Track uses of "TODO" tags	JavaScript	51	⊕
Track uses of "TODO" tags	PL/SQL	1	⊕

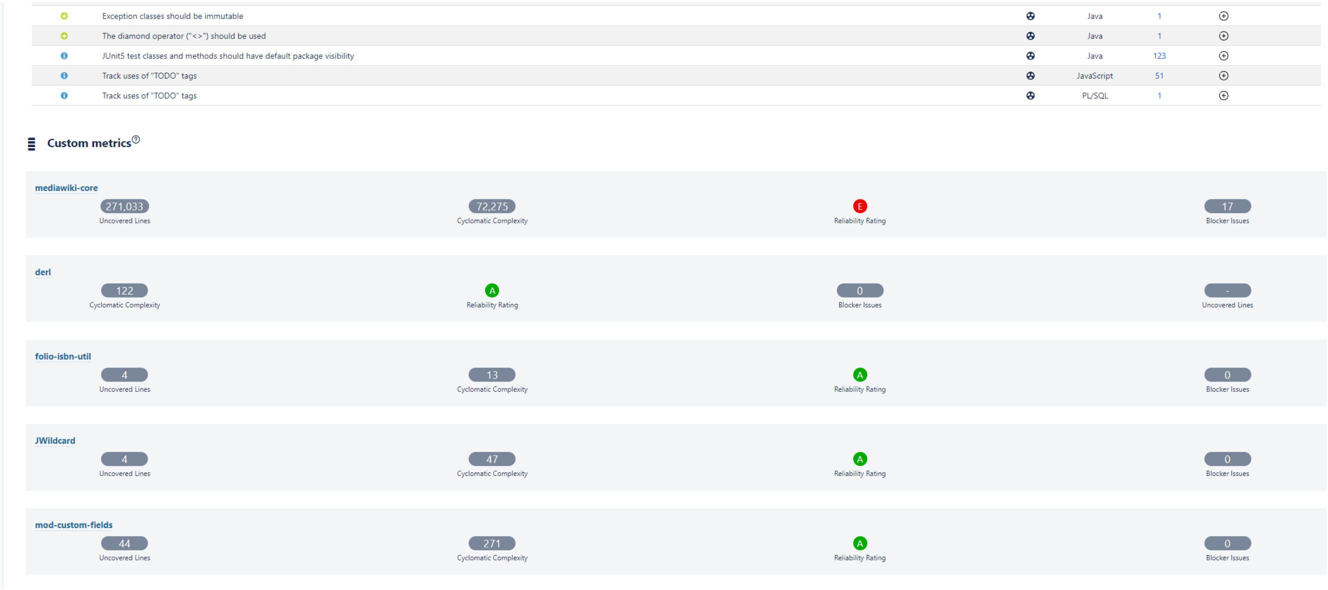
Custom metricsⓘ

271,033	72,275	E	17
Uncovered Lines	Cyclomatic Complexity	Reliability Rating	Blocker Issues

==================================

Multi project

==================================



Exception classes should be immutable	Java	1	⊕
The diamond operator ("<>") should be used	Java	1	⊕
JUnit5 test classes and methods should have default package visibility	Java	123	⊕
Track uses of "TODO" tags	JavaScript	51	⊕
Track uses of "TODO" tags	PL/SQL	1	⊕

Custom metricsⓘ

mediawiki-core

271,033	72,275	E	17
Uncovered Lines	Cyclomatic Complexity	Reliability Rating	Blocker Issues

derl

122	A	0	-
Cyclomatic Complexity	Reliability Rating	Blocker Issues	Uncovered Lines

folio-isbn-util

4	13	A	0
Uncovered Lines	Cyclomatic Complexity	Reliability Rating	Blocker Issues

JWildcard

4	47	A	0
Uncovered Lines	Cyclomatic Complexity	Reliability Rating	Blocker Issues

mod-custom-fields

44	271	A	0
Uncovered Lines	Cyclomatic Complexity	Reliability Rating	Blocker Issues