

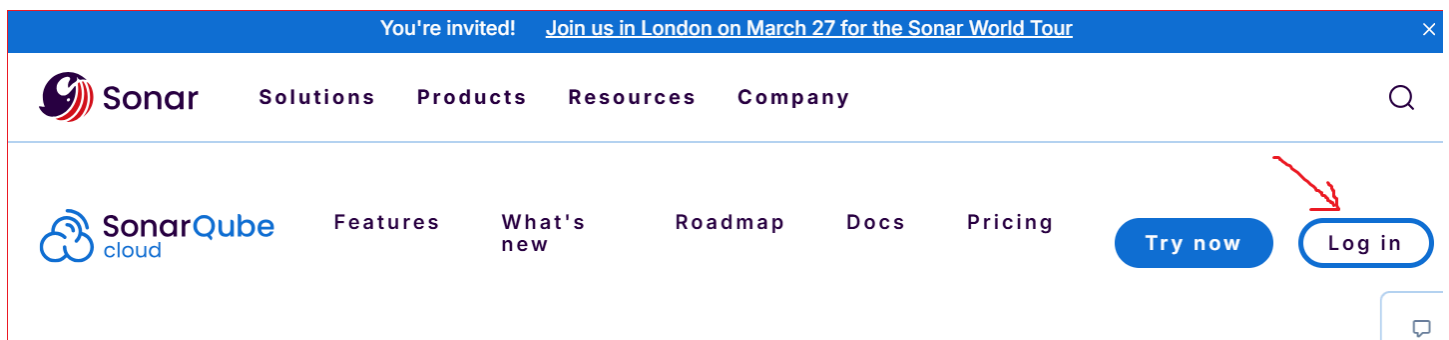
# [CS304] Tutorial 7 - Linters and Code Review Tools

## 1. SonarQube

SonarQube Cloud is a code analysis tool. Let's follow the steps in [SonarQube doc](#) to analyse our Teedy repo with SonarQube. See how much information it provides us.

### 1.1 Sign up SonarQube with Github account

First, go to [SonarQuad products](#) page, click "Log in" or "Sign up" to login SonarQuad for the first time.



Choose "Log in or Sign up with GITHUB", then authorize SonarQube to access your Github account.

You only need to follow the steps.

### 1.2 Import an organization from GitHub

In the SonarQube welcome page, choose "Import an organization from GitHub". Then select your Github account.

Choose "Free" plan and follow the steps.

After you finish this step, if you open your Github account, go to "Settings" and then "Applications", you will see "SonarQubeCloud" in your list:

## Applications

📄 Installed GitHub Apps   🗄️ Authorized GitHub Apps   👤 Authorized OAuth Apps

GitHub Apps augment and extend your workflows on GitHub with commercial, open source, and homegrown tools.



GitHub Learning Lab

Automatic

Configure



SonarQubeCloud

Configure



Travis CI

Configure

## 1.3 Import your repo to analyse

In the next page, choose your repo for SonarQube to analyze:

The screenshot shows the SonarQube cloud interface. At the top, there's a navigation bar with 'My Projects', 'My Issues', and 'Explore'. Below this, the main heading is 'Analyze projects'. A sub-heading says 'Select repositories from one of your GitHub organization.' Under 'Organization', there's a text input field containing 'selab722'. Below that is a link 'Import another organization'. A checkbox labeled 'Select all available repositories' is present. Below that, a list of repositories is shown, with a red arrow pointing to the first one, 'Teedy', which has a checkbox next to it.

## 1.4 Choose new code definition

In the next step, you need to choose the "new code definition" of your repo. This definition defines which part of the code is new code so you can [clean as you code](#).

Here I choose "previous version" since Github is based on version control.

## 1.5 See analyze report

Now SonarQube will take some time to analyze code and produce the report. The report contains many statistics and recommendations of coding.

Security

5 Open issues

D

Reliability

170 Open issues

E

Maintainability

2.8k Open issues

A

Accepted Issues

0

Coverage

A few extra steps are needed for SonarQube Cloud to analyze your code coverage.  
[Set up coverage analysis](#)

Duplications

4.8%

No conditions set on 93k Lines

Security Hotspots

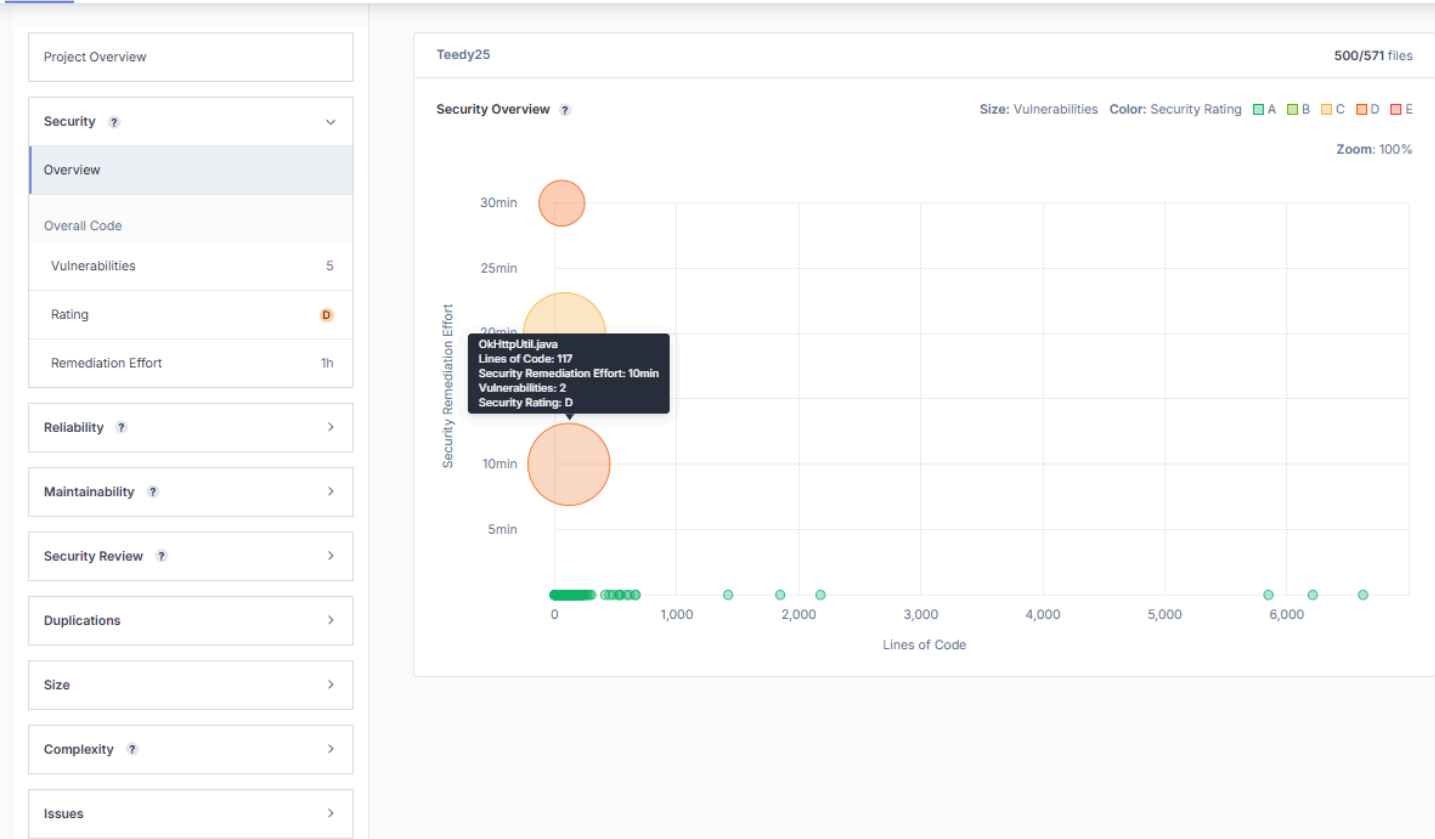
40

Lines of CodeSecurityReliabilityMaintainabilitySecurity HotspotsCoverageDuplications

Teady25

docs-android/app/src/main	6,864	4	0	55	5	—	0.8%
docs-core	10,895	1	6	240	6	—	3.0%
docs-web	40,347	0	164	2471	29	—	6.2%
docs-web-common	917	0	0	27	0	—	0.0%
pom.xml	414	0	0	0	0	—	0.0%

MeasuresCodeActivity



Security Hotspots

Measures

Code

Activity

0.0% Security Hotspots Reviewed

To review

Fixed

Safe

40 Security Hotspots to review

Review priority: High

Authentication

1

'PASSWORD' detected in this expression, review this potentially hard-coded password.

Review priority: Medium

Denial of Service (DoS)

19

Code Injection (RCE)

1

Weak Cryptography

6

Review priority: Low

Encryption of Sensitive Data

1

Insecure Configuration

3

Others

9

40 of 40 shown

'PASSWORD' detected in this expression, review this potentially hard-coded password.

Hard-coded passwords are security-sensitive [java:S2068](#)

Status: To Review

This Security Hotspot needs to be reviewed to assess whether the code poses a risk.

Review

Where is the risk?

What's the risk?

Assess the risk

How can I fix it?

Activity

docs-core/.../java/com/sismics/docs/core/constant/Constants.java

Show 15 more lines

16

17

18

19

20

21

22

23

24

25

26

public static final String DEFAULT\_TIMEZONE\_ID = "Europe/London";

/\*\*

\* Administrator's default password ("admin").

\*/

public static final String DEFAULT\_ADMIN\_PASSWORD = "\$2y\$10\$Xg8EEKVUehutD11m6qQhVeFz75HQHl1jQzjf2KkVsR2c7aV2vyyjK";

'PASSWORD' detected in this expression, review this potentially hard-coded password.

/\*\*

\* Administrator's default email.

\*/

public static final String DEFAULT\_ADMIN\_EMAIL = "admin@localhost";

Show 80 more lines

Review priority: High

Category: Authentication

Assignee: Yida Tao

## 2. JArchitect

JArchitect is a static code analysis tool to show you some hints of your project's code quality. Have a glance of its features:

<https://www.jarchitect.com/features>.

It is a non-free software and you can try it free for a few days:

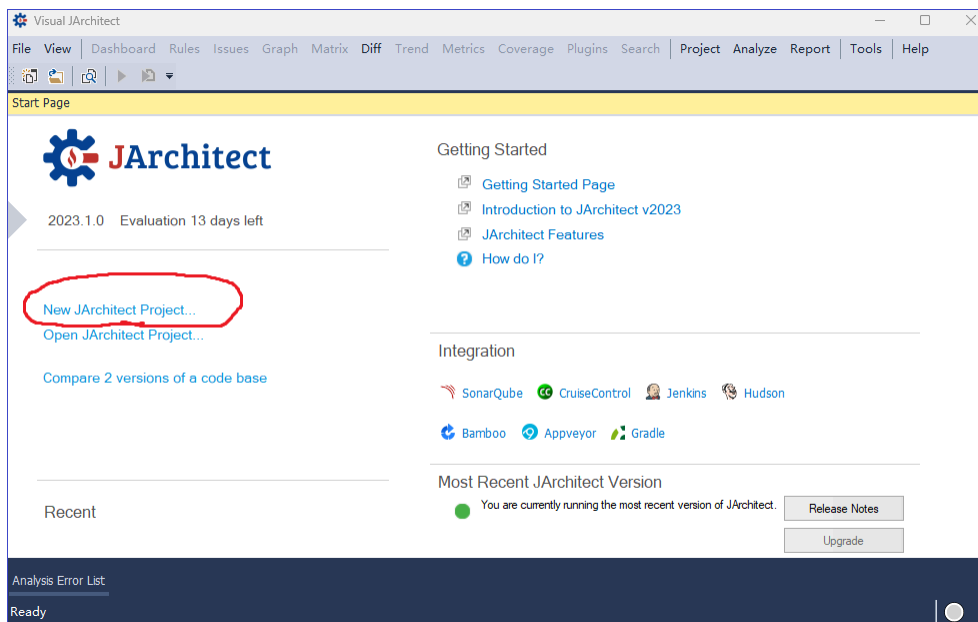
<https://www.jarchitect.com/download>.

### 2.1 Getting started

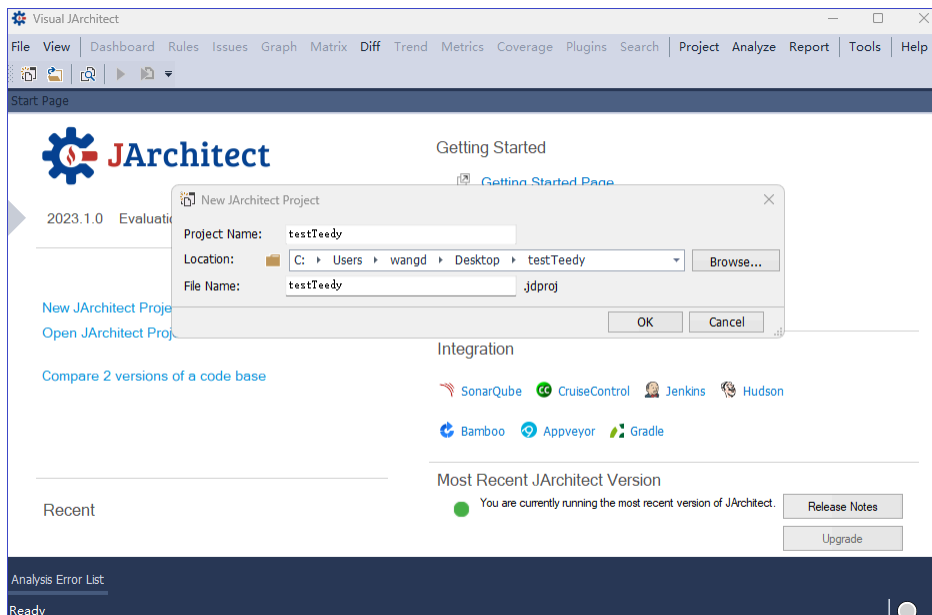
To start with JArchitect, first download the package and uncompress it:

Integration	2023/3/30 17:12
Lib	2023/3/30 17:12
Plugins	2023/3/30 17:13
SourceAnalyzer	2023/3/30 17:13
Tools	2023/3/30 17:13
JArchitect.Console.exe	2023/3/30 17:12
JArchitect.PowerTools.exe	2023/3/30 17:12
VisualJArchitect.exe	2023/3/30 17:13

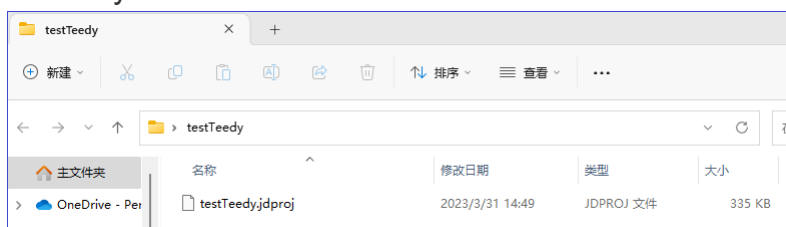
Then double click "VisualJArchitect.exe" to run the gui of JArchitect:



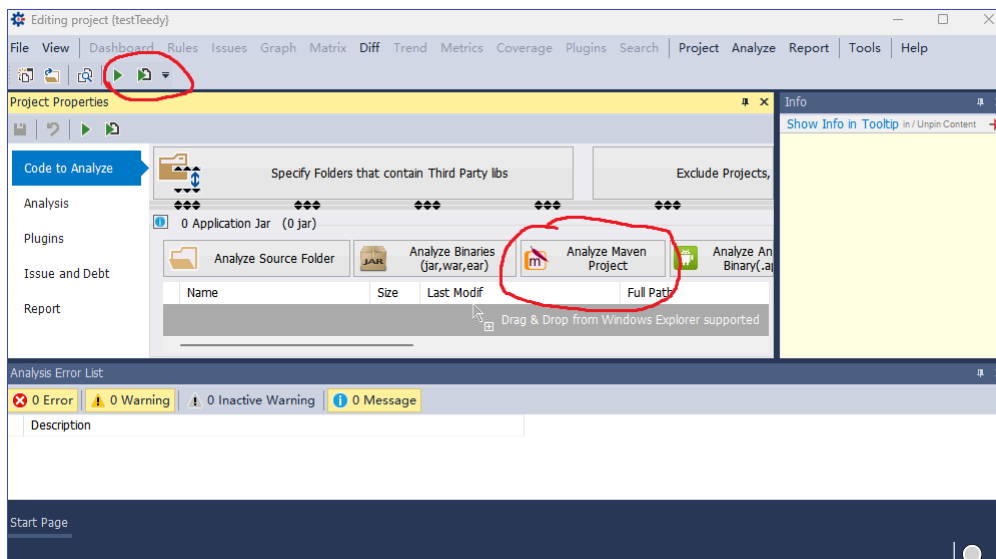
In the above figure, click "new project" and select when you want to put your "JArchitect" project.



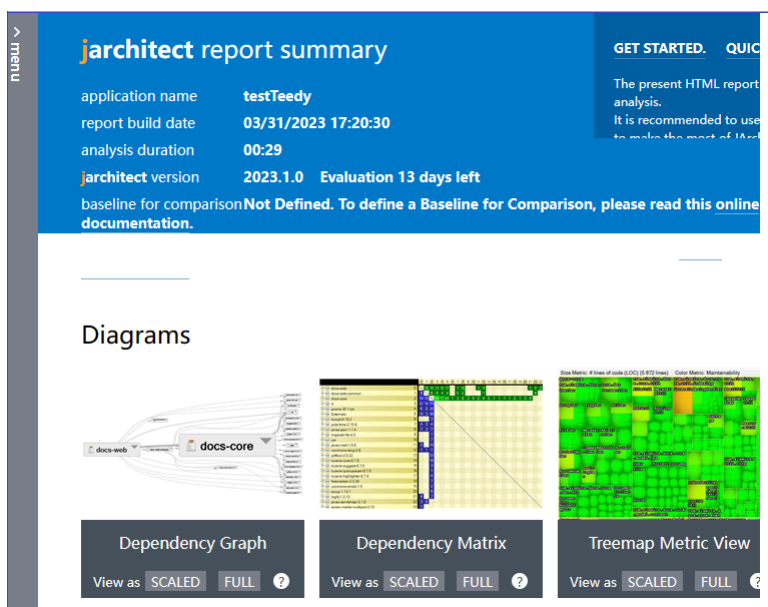
We are not asking JArchitect to create a maven project. We are asking it to create a "JArchitect project". When the project is created, we can see the "jdproj" file instead of the "pom.xml" file in this directory:



Now we can start to analyse our "Teedy" project. Just click "Analyse Maven Project" and select our Teedy project's pom:



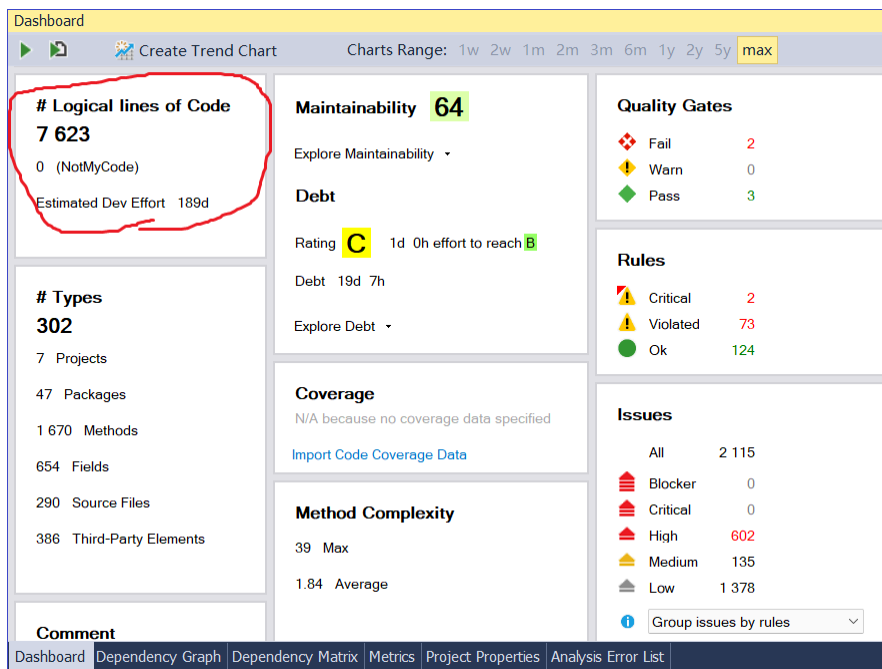
In the above figure, find the green triangle there and run it. The running time may depends on the size of your project. After the project finished running, the report may show up in your default browser:



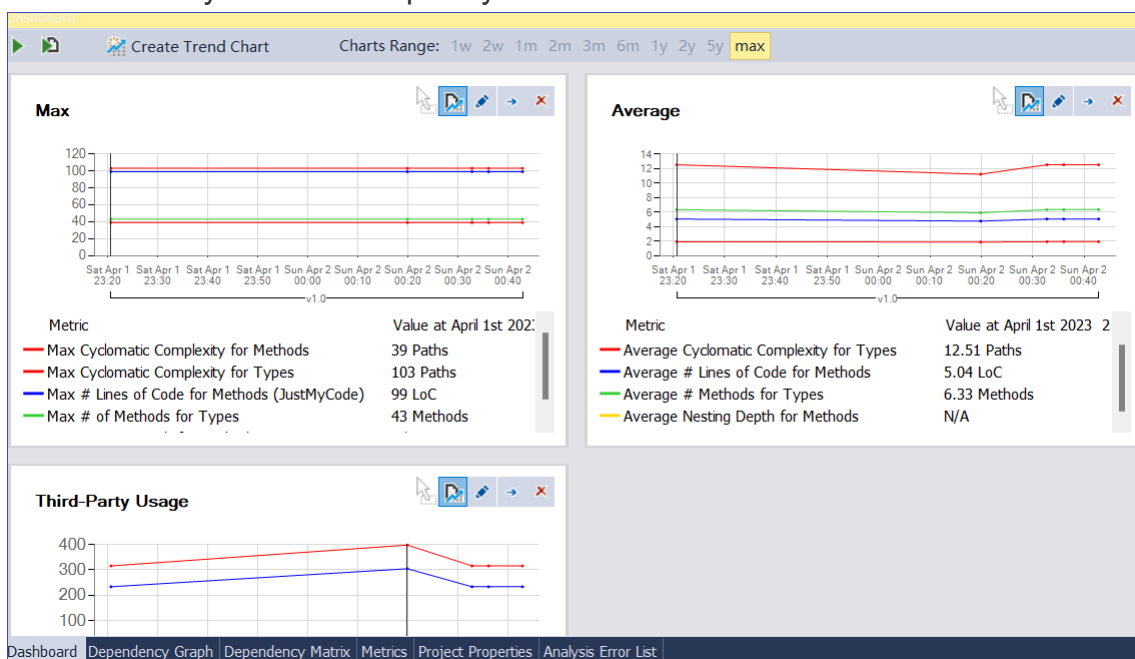
## 2.2 Viewing JArchitect report summary

Let's view some of the informations JArchitect provided for us.

On the "Dashboard" section the you can see many statistics of this project. Including the line of code:



The trend of cyclomatic complexity:

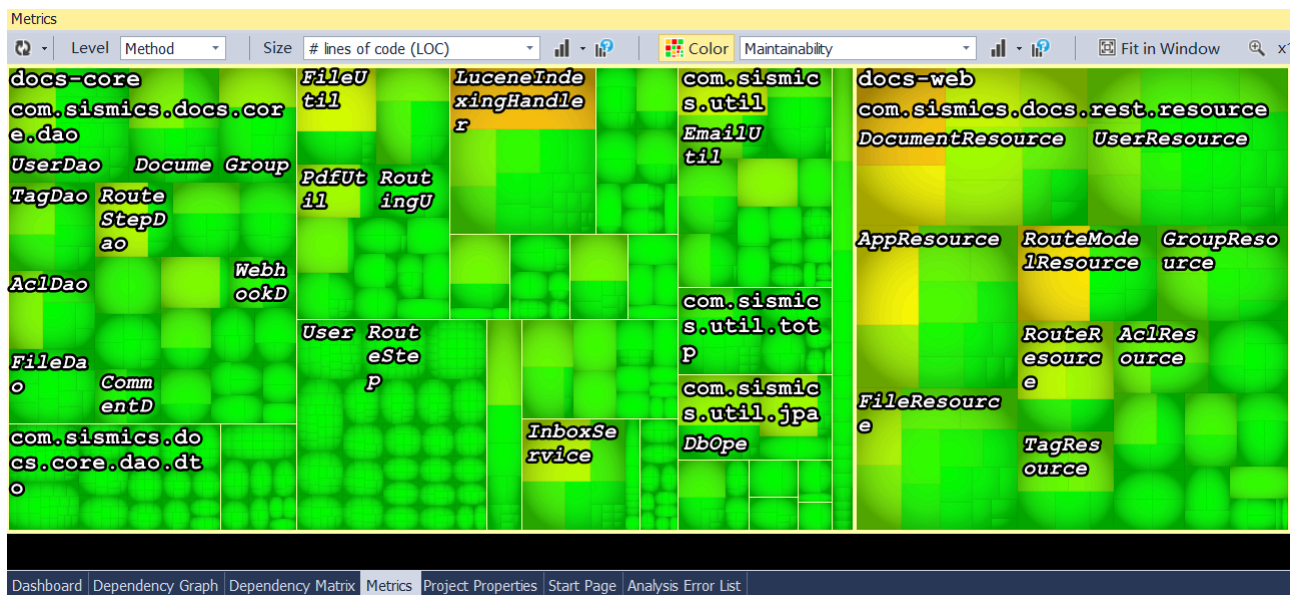


JArchitect also shows you see potential problems of this project, such as "some classes contains too many methods". The details can be explored with the gui.

Queries and Rules Explorer			Create Query	Delete Query	Project Rules
<ul style="list-style-type: none"> <li>Project Rules (395 queries) <ul style="list-style-type: none"> <li>Quality Gates (14 queries) <ul style="list-style-type: none"> <li>Code Smell (8 queries)</li> <li>Code Smells Regression (9 queries)</li> <li>Object Oriented Design (4 queries)</li> <li>Architecture (6 queries)</li> <li>Design (59 queries)</li> <li>Import Statements (6 queries)</li> <li>Optimization (11 queries)</li> <li>Strict Exceptions (12 queries)</li> <li>String and StringBuffer (15 queries)</li> <li>API Best Practices (9 queries)</li> </ul> </li> </ul> </li> </ul>					
			Active	#Items	Code Queries and Rules
			🔴	5	Avoid types too big
			🔴	2	Avoid types with too many methods
			🟢	0	Avoid types with too many fields
			🟢	0	Avoid methods too big, too complex
			🔴	6	Avoid methods with too many parameters
			🔴	7	Avoid methods with too many local variables
			🔴	31	Avoid methods potentially poorly commented
			🔴	8	Avoid types with poor cohesion
					Project Rules \ Code Smells Regression

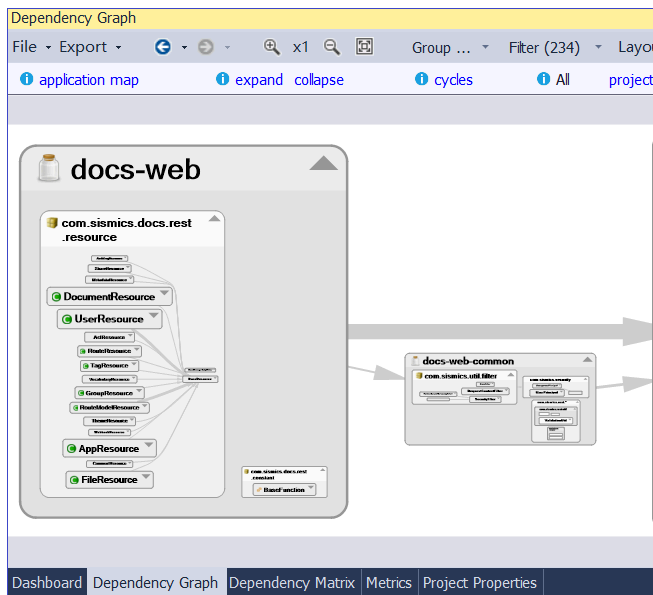
## 2.3 View metrics for each method

You can also view the metrics of each method in the "metrics" view:



## 2.4 View dependency

The dependency graph shows us the dependency relationship between the modules and classes:



This dependency graph is intuitive but not friendly for counting statistics, so their is another view called "dependency matrix":

The Dependency Matrix view displays a table of dependency counts between modules. The table has columns for the source module, the target module, and the count of dependencies. The data is as follows:

Source Module	Target Module	Count
docs-web	docs-web-common	26
docs-web-common	docs-core	66
docs-core	rt	19
rt	Externals	0
Externals	jce	11

The bottom navigation bar includes: Dashboard, Dependency Graph, Dependency Matrix, Metrics, Project Properties.



The matrix shows the coupling of between components of this project. The numbers on the graph shows the number of members involved in the coupling.

## Summary

There are more functions in these tools than this introduction. It is encouraged to explore those functions either through the gui of this software or using the documentation on the official website:

<https://docs.sonarsource.com/sonarqube-cloud>

<https://www.jarchitect.com/GettingStarted>.