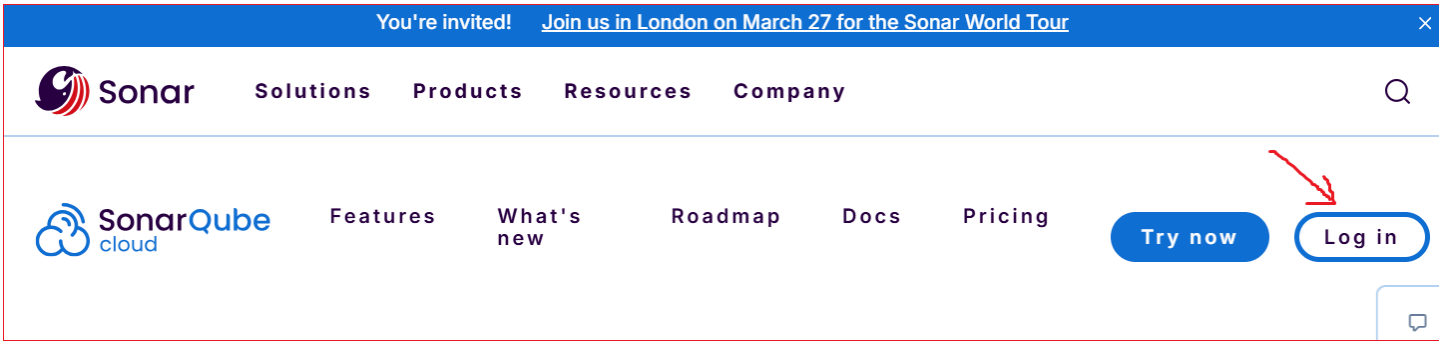
# [CS304] Tutorial 7 - Linters and Code Review Tools

## 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.

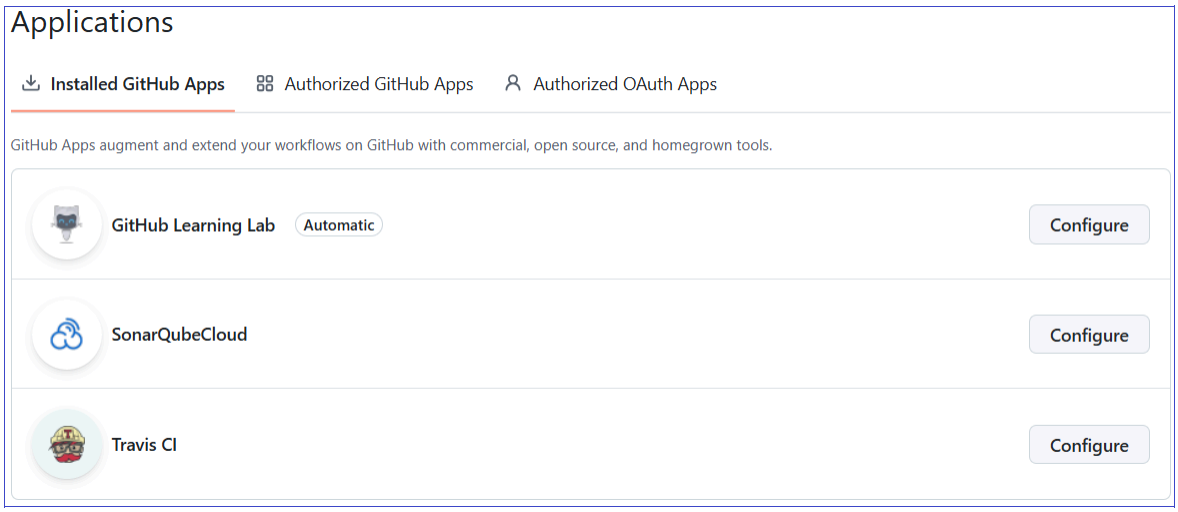
You only need to follow the steps.

### 1.2 Import an organization from GitHub

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

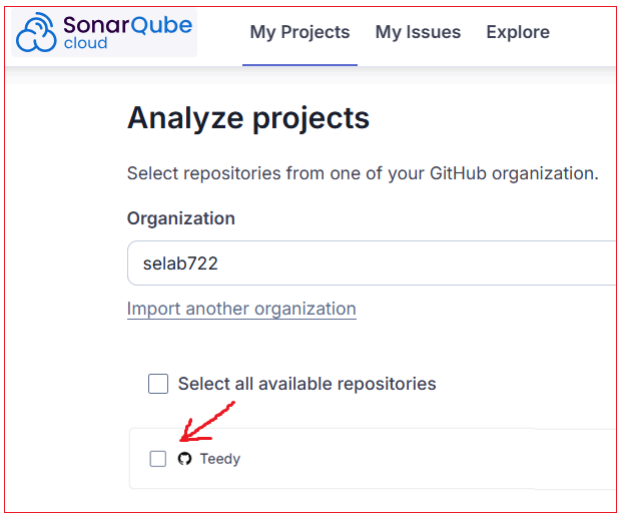
Choose "Free" plan and follow the steps.

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




## 1.3 Import your repo to analyse

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




## 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 with take some time to analyze code and produce the report. The report contains many statistics and recommandations 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 Code | Security | Reliability | Maintainability | Security Hotspots | Coverage | Duplications |
|---|---|---|---|---|---|---|---|
| Teedy25 | | | | | | | |
| 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% |

Measures Code Activity

Project Overview

Security

Overview

Overall Code

Vulnerabilities 5

Rating D

Remediation Effort 1h

Reliability

Maintainability

Security Review

Duplications

Size

Complexity

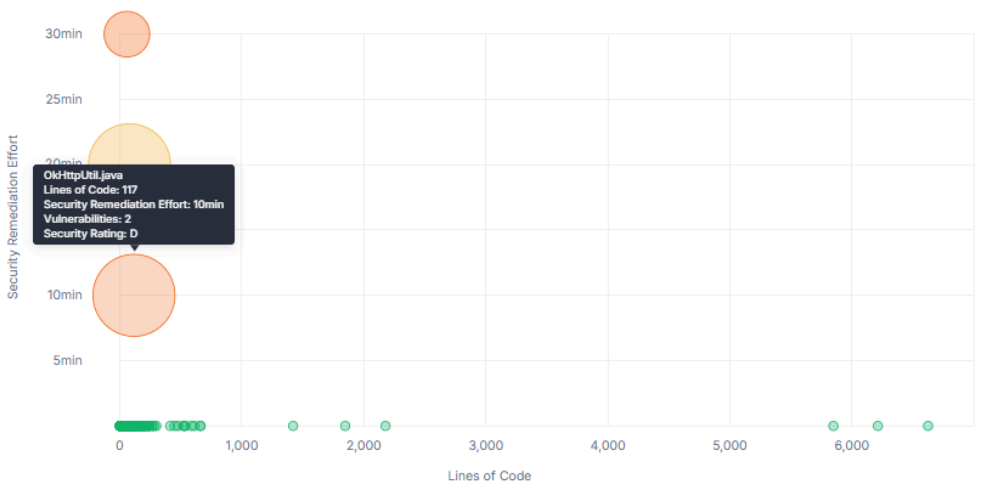
Issues

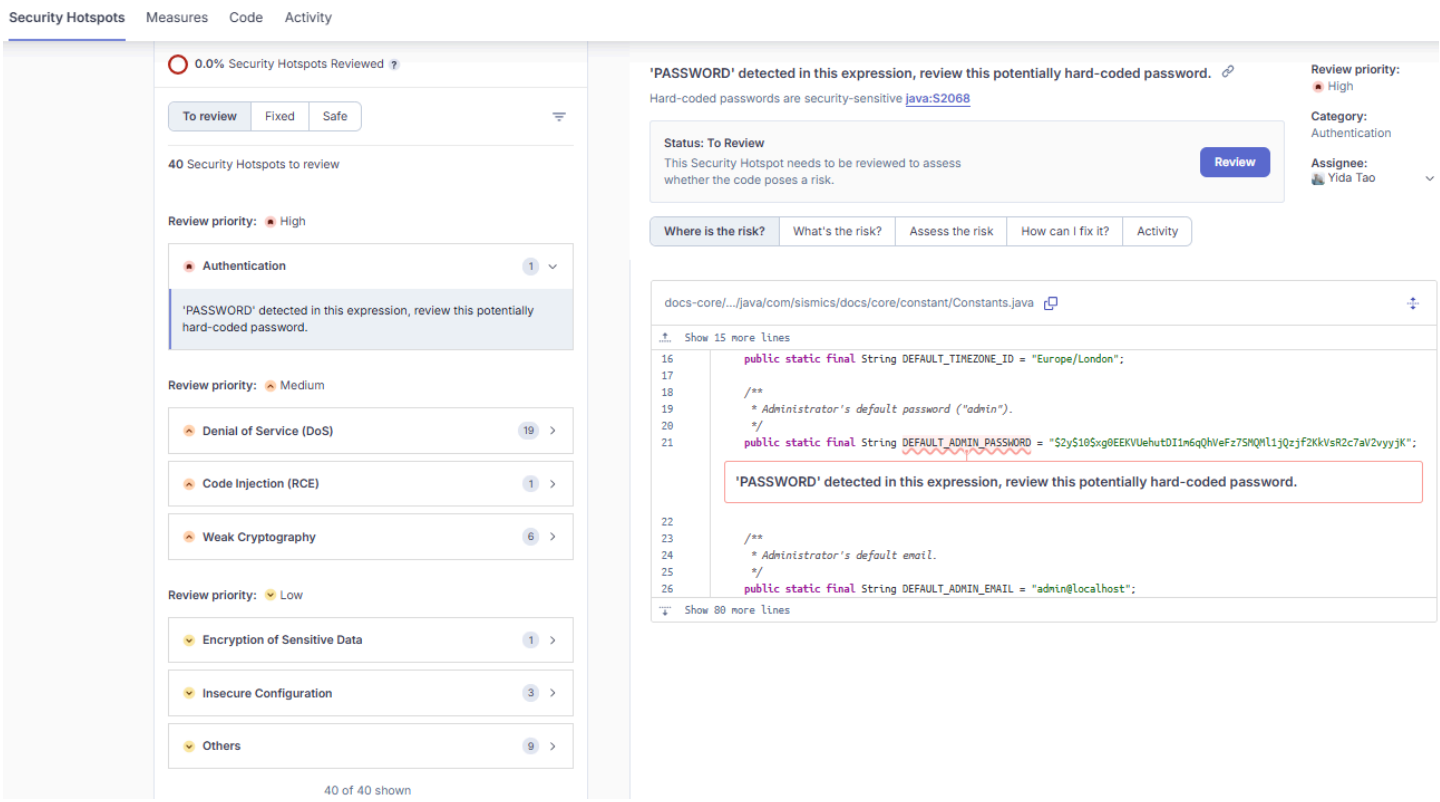
Teedy25 **500/571** files

Security Overview

**Size:** Vulnerabilities **Color:** Security Rating A B C D E

**Zoom:** 100%





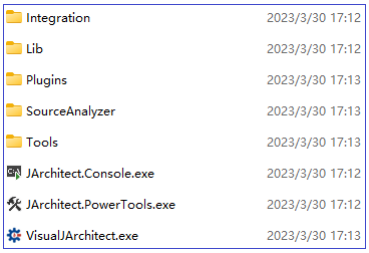

# 2. JArchitect

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.

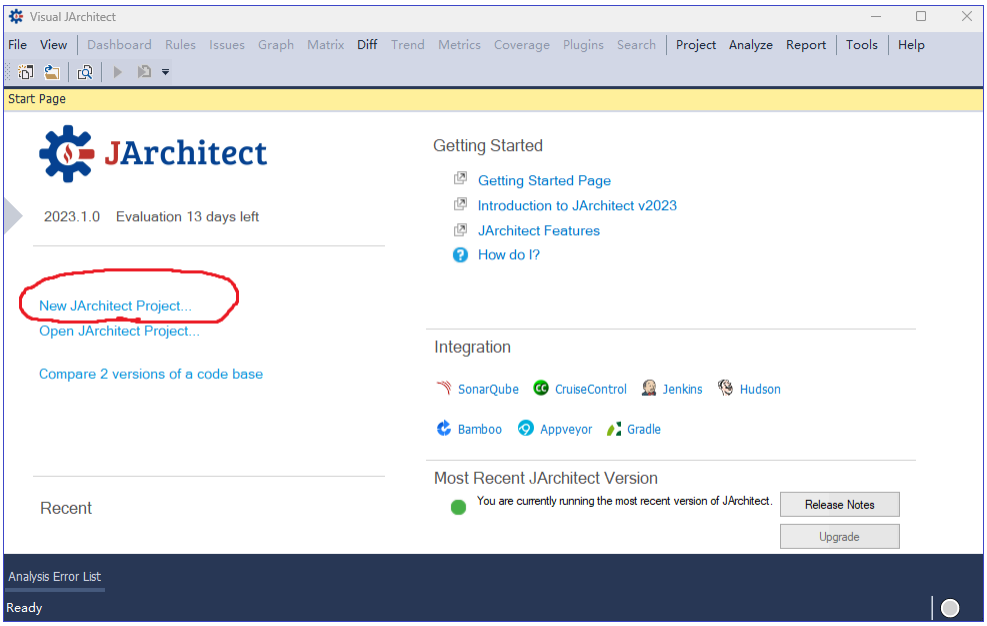
It is a non-free software and you can try it free for a few days:
https://www.jarchitect.com/download.

## 2.1 Getting started

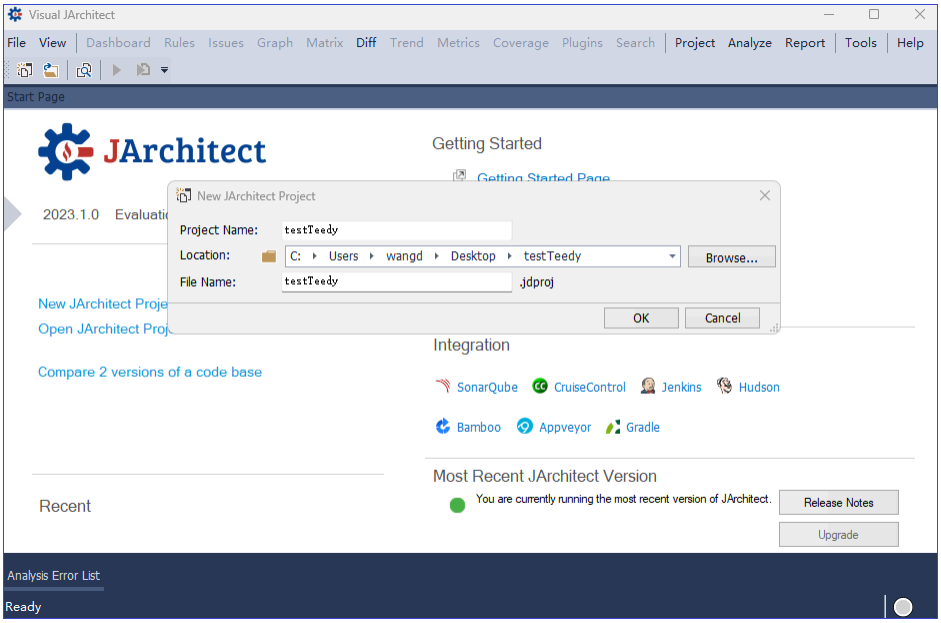
To start with JArchitect, first download the package and uncompress it:



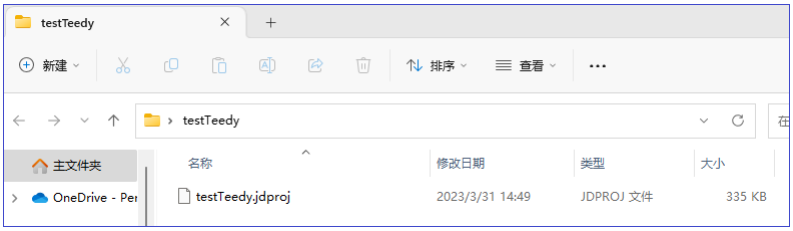

Then double click "VisualJArchitect.exe" to run the gui of JArchitect:



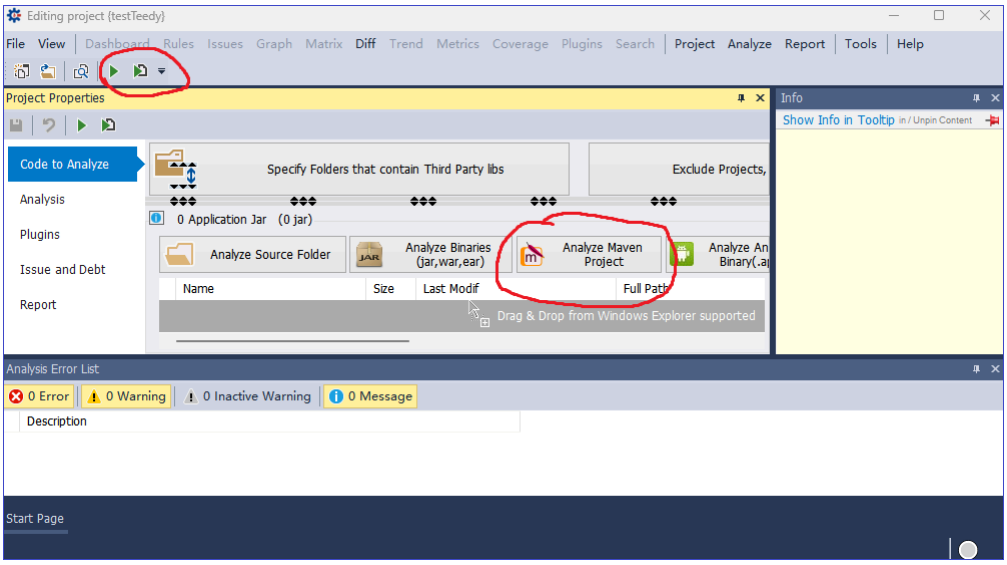

In the above figure, click "new project" and select when you want to put your "JArchitect" project.



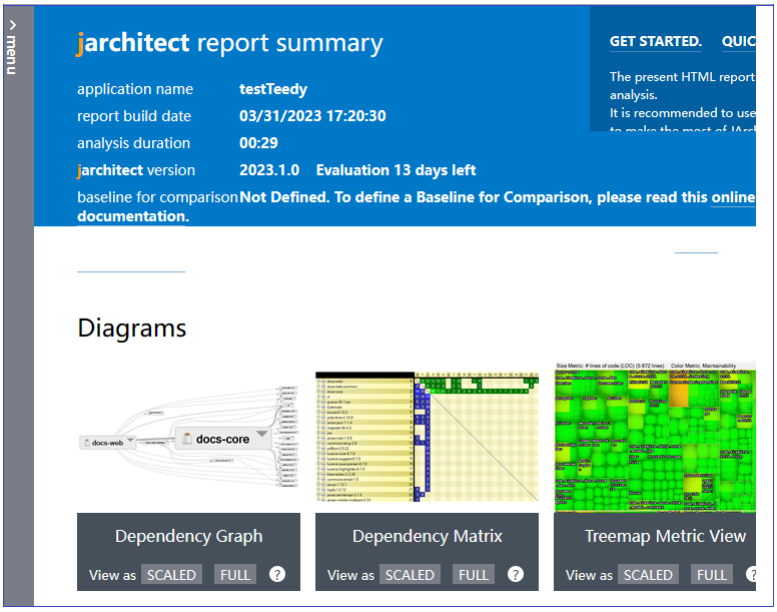

We are not asking JArchitect to create a maven project. We are asking it to create a "JArchitect project". When the project is created, we can see the "jdproj" file instead of the "pom.xml" file in this directory:




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



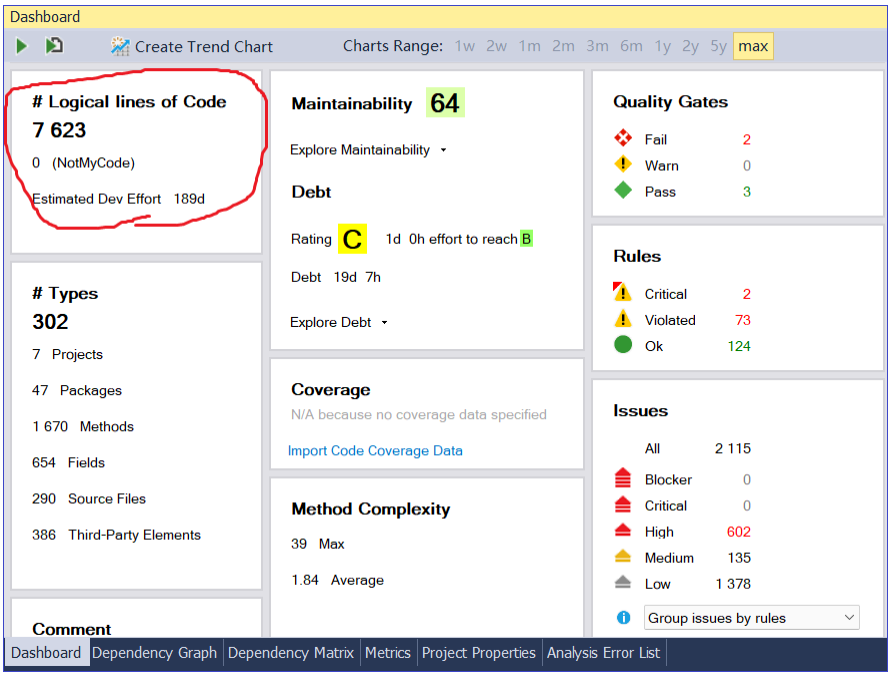

In the above figure, find the green triangle there and run it. The running time may depends on the size of your project. After the project finished running, the report may show up in your default browser:



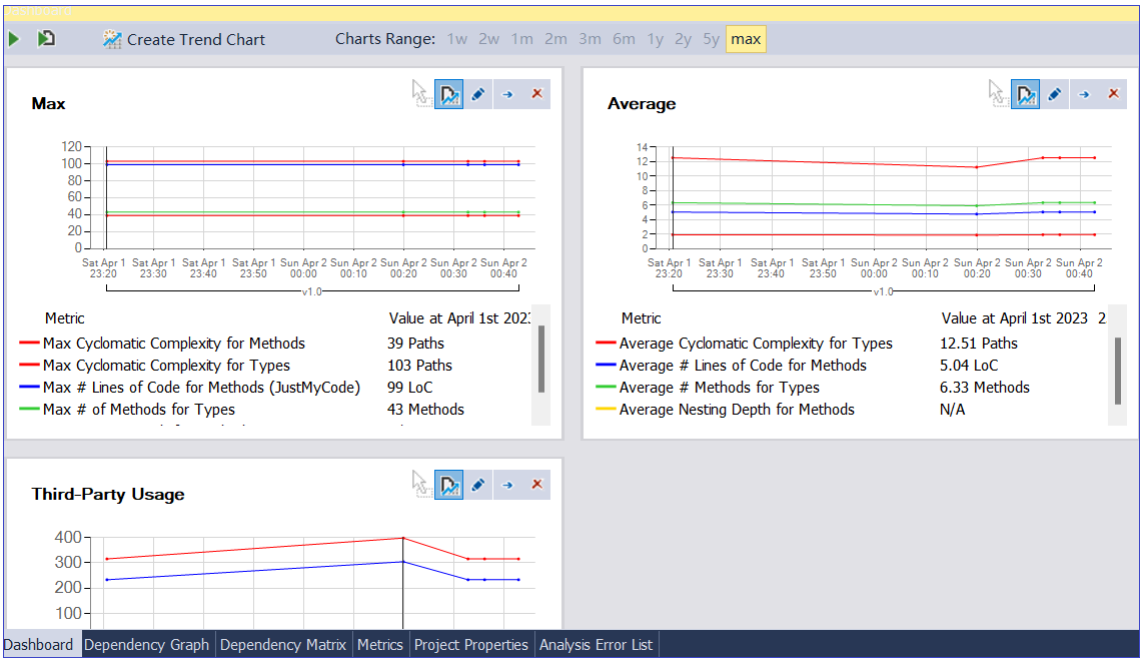

## 2.2 Viewing JArchitech report summary

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

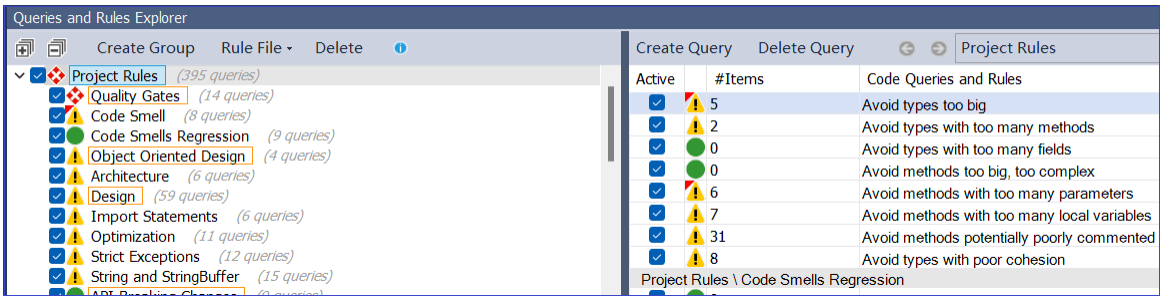
On the "Dashboard" section the you can see many statistics of this project. Including the line of code:




The trend of cyclomatic complexity:

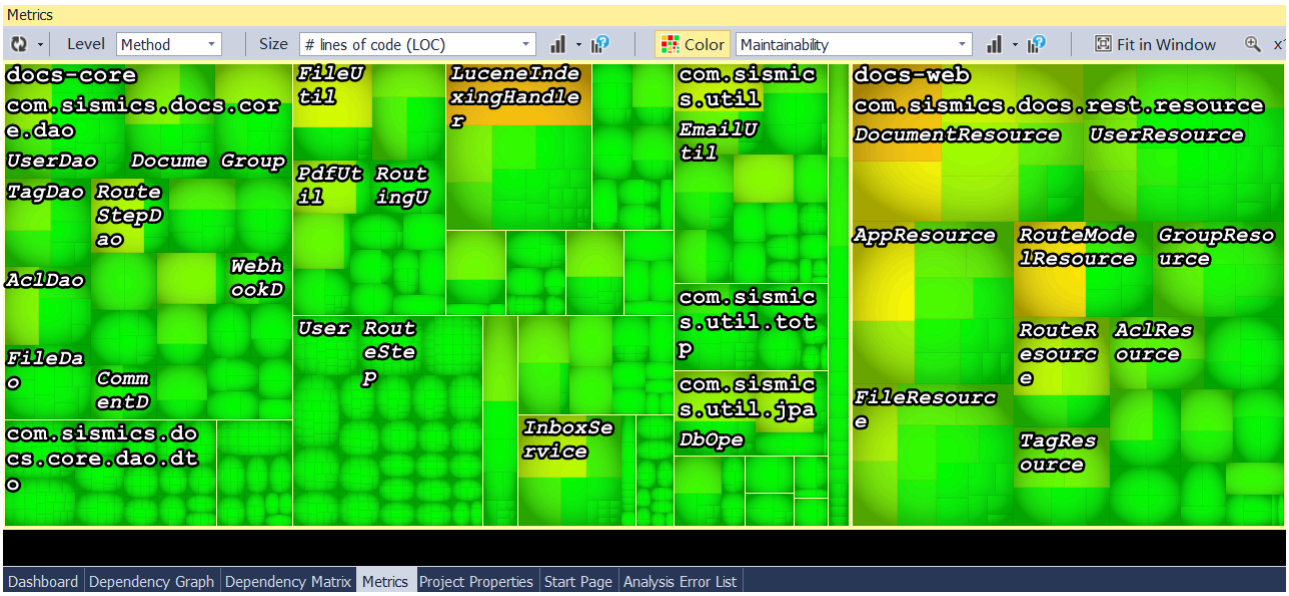



JArchitect also shows you see potential problems of this project, such as "some classes contains too many methods". The details can be explored with the gui.



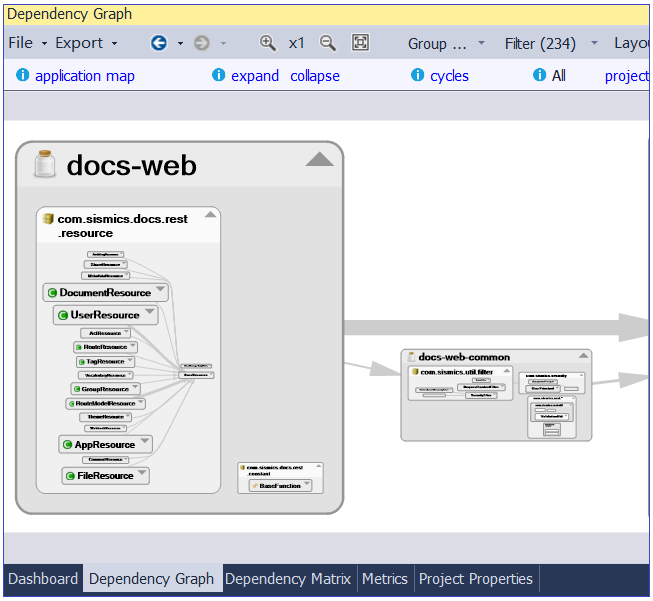

# 2.3 View metrics for each method

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

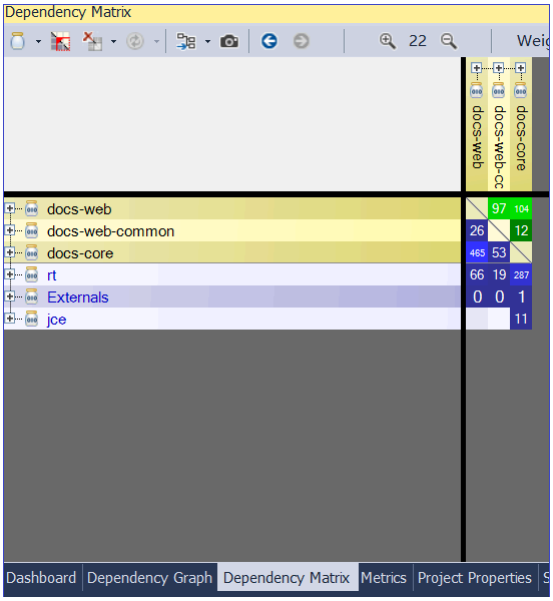



# 2.4 View dependency

The dependency graph shows us the depency 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 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.