

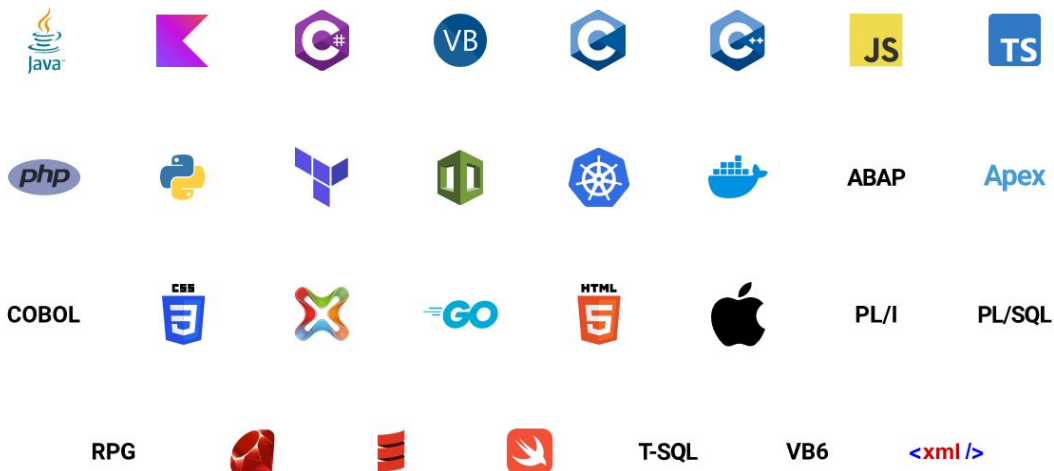
The logo for SonarQube, featuring the word "sonarqube" in a white, lowercase, sans-serif font. To the right of the text are three concentric, light blue curved lines representing sonar waves. The logo is centered on a dark blue rectangular background that has a subtle pattern of light blue shapes, including a circle and several stylized insects (beetles or crickets).

sonarqube

*Kacper Machnik
Tomasz Madeja*

Czym jest SonarQube?

SonarQube jest platformą open source opracowaną przez SonarSource do ciągłej kontroli jakości kodu w celu przeprowadzania automatycznych przeglądów ze statyczną analizą kodu w celu wykrycia błędów i zapachów kodu* w ponad 30 językach programowania i framework'ach SonarQube oferuje raporty dotyczące zduplikowanego kodu, standardów kodowania, testów jednostkowych, pokrycia kodu testami, złożoności kodu, komentarzy, błędów i zaleceń dotyczących bezpieczeństwa.



*Zapach kodu (ang. code smell)

Jak działa SonarQube?

Najprościej rzecz ujmując w wyniku przeprowadzonej analizy otrzymujemy ogólną miarę jakości oraz listę przypadków, w których zdefiniowane reguły zostały złamane. Jedną z najbardziej użytecznych z punktu widzenia programisty funkcjonalności opisywanej platformy jest Sonarlint – rozszerzenie dostępne dla wszystkich popularnych IDE – pozwalające na wykrywanie nieprawidłowości już w momencie pisania kodu. W celu integracji z IntelliJ IDEA instalujemy plugin Sonarlint.

The logo for Sonarlint, featuring the word "sonarlint" in a bold, dark blue, sans-serif font. Below the text is a red, wavy line that resembles a sonar wave or a stylized underline.

sonarlint

Instalacja środowiska (dla systemu Windows):

1. Instalacja JDK 17 oraz dodanie do zmiennej środowiskowej PATH

C:\Program Files\Java\jdk-17\bin

2. Instalacja Maven oraz dodanie do zmiennej środowiskowej PATH

C:\\$MAVEN_HOME\$\apache-maven-3.9.5-bin\apache-maven-3.9.5\bin

3. Pobranie SonarQube: (<https://www.sonarqube.org/downloads>)

4. Włączenie serwera SonarQube

C:\\$SONARQUBE_HOME\$\sonarqube-10.2.1.78527\bin\windows-x86-64\StartSonar.bat

5. Pobranie Sonar Scanner (<https://docs.sonarqube.org/latest/analysis/scan/sonarscanner/>)

6. Dodanie go do zmiennej środowiskowej PATH

C:\\$SONAR_SCANNER_HOME\$\sonar-scanner-cli-5.0.1.3006-windows\sonar-scanner-5.0.1.3006-windows\bin

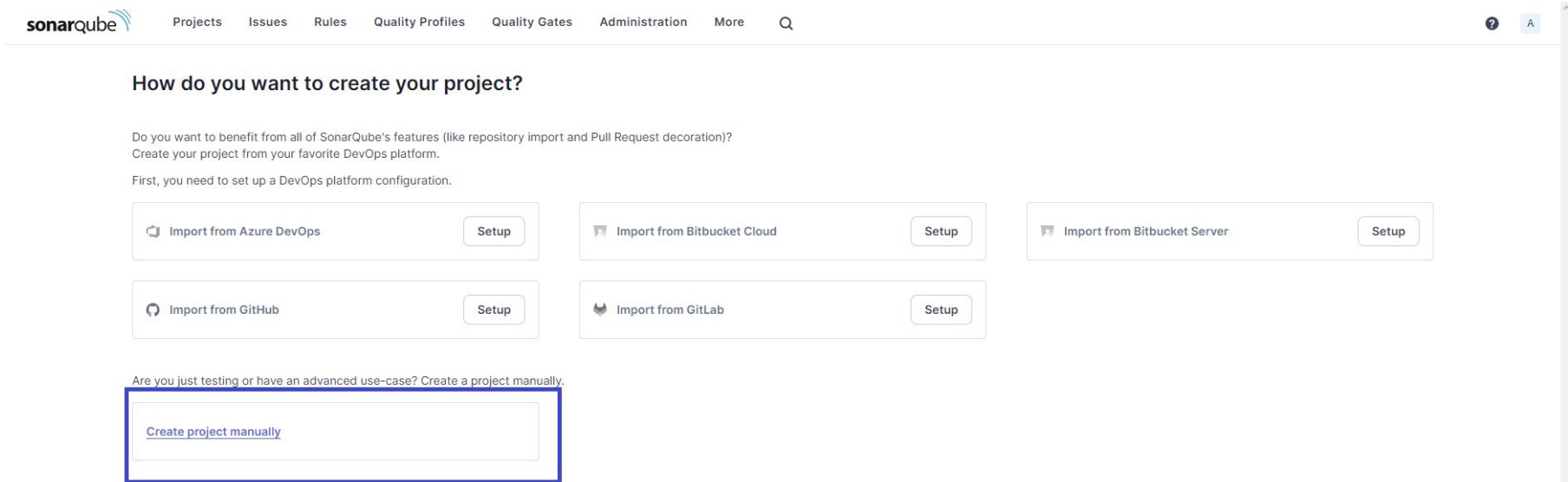
7. Logowanie na (<http://localhost:9000>) Login: admin | Password: admin

8. Zmiana hasła na inne



Generacja pierwszego raportu

1. Tworzenie nowego projektu.



sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More Q

How do you want to create your project?

Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)?
Create your project from your favorite DevOps platform.

First, you need to set up a DevOps platform configuration.

Import from Azure DevOps Setup

Import from Bitbucket Cloud Setup

Import from Bitbucket Server Setup

Import from GitHub Setup

Import from GitLab Setup

Are you just testing or have an advanced use-case? Create a project manually.

Create project manually

2. Nazwanie projektu oraz utworzenie klucza projektu

 [Projects](#) [Issues](#) [Rules](#) [Quality Profiles](#) [Quality Gates](#) [Administration](#) [More](#)

?

A

Create a project

Project display name *



Up to 255 characters. Some scanners might override the value you provide.

Project key *



The project key is a unique identifier for your project. It may contain up to 400 characters.
Allowed characters are alphanumeric, '-' (dash), '_' (underscore), '.' (period) and ':' (colon), with at least one non-digit.

Main branch name *

The name of your project's default branch [Learn More](#) 

Next

3. Wybór ustawień projektu

[Projects](#)[Issues](#)[Rules](#)[Quality Profiles](#)[Quality Gates](#)[Administration](#)[More](#)

Set up project for Clean as You Code

The new code definition sets which part of your code will be considered new code. This helps you focus attention on the most recent changes to your project, enabling you to follow the Clean as You Code methodology. Learn more: [Defining New Code](#)

Choose the baseline for new code for this project

☒ Use the global setting

Previous version

Any code that has changed since the previous version is considered new code.
Recommended for projects following regular versions or releases.

☐ Define a specific setting for this project

☐ Previous version

Any code that has changed since the previous version is considered new code.
Recommended for projects following regular versions or releases.

☐ Number of days

Any code that has changed in the last x days is considered new code. If no action is taken on a new issue after x days, this issue will become part of the overall code.
Recommended for projects following continuous delivery.

☐ Reference branch

Choose a branch as the baseline for the new code.
Recommended for projects using feature branches.

Create project

4. Wybór metody analizy projektu

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration more

Congratulations! Your project has been created. X

test / main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Analysis Method

Use this page to manage and set-up the way your analyses are performed.

How do you want to analyze your repository?

 [With Jenkins](#)

 [With GitHub Actions](#)

 [With Bitbucket Pipelines](#)

 [With GitLab CI](#)

 [With Azure Pipelines](#)

[Other CI](#)

SonarQube integrates with your workflow no matter which CI tool you're using.

[Locally](#)

Use this for testing or advanced use-case. Other modes are recommended to help you set up your CI environment.

5. Generowanie tokenu

**możliwe jest również użycie wcześniej wygenerowanego tokenu*



ProjectsIssuesRulesQuality ProfilesQuality GatesAdministrationMore

?

A

☆ test / ⓘ main ?

Project Settings ▾Project Information

OverviewIssuesSecurity HotspotsMeasuresCodeActivity

Analysis Method > Locally

Analyze your project

We initialized your project on SonarQube, now it's up to you to launch analyses!

1 Provide a token

Generate a project token

Token name ?

Analyze "test"

Expires in

30 days ▾

Generate

ⓘ

Please note that this token will only allow you to analyze the current project. If you want to use the same token to analyze multiple projects, you need to generate a global token in your [user account](#). See the [documentation](#) [🔗](#) for more information.

The token is used to identify you when an analysis is performed. If it has been compromised, you can revoke it at any point in time in your [user account](#).

2 Run analysis on your project

6. Generacja raportu przy użyciu Mavena



ProjectsIssuesRulesQuality ProfilesQuality GatesAdministrationMore

test / main

OverviewIssuesSecurity HotspotsMeasuresCodeActivity

Project SettingsProject Information

Analysis Method > Locally

Analyze your project

We initialized your project on SonarQube, now it's up to you to launch analyses!

1 Provide a token

Analyze "test": `sqp_1a98e7772b26fa718c91213d2ec2f2c7dcaf83d3`

2 Run analysis on your project

What option best describes your build?

MavenGradle.NETOther (for JS, TS, Go, Python, PHP, ...)

Execute the Scanner for Maven

Running a SonarQube analysis with Maven is straightforward. You just need to run the following command in your project's folder.

```
mvn clean verify sonar:sonar \
-Dsonar.projectKey=test \
-Dsonar.projectName=test \
-Dsonar.host.url=http://localhost:9000 \
-Dsonar.token=sqp_1a98e7772b26fa718c91213d2ec2f2c7dcaf83d3
```

Copy

Please visit the [official documentation of the Scanner for Maven](#) for more details.



Uwaga!

Aby komenda zadziałała prawidłowo musimy usunąć “\” oraz dodać spację pomiędzy “-D”, a “sonar”

Execute the Scanner for Maven

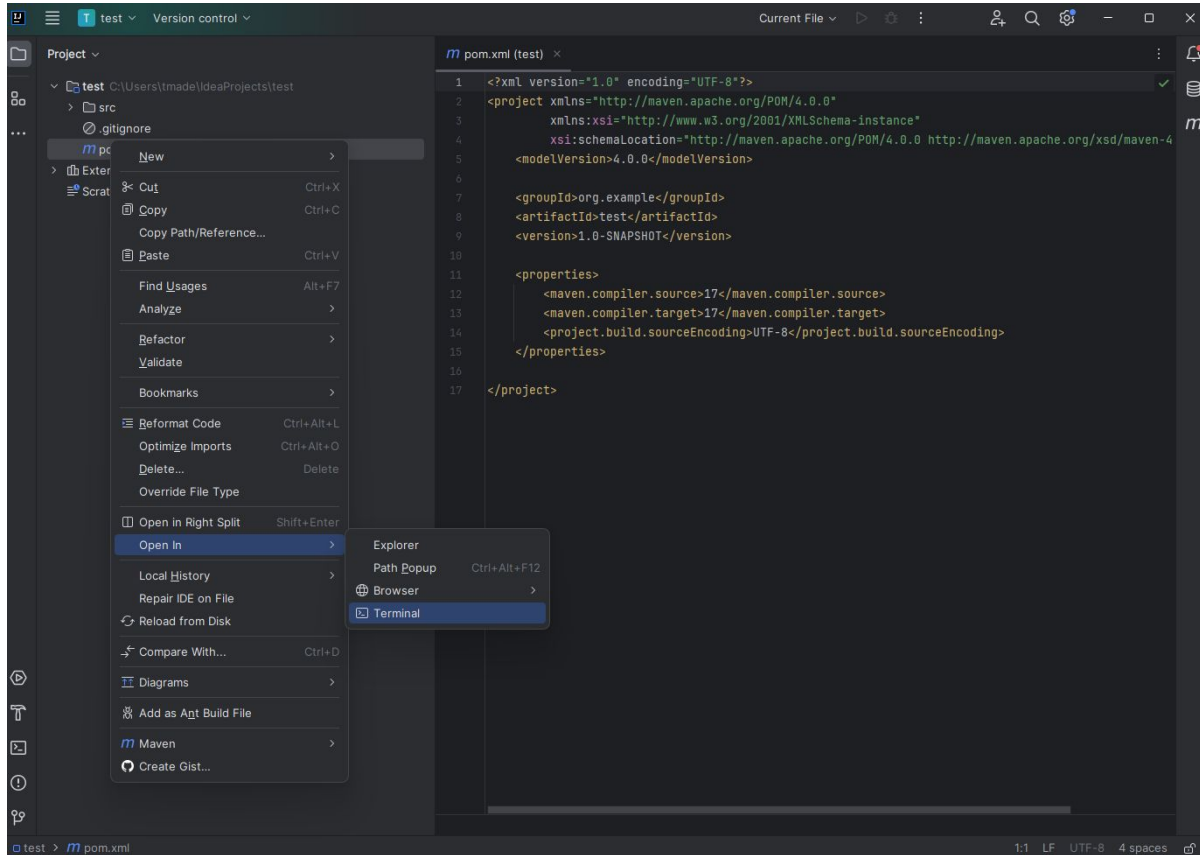
Running a SonarQube analysis with Maven is straightforward. You j

```
mvn clean verify sonar:sonar \  
-Dsonar.projectKey=test \  
-Dsonar.projectName='test' \  
-Dsonar.host.url=http://localhost:9000 \  
-Dsonar.token=sqp_1a98e7772b26fa718c91213d2ec2f2c7dcaf83d3
```

Prawidłowa komenda powinna wyglądać następująco:

```
mvn clean verify sonar:sonar -D sonar.projectKey=YOUR_projectKey -D sonar.projectName='YOUR_projectName' -D sonar.host.url=YOUR_host.url -D sonar.token=YOUR_token
```

7. Wpisanie komendy w terminalu otwartym w projekcie



8. Jeżeli wszystkie kroki zostały wykonane poprawnie wywołane komendy powinno zakończyć się sukcesem

```
Terminal  Local x + v
[INFO] Analysis report uploaded in 14ms
[INFO] ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=test
[INFO] Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
[INFO] More about the report processing at http://localhost:9000/api/ce/task?id=AYtYsxTDndyiiSeTbNps
[INFO] Analysis total time: 4.220 s
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 6.413 s
[INFO] Finished at: 2023-10-22T20:43:33+02:00
[INFO] -----
PS C:\Users\tmade\IdeaProjects\test>
```

Raport

 Projects Issues Rules Quality Profiles Quality Gates Administration More  

☆ test /  main   ?

The last analysis has warnings. [See details](#) Version 1.0-SNAPSHOT 

Overview Issues Security Hotspots Measures Code Activity Project Settings ▾ Project Information

 To benefit from more of SonarQube's features, [set up analysis in your favorite CI.](#) 

Quality Gate Status ?

 Quality Gate
Passed



Enjoy your sparkling clean code!

Measures

New Code

Overall Code

 Reliability

0 Bugs 

 Security

0 Vulnerabilities 

Duplications

0.0% Duplications

Duplications on 14 Lines 

0 Duplicated Blocks

 Maintainability

0 Code Smells 

 Security Review

0 Security Hotspots ? 

Activity

Raport



Projects Issues Rules Quality Profiles Quality Gates Administration More Q



☆ test / ⓘ main ❌ ?

⚠ The last analysis has warnings. [See details](#)

Version 1.0-SNAPSHOT 🏠

Overview Issues Security Hotspots Measures Code Activity

Project Settings ▾ Project Information

ℹ To benefit from more of SonarQube's features, set up analysis in your favorite CI.



Quality Gate Status ?



Quality Gate

Failed

1 failed condition



Security Rating on New Code

is worse than A

Fix issues before they fail your Quality Gate with [SonarLint](#) ⓘ in your IDE. Power up with connected mode!

Measures

New Code ⓘ

Overall Code

1 failed condition

🔧 Reliability

0 Bugs



⚙ Maintainability

13 Code Smells



🔒 Security

1 Vulnerabilities



🛡 Security Review

0 Security Hotspots ?



Coverage

0.0% Coverage

Coverage on 314 Lines to cover

– Unit Tests



Duplications

0.0% Duplications

Duplications on 572 Lines

0 Duplicated Blocks



Zasady



Projects Issues **Rules** Quality Profiles Quality Gates Administration More



Filters

Language

Java 626
C# 435
TypeScript 345
JavaScript 339
PHP 253
Python 251
VB.NET 215
Kotlin 133
Flex 76
HTML 65

10 shown [Show More](#)

Clean Code Attribute

Consistency 622
Intentionality 1.7k
Adaptability 248
Responsibility 117

Software Quality

Security 201
Reliability 630
Maintainability 1.8k

Severity

Type

Tag

Bulk Change

1 / 3,078 rules

"!important" should not be used on "keyframes"	Intentionality rule	CSS	Reliability	
"\$this" should not be used in a static context	Intentionality rule	PHP	Reliability	
"&&" and " " should be used	Intentionality rule	PHP	Maintainability	suspicious
".equals()" should not be used to test the values of "Atomic" classes	Intentionality rule	Java	Reliability	multi-threading
"<!DOCTYPE>" declarations should appear before "<html>" tags	Consistency rule	HTML	Reliability	user-experience
"<>" should not be used to test inequality	Consistency rule	Python	Maintainability	obsolete
"<?php" and "<?=" tags should be used	Consistency rule	PHP	Maintainability	convention, psr1
"<fieldset>" tags should contain a "<legend>"	Consistency rule	HTML	Reliability	accessibility
"<frames>" should have a "title" attribute	Consistency rule	HTML	Reliability	accessibility
"<html>" element should have a language attribute	Intentionality rule	HTML	Reliability	accessibility, wcag2-a
" " and "<dt>" item tags should be in " ", " " or "<dl>" container tags	Consistency rule	HTML	Reliability	
"<object>" tags should provide an alternative content	Intentionality rule	HTML	Maintainability	accessibility, wcag2-a
"" and "" tags should be used	Consistency rule	DEPRECATED	HTML	Reliability
"<table>" tags should have a description	Consistency rule	HTML	Reliability	accessibility, wcag2-a
"<th>" tags should have "id" or "scope" attributes	Consistency rule	HTML	Reliability	accessibility, wcag2-a
"<title>" should be present in all pages	Consistency rule	HTML	Reliability	user-experience
"==" and "!=" should not be used when "equals" is overridden	Intentionality rule	Java	Maintainability	cert, cwe, suspicious

Koncepty

Dzieli się na architekuralne i jakościowe. Do tych pierwszych należą: analizator, baza danych, oraz serwer. Natomiast do jakościowych między innymi należy: "Clean Code", "Code Smell" i "Bug". Więcej na ich temat można przeczytać na:

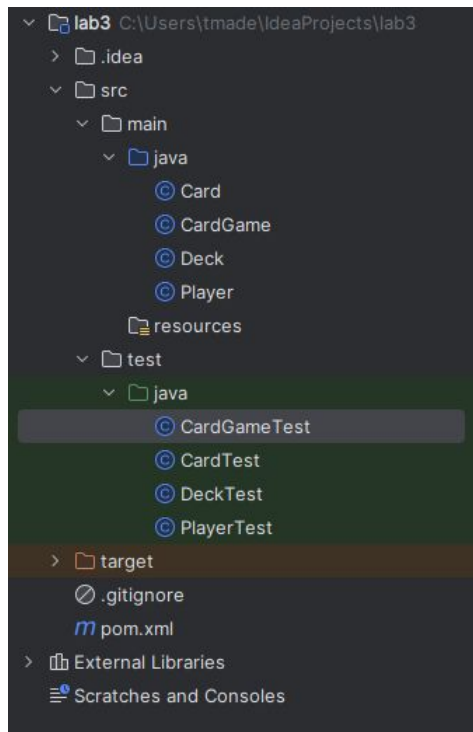
<https://docs.sonarsource.com/sonarqube/latest/user-guide/concepts/>



Pokrycie testami (Coverage)

1. Napisanie testów jednostkowych

Korzystamy z JUnita do napisania testów. Powinny się one znaleźć w zakładce src/test/java. Każda klasa ma odpowiadającą klasę testującą.



2. JaCoCo coverage tool

SonarQube nie generuje raportu pokrycia samodzielnie. Zamiast tego konfigurujemy narzędzie do tworzenia raportu w ramach procesu kompilacji. W przypadku projektów Java, SonarQube bezpośrednio obsługuje narzędzie pokrycia JaCoCo.



Dodanie JaCoCo do projektu Maven

```
<profile>
  <id>coverage</id>
  <build>
    <plugins>
      <plugin>
        <groupId>org.jacoco</groupId>
        <artifactId>jacoco-maven-plugin</artifactId>
        <version>0.8.7</version>
        <executions>
          <execution>
            <id>prepare-agent</id>
            <goals>
              <goal>prepare-agent</goal>
            </goals>
          </execution>
          <execution>
            <id>report</id>
            <goals>
              <goal>report</goal>
            </goals>
            <configuration>
              <formats>
                <format>XML</format>
              </formats>
            </configuration>
          </execution>
        </executions>
      </plugin>
      ...
    </plugins>
  </build>
</profile>
```

Do pliku pom.xml dodajemy odpowiedni profil*,
według wzoru z dokumentacji:

<https://docs.sonarsource.com/sonarqube/9.9/analyzing-source-code/test-coverage/java-test-coverage/>

**Całość ma się znaleźć w sekcji
<profiles></profiles>*

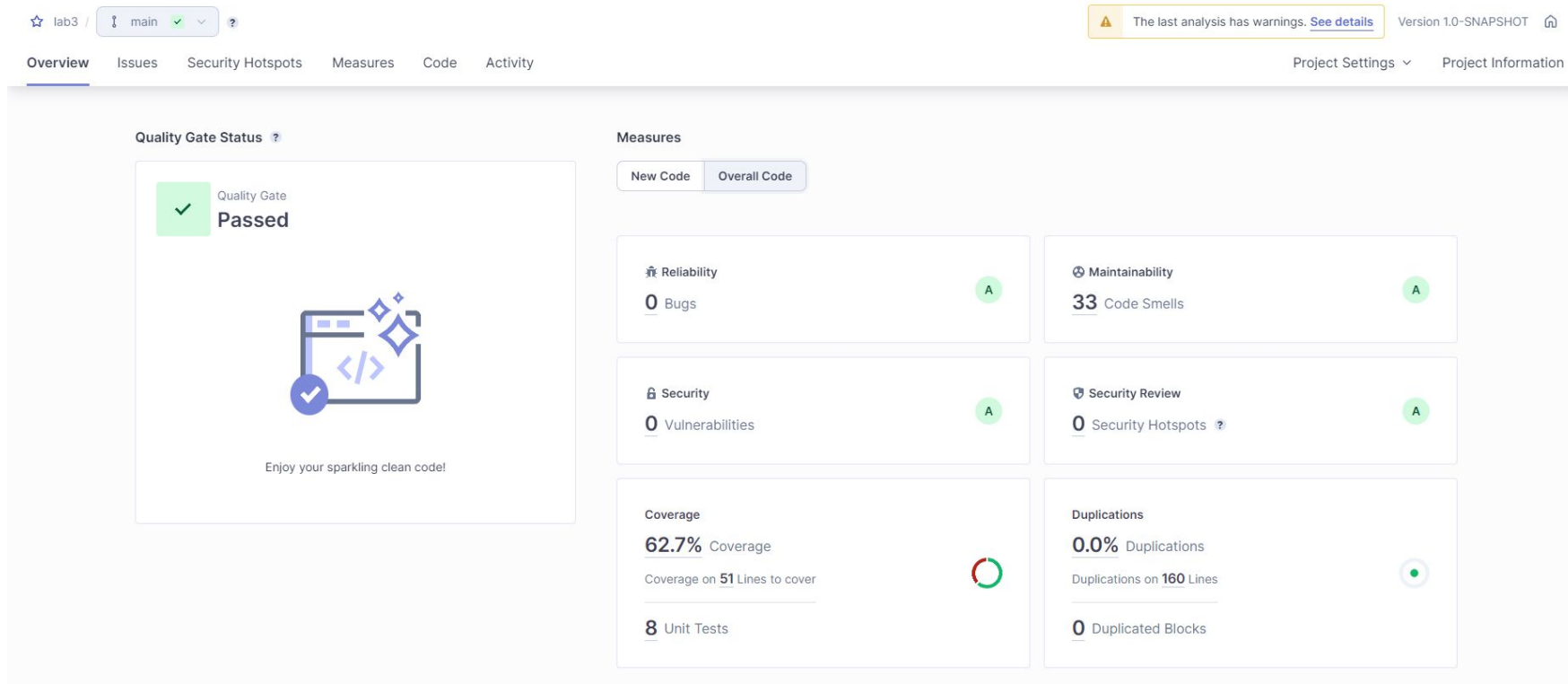
3. Generacja raportu

Aby wygenerować raport wykorzystujemy poprzednio podaną komendę, dopisując do niej **-P coverage** i wklejamy ją do terminala, powinna ona wyglądać następująco:

```
mvn clean verify sonar:sonar -D sonar.projectKey=YOUR_projectKey -D  
sonar.projectName='YOUR_projectName' -D sonar.host.url=YOUR_host.url -D  
sonar.token=YOUR_token -P coverage
```



Raport po napisaniu części testów



Strona pokazuje do jakich części kodu nie ma testów

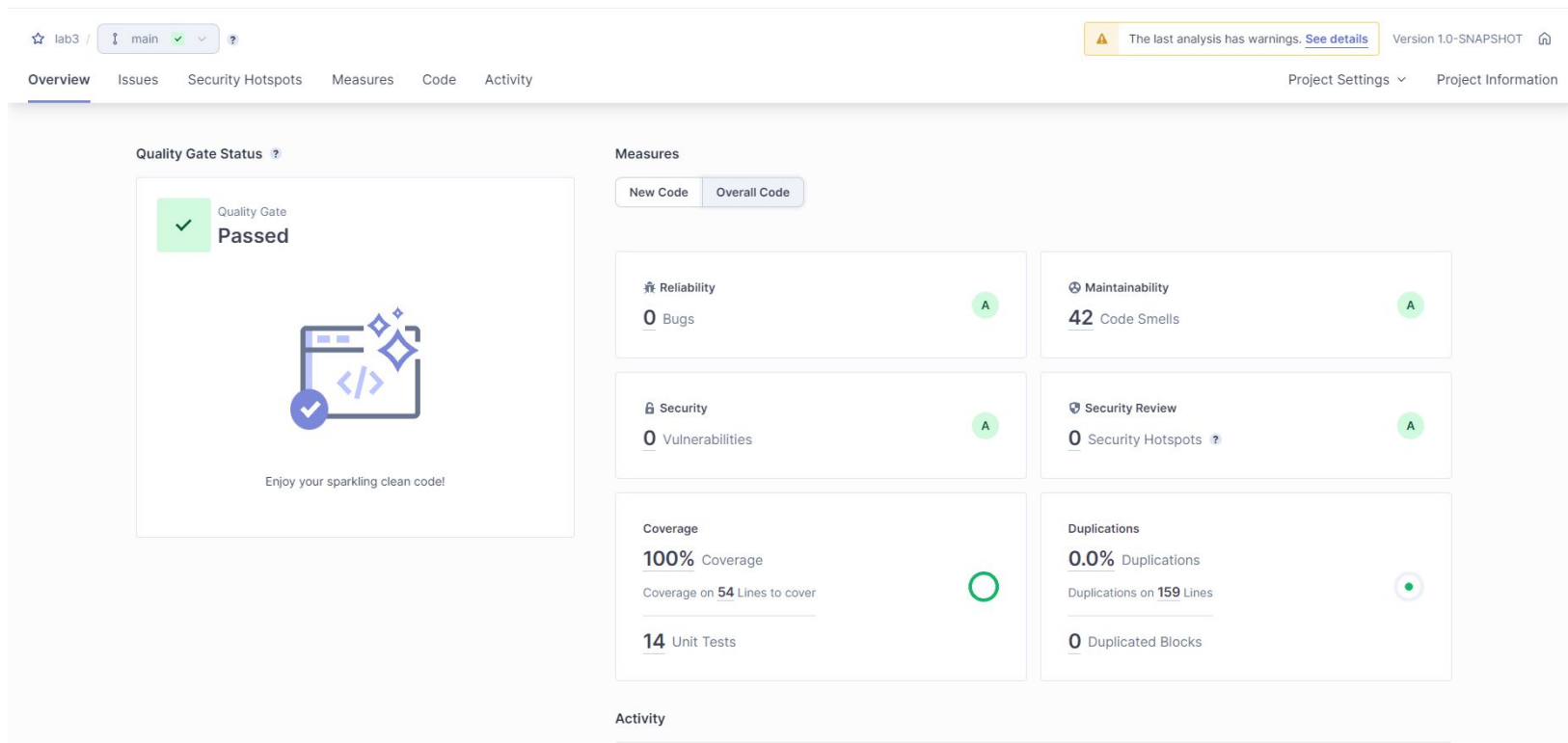
The screenshot displays a web-based code coverage tool interface. At the top, a navigation bar includes a star icon, the text 'lab3 /', a dropdown menu showing 'main' with a green checkmark, and a help icon. On the right side of the navigation bar, a yellow warning box states 'The last analysis has warnings. [See details](#)', followed by 'Version 1.0-SNAPSHOT' and a home icon. Below the navigation bar, a horizontal menu contains 'Overview', 'Issues', 'Security Hotspots', 'Measures' (which is underlined), 'Code', and 'Activity'. On the right side of this menu, there are links for 'Project Settings' and 'Project Information'. The left sidebar contains a list of project overview items: 'Project Overview', 'Reliability', 'Security', 'Security Review', 'Maintainability', and 'Coverage'. The 'Coverage' item is expanded, showing a sub-menu with 'Overview', 'New Code', 'Lines to Cover' (0), 'Uncovered Lines' (0), 'Conditions to Cover' (0), and 'Uncovered Conditions' (0). The main content area shows the file path 'lab3 > src > main/java > CardGame.java' and a code editor. The code editor displays the following Java code:

```
1 import java.util.*;
2
3 public class CardGame {
4     public static void main(String[] args) {
5         Deck shuffledDeck = Deck.shuffle(Deck.FactorySortedDeck());
6
7         Player player1 = new Player("Player1");
8         Player player2 = new Player("Player2");
9         Player player3 = new Player("Player3");
10        Player player4 = new Player("Player4");
11
12        List<Player> players = Arrays.asList(player1, player2, player3, player4);
13
14        for (Player player : players) {
15            player.recvCards(shuffledDeck.deal(5));
16            System.out.println(player);
17        }
18    }
19 }
```

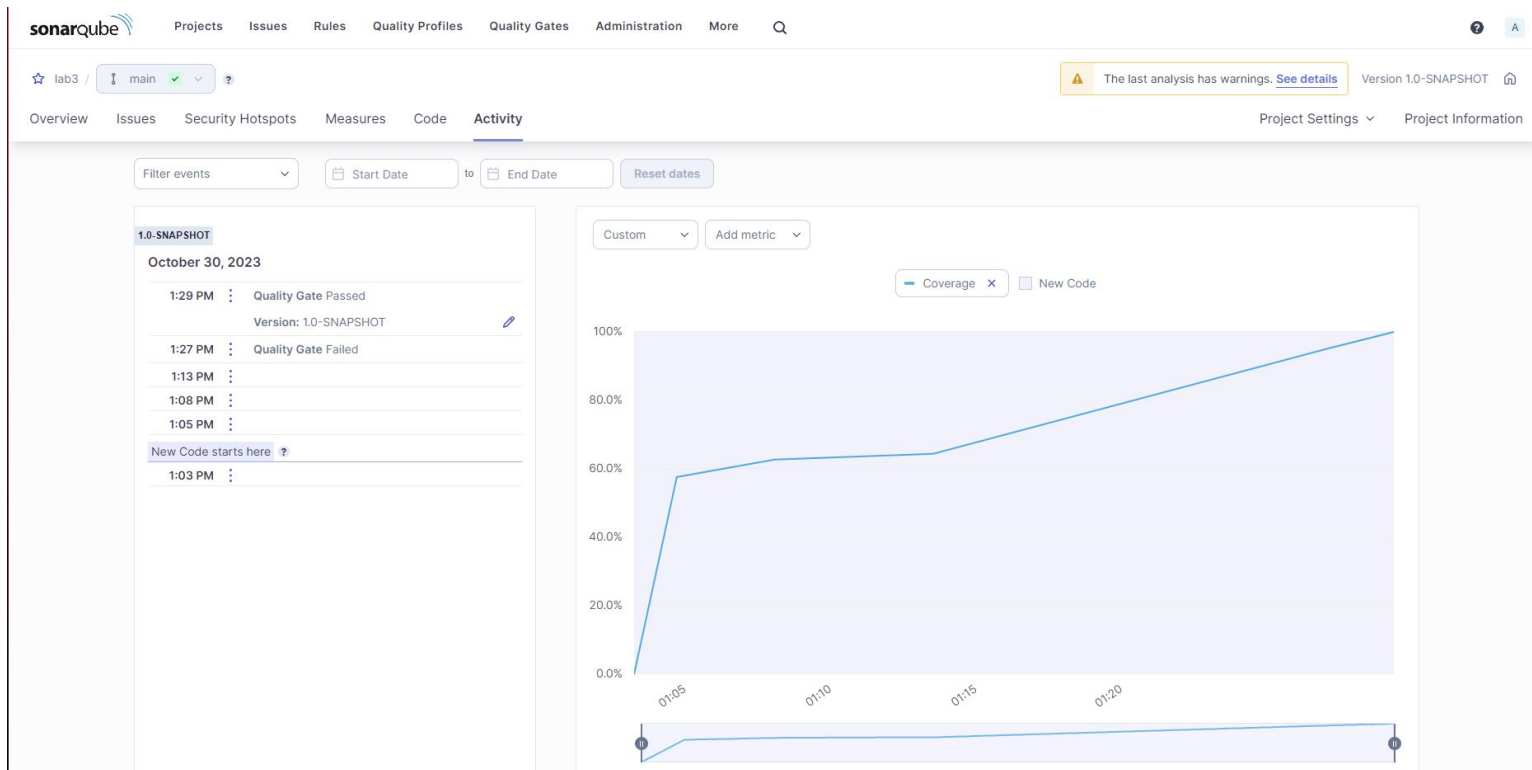
Below the code editor, a table shows the coverage status for different parts of the code:

Measure	Value
Lines to Cover	0
Uncovered Lines	0
Conditions to Cover	0
Uncovered Conditions	0

Po napisaniu testów do każdej funkcji



Wykres pokrycia testami w czasie



Stopień pokrycia testami danych klas

The screenshot displays the SonarQube web interface for project 'lab3'. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, and More. A search icon is also present. A notification banner at the top right states: 'The last analysis has warnings. See details' with a link to 'See details' and 'Version 1.0-SNAPSHOT'.

The left sidebar contains a menu with the following items: Project Overview, Reliability, Security, Security Review, Maintainability, and Coverage. The 'Coverage' item is expanded, showing a sub-menu with 'Overview' and 'New Code'. The 'New Code' section shows a table with the following data:

Metric	Value
Coverage	100%
Lines to Cover	3
Uncovered Lines	0
Line Coverage	100%

The main content area shows the 'Measures' tab for project 'lab3'. It displays a table of test coverage for four files:

File	Coverage	Uncovered Lines	Uncovered Conditions
src/main/java/Card.java	100%	0	100
src/main/java/CardGame.java	100%	0	0
src/main/java/Deck.java	100%	0	0
src/main/java/Player.java	100%	0	100

At the bottom of the table, it indicates '4 of 4 shown'. A 'New Code' button is also visible, showing 'Since October 30, 2023'.

Przydatne linki

- <https://docs.sonarsource.com/sonarqube/latest/user-guide/concepts/>
- <https://www.swtestacademy.com/sonarqube-tutorial/>
- <https://www.sonarsource.com/products/sonarqube/>
- <https://www.altkomsoftware.com/pl/blog/sonarqube-pierwsze-kroki/>
- <https://docs.sonarsource.com/sonarqube/latest/>
- <https://docs.sonarsource.com/sonarqube/9.9/analyzing-source-code/test-coverage/java-test-coverage/>



Dziękujemy za uwagę! :)