

SonarQube

A TOOL FOR STATIC CODE ANALYSIS

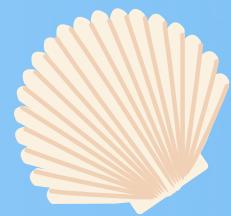
1953067 宋潇歌

2052225 张勤杭

2051840 梁厚

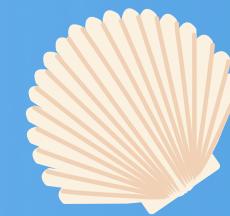
1950389 季艺

Catalog



Introduction

PART-01



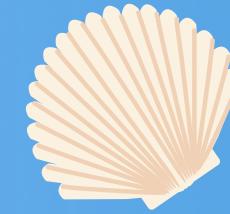
Objectives

PART-02



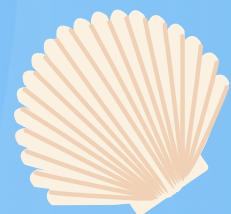
Functionalities

PART-03



Features

PART-04



Usage

PART-05



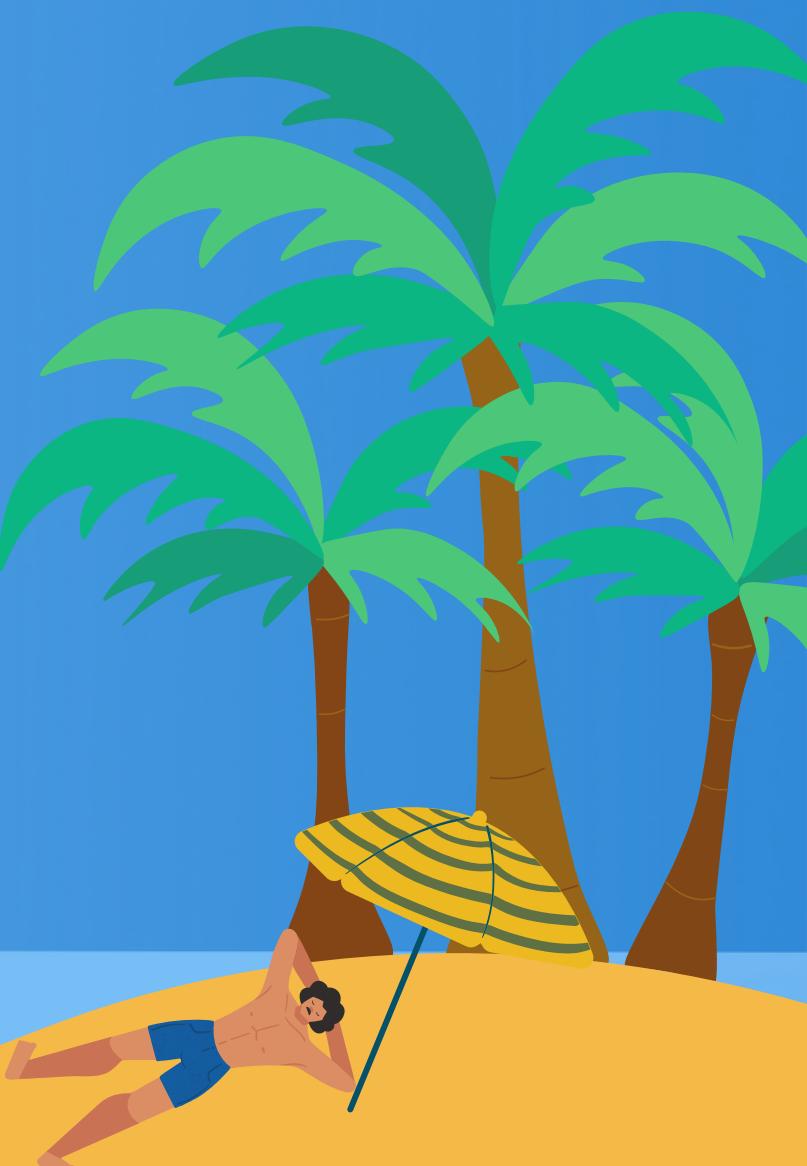
Experimentations and Result Analysis

PART-06



PART-01

Introduction





Introduction

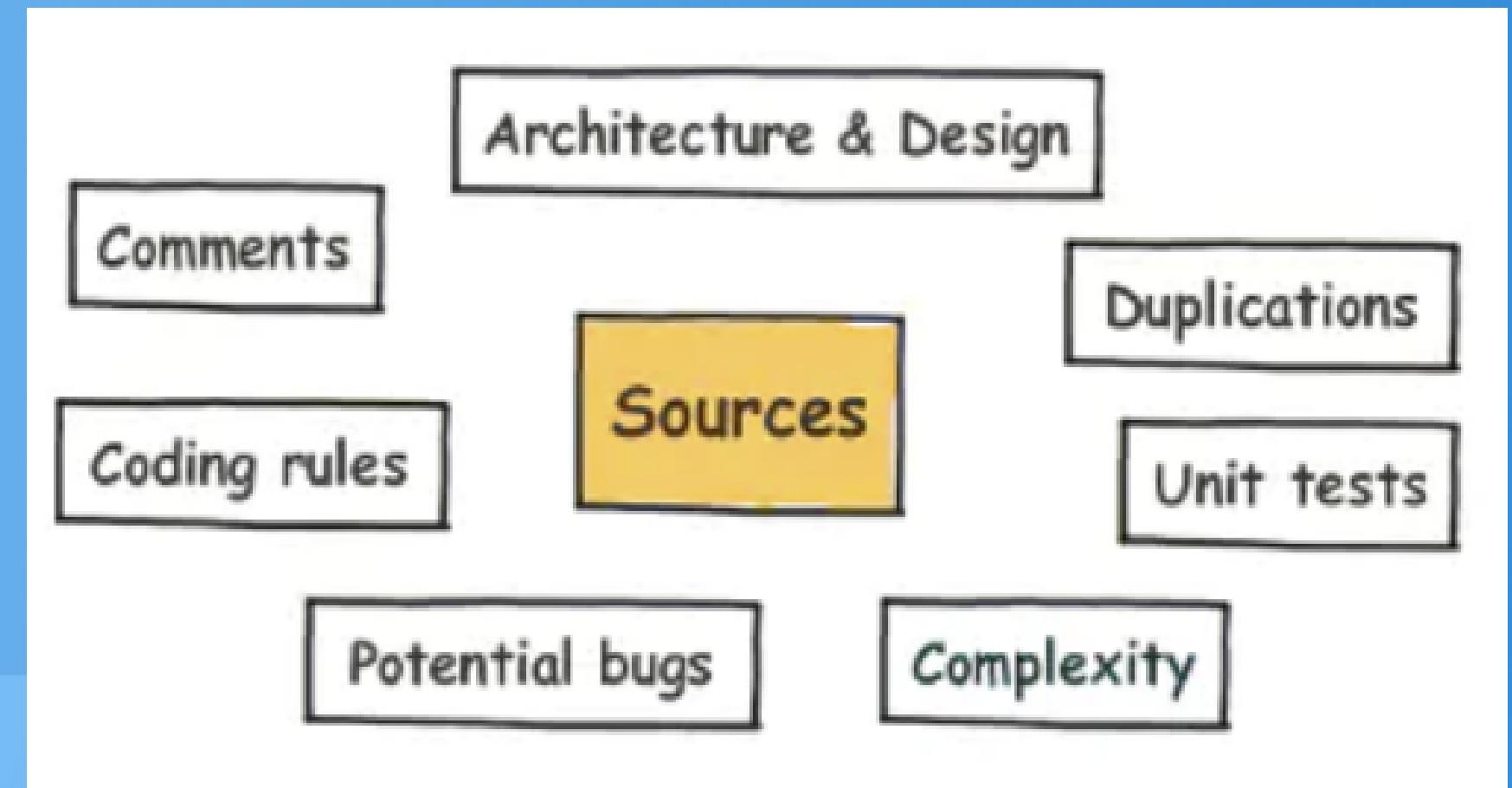
- SonarQube is an open-source static analysis code platform.
- It basically goes through developers' codes and identifies errors at the early stage.
- It is used by developers to manage source code quality and consistency.





ADVANTAGES OVERVIEW

- Check Code Quality and Security
- Multi-dimensional Detection
- Quantitative Analysis Results
- Supports Multiple Languages
- Convenient Configuration





PART-02

OBJECTIVES





PART-02

The main goal of SonarQube is to provide a platform for continuous inspection and analysis of code quality across different programming languages.

It helps developers to identify and fix code smells, vulnerabilities, and bugs early in the development cycle.

It also provides various reports, dashboards, and metrics to track the progress of code quality over time.

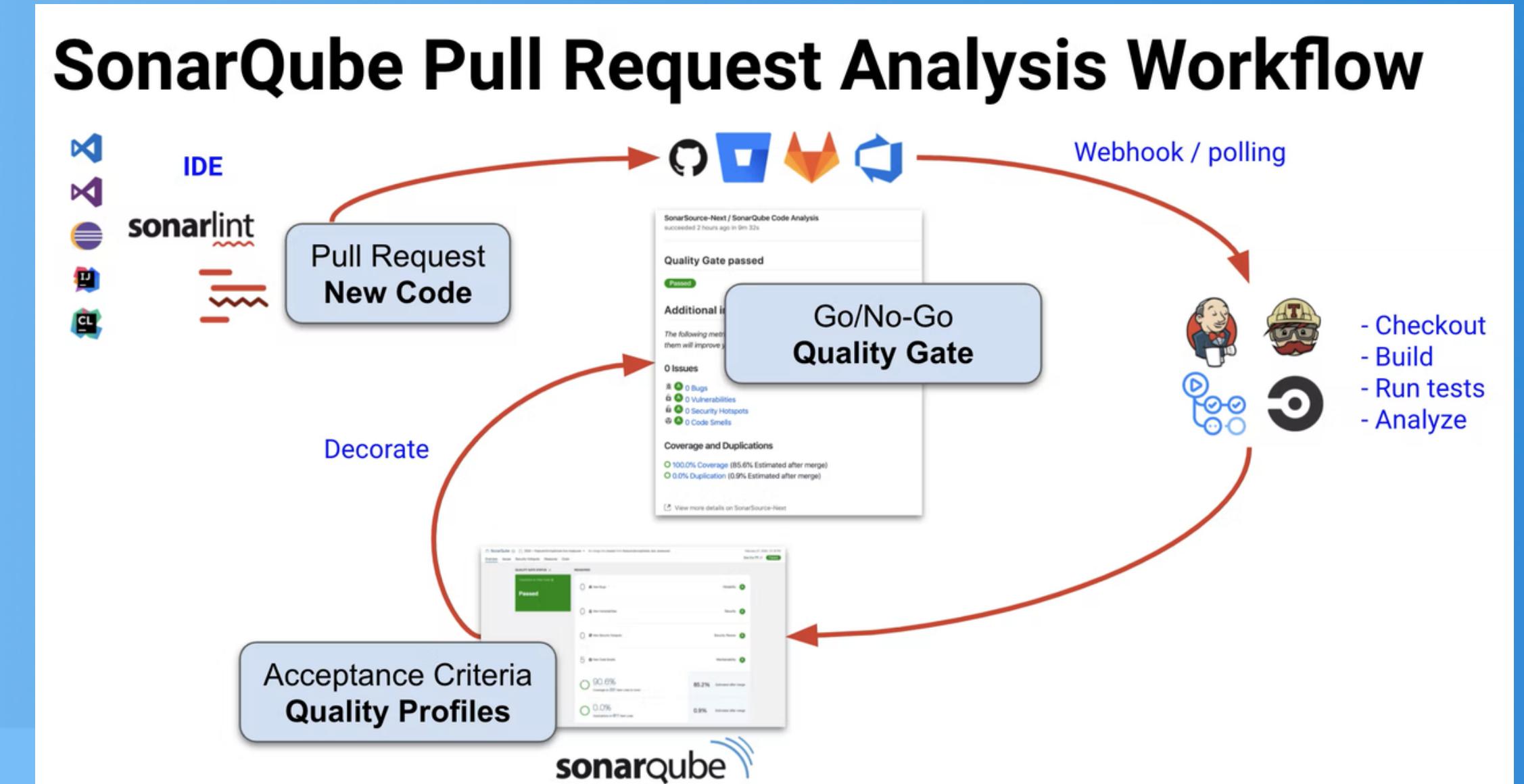
SonarQube aims to promote the best coding practices and improve the overall software quality.





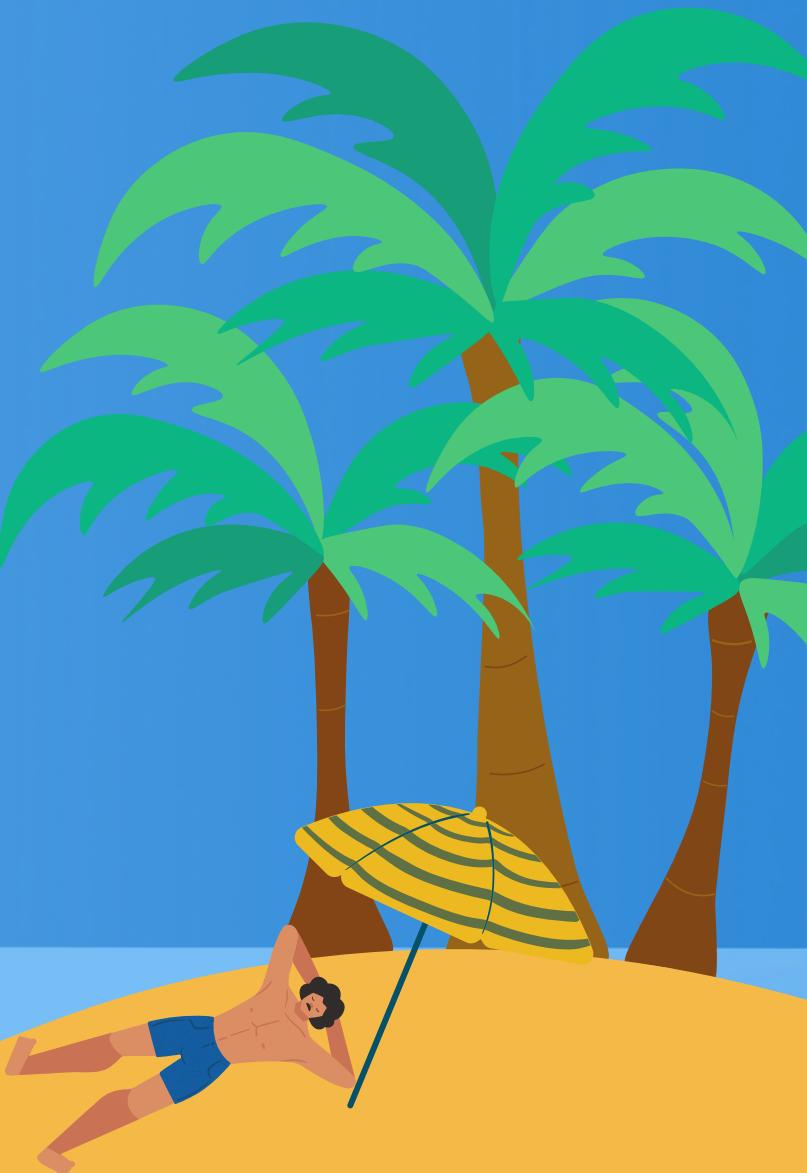
PART-02

All in all, the main goal of SonarQube is to help fix vulnerabilities by enhancing code readability and quality, thus making workflow runs smarter.



PART-03

Functionalities





1. BUG DETECTION

ONE OF THE MOST IMPORTANT FEATURES THAT SONARQUBE PROVIDES IS BUG DETECTION.

A BUG HERE IS NOT NECESSARILY LOGICAL DEFECTS BECAUSE STATIC TESTS DON'T ACTUALLY EXECUTE THE ENTIRE CODE. IT MAY BE CODE THAT WILL CAUSE A LOGICAL DEFECT WHEN EXECUTED. HERE ARE SOME EXAMPLES:



PART-03

1. BUG DETECTION

Example

Cast one of the operands of this multiplication operation to a "long".

Math operands should be cast before assignment [java:S2184](#)

[Get permalink](#)

1 month ago ▾ L27

 Bug ▾ Minor ▾ Open ▾ Not assigned ▾ 5min effort 0 comments

→ cert, cwe, overflow, sans-top25-risky ▾

Where is the issue?

Why is this an issue?

se-proj-rear-end src/.../java/com/evan/seprojrearend/utils TokenName.java

[See all issues in this file](#)

22 10319...
23
24
25 10319...
26
27

```
* 对于一个final变量，如果是基本数据类型的变量，则其数值一旦在初始化之后便不能更改；  
* 如果是引用类型的变量，则在对其初始化之后便不能再让其指向另一个对象  
* **/  
  
//token到期时间10小时  
private static final long EXPIRE_TIME = 10*60*60*1000;
```

 Cast one of the operands of this multiplication operation to a "long".

```
28         //密钥
29         private static final String TOKEN_SECRET = "ljdyaisijin**3nkjnjin??" ;
30 10319...     //token发行人
31         private static final String ISSUER = "auth0";
32 10319...
33         /**
34             * 生成token
35             * @param user
36             * @return
```

Where is the issue? Why is this an issue?

Compliant Solution

```
float twoThirds = 2f/3; // 2 promoted to float. Yields 0.6666667
long millisInYear = 1_000L*3_600*24*365; // 1000 promoted to long. Yields 31_536_000_000
long bigNum = Integer.MAX_VALUE + 2L; // 2 promoted to long. Yields 2_147_483_649
long bigNegNum = Integer.MIN_VALUE-1L; // Yields -2_147_483_649
Date myDate = new Date(seconds * 1_000L);
...
public long compute(int factor){
    return factor * 10_000L;
}

public float compute2(long factor){
    return factor / 123f;
}
```

6



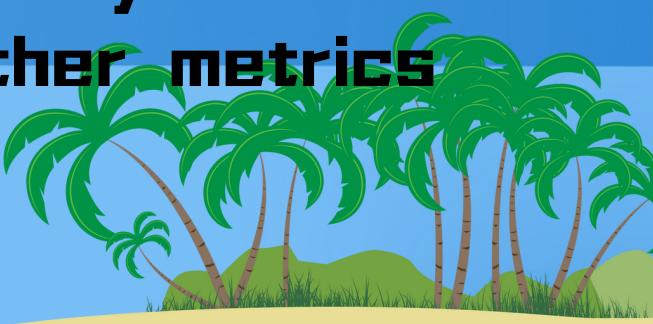
PART-03

2. UNIT TEST EXECUTION

If the project tested has unit tests, SonarQube will execute these tests and give the result to developers.

To execute Unit Tests in SonarQube, the following steps need to be performed:

1. **Configure the Unit Test framework:** SonarQube requires a Unit Test framework to be configured and integrated with the project. This can be done using build tools like Maven, Gradle, or Ant.
2. **Run the Unit Test:** Once the Unit Test framework is configured, the next step is to execute the Unit Tests. This can be done using the build tool's command-line interface or integrated development environment (IDE) plugins.
3. **Generate and import test reports:** After running the Unit Tests, SonarQube requires the test reports generated by the Unit Test framework to be imported. This can be done using SonarQube's Importing Test Results feature.
4. **Analyze Unit Test results:** Once test reports are imported, SonarQube analyzes the Unit Test results and provides quality gate status, code coverage, and other metrics to assess the quality of the code





3. BAD SMELL DETECTION

Bad smells in code are indications of design or implementation issues that can affect the overall quality and performance of the software.
Examples of bad smells include code duplication, complex control flow, long methods, and excessive coupling, which can be divided into these types:

Application-level Smell

Class-level Smell

Method-level Smell

Code Style



3. BAD SMELL DETECTION

Code Style

- Naming style of the code(e.g., camelCase or snake_case)
- Format of the code

Method-level Smell

- Number of parameters
- Clarity of the method name

Class-level Smell

- Size of the class
- Coherence & Coupling of classes
- Cyclomatic complexity (concretely described later)

Application-level Smell

- Controllable side-effects
- Duplicated code (concretely described later)

Here is an example of the detection of a bad smell of code style:





PART-03

EXAMPLE #1

Remove this unused import 'org.springframework.boot.autoconfigure.jdbc.DataSourceAutoConfiguration'. ⓘ
Unnecessary imports should be removed [java:S1128](#)

1 month ago L6

Code Smell Minor Open Not assigned 2min effort 0 comments

unused

Where is the issue? Why is this an issue?

se-proj-rear-end src/main/java/com/evan/seprojrearend/SeProjRearEndApplication.java See all issues in this file

```
1 13067... package com.evan.seprojrearend;
2
3 import org.mybatis.spring.annotation.MapperScan;
4 import org.springframework.boot.SpringApplication;
5 import org.springframework.boot.autoconfigure.SpringBootApplication;
6 import org.springframework.boot.autoconfigure.jdbc.DataSourceAutoConfiguration;
```

Remove this unused import 'org.springframework.boot.autoconfigure.jdbc.DataSourceAutoConfiguration'.

```
7 import org.springframework.context.annotation.ComponentScan;
8 import springfox.documentation.swagger2.annotations.EnableSwagger2;
```

Remove this unused import 'springfox.documentation.swagger2.annotations.EnableSwagger2'.

```
9 13067...
10 13067...
11 13067...
12 13067...
13 13067... //EnableSwagger2
@MapperScan("com.evan.seprojrearend.mapper")
@ComponentScan
@SpringBootApplication(scanBasePackages="controller")
```

THIS BAD SMELL IS CAUSED BY UNNECESSARY IMPORTS. THE IMPORTS PART OF A FILE SHOULD BE HANDLED BY THE INTEGRATED DEVELOPMENT ENVIRONMENT (IDE), NOT MANUALLY BY THE DEVELOPER. UNUSED AND USELESS IMPORTS SHOULD NOT OCCUR IF THAT IS THE CASE. LEAVING THEM IN REDUCES THE CODE'S READABILITY, SINCE THEIR PRESENCE CAN BE CONFUSING.





PART-03

EXAMPLE #2

Rename this field "DEFAULT" to match the regular expression '^[a-z][a-zA-Z0-9]*\$'.

Static non-final field names should comply with a naming convention [java:S3008](#)

1 month ago L31

Get permalink ↗

Convention

Code Smell Minor Open Not assigned 2min effort 0 comments

Where is the issue? Why is this an issue?

se-proj-rear-end src/.../java/com/evan/seprojrearend/config/SwaggerConfig.java See all issues in this file

```
27
28     @EnableSwagger2
29     @Configuration
30     public class SwaggerConfig {
31         private static ApiInfo DEFAULT = null;
```

Remove this unused import 'java.util.List'.

```
27
28     @EnableSwagger2
29     @Configuration
30     public class SwaggerConfig {
31         private static ApiInfo DEFAULT = null;
```

Rename this field "DEFAULT" to match the regular expression '^[a-z][a-zA-Z0-9]*\$'.

Remove the "DEFAULT" field and declare it as a local variable in the relevant methods.

```
32     @Bean
33     public Docket docket() {
34         Contact DEFAULT_CONTACT = new Contact("Onsite2.0", "https://github.com/RiverLiangH/Onsite_2.0_API",
35         "1031903858@qq.com");
```

Rename this local variable to match the regular expression '^[a-z][a-zA-Z0-9]*\$'.

**SHARED NAMING CONVENTIONS
ALLOW TEAMS TO
COLLABORATE EFFICIENTLY.
THIS RULE CHECKS THAT STATIC
NON-FINAL FIELD NAMES
MATCH A PROVIDED REGULAR
EXPRESSION.**





4. CYCLOMATIC COMPLEXITY COMPUTATION

CYCLOMATIC COMPLEXITY IS ONE OF THE CLASS-LEVEL CODE SMELLS, AND IS COMPUTED BY COUNTING THE NUMBER OF DECISION POINTS (IF STATEMENTS, WHILE LOOPS, FOR LOOPS, SWITCH STATEMENTS, ETC.) IN THE CODE. EACH DECISION POINT INCREMENTS THE CYCLOMATIC COMPLEXITY BY ONE. THE FINAL COMPLEXITY SCORE IS CALCULATED BY ADDING ONE TO THE TOTAL NUMBER OF DECISION POINTS IN THE CODE. THE HIGHER THE CYCLOMATIC COMPLEXITY SCORE, THE MORE COMPLEX THE CODE IS AND THE HIGHER THE RISK OF ERRORS AND MAINTENANCE ISSUES.

SONARQUBE USES CYCLOMATIC COMPLEXITY AS ONE OF THE MANY METRICS TO EVALUATE THE QUALITY OF THE CODE. IT PROVIDES A VISUAL REPRESENTATION OF CODE COMPLEXITY THROUGH A CHART, WHICH HELPS DEVELOPERS TO IDENTIFY PROBLEMATIC AREAS AND MAKE IMPROVEMENTS TO THE CODE. HERE IS AN EXAMPLE:





5. DUPLICATED CODE DETECTION

SONARQUBE PROVIDES DUPLICATED CODE DETECTION TO REDUCE COMPLEXITY AND IMPROVE THE QUALITY OF THE CODE. DEVELOPERS MAY ENCAPSULATE THE DUPLICATED CODES IN A METHOD AS POSSIBLE AS THEY CAN.

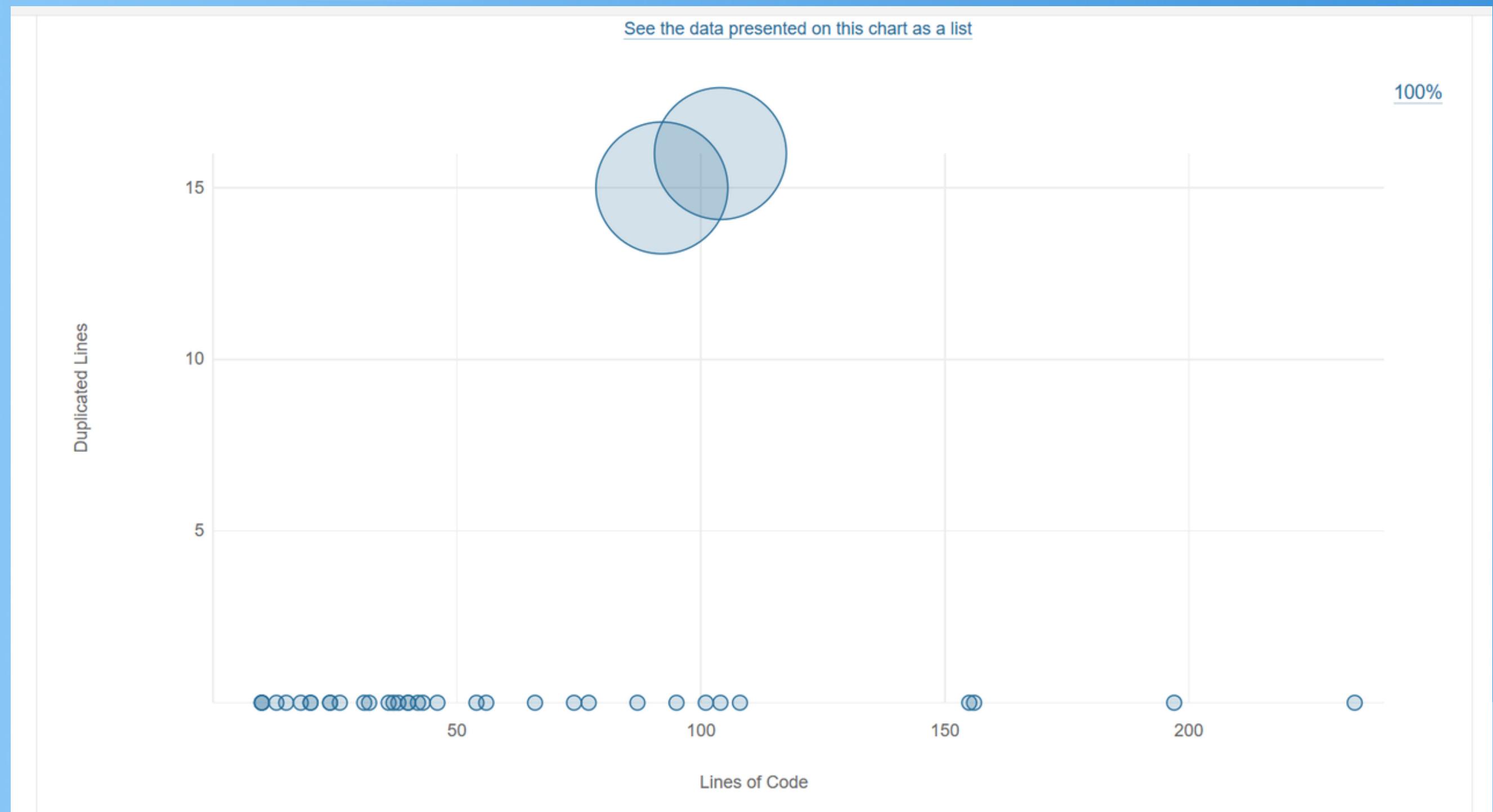
THE DUPLICATED CODE DETECTION FEATURE PROVIDES A GRAPHICAL REPRESENTATION OF CODE DUPLICATION, HIGHLIGHTING AREAS OF CODE THAT ARE REPEATED, ALLOWING DEVELOPERS TO QUICKLY IDENTIFY POTENTIAL CODE SMELLS. THE TOOL ALSO SUPPORTS THE IDENTIFICATION OF FALSE POSITIVES, WHERE CODE MAY APPEAR DUPLICATED DUE TO VARIABLE NAMING OR FORMATTING DIFFERENCES, BUT IN FACT, IS NOT REDUNDANT.





PART-03

EXAMPLE



PART-04

Features





PART-04

SUPPORT MULTI-LANGUAGES

SonarQube can support more than 25 programming languages. For all supported programming languages, SonarQube provides static analysis of source code.





MINIMAL CONFIGURATION REQUIRED

- **Build Frameworks**
- **For Java, C#, C, C++ and Objective-C, SonarQube couples tightly with build frameworks to get project info and ensure accurate analysis.**
- **Track Code Changes**

Native Git and SVN support.

Just check out your repo and let SonarQube track new code.

- **Flexibility to connect with DevOps Platform**
- **Whether it's self-managed/on-prem or in-cloud/SaaS, SonarQube has the flexibility to connect with the DevOps Platform.**





PART-04

INTEGRATE OTHER TESTING TOOLS

- PMD-CPD, FindBugs, CheckStyle, PMD-P3C(alibaba) ...

TEAM MANAGEMENT

Number of projects
83

—

Number of lines of code
933k

Releasability	Reliability	Security Vulnerabilities	Security Review	Maintainability
<div style="text-align: center;">A</div> <p>Metric trend was B 7 months ago</p> <p>Lowest rated projects 7 projects Failed</p> <p>Measure Activity</p>	<div style="text-align: center;">A</div> <p>Metric trend was B 3 months ago</p> <p>Lowest rated projects 3 projects in E</p> <p>Measure Activity</p>	<div style="text-align: center;">A</div> <p>Metric trend was B 7 months ago</p> <p>Lowest rated projects 1 project in D</p> <p>Measure Activity</p>	<div style="text-align: center;">A</div> <p>Metric trend has always been A</p> <p>Lowest rated projects 3 projects in B</p> <p>Measure Activity</p>	<div style="text-align: center;">A</div> <p>Metric trend has always been A</p> <p>Measure Activity</p>

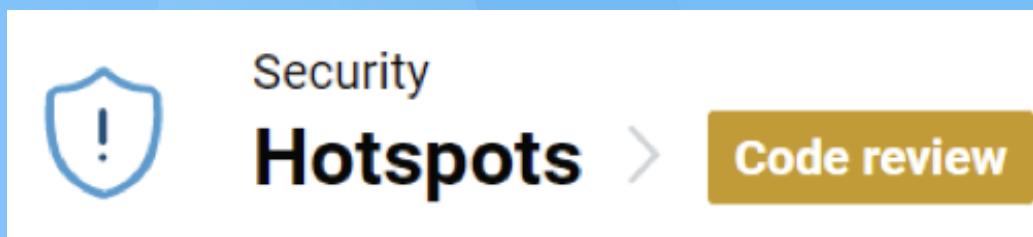
A decorative illustration of several green palm trees with brown trunks, set against a light blue background.



PART-04

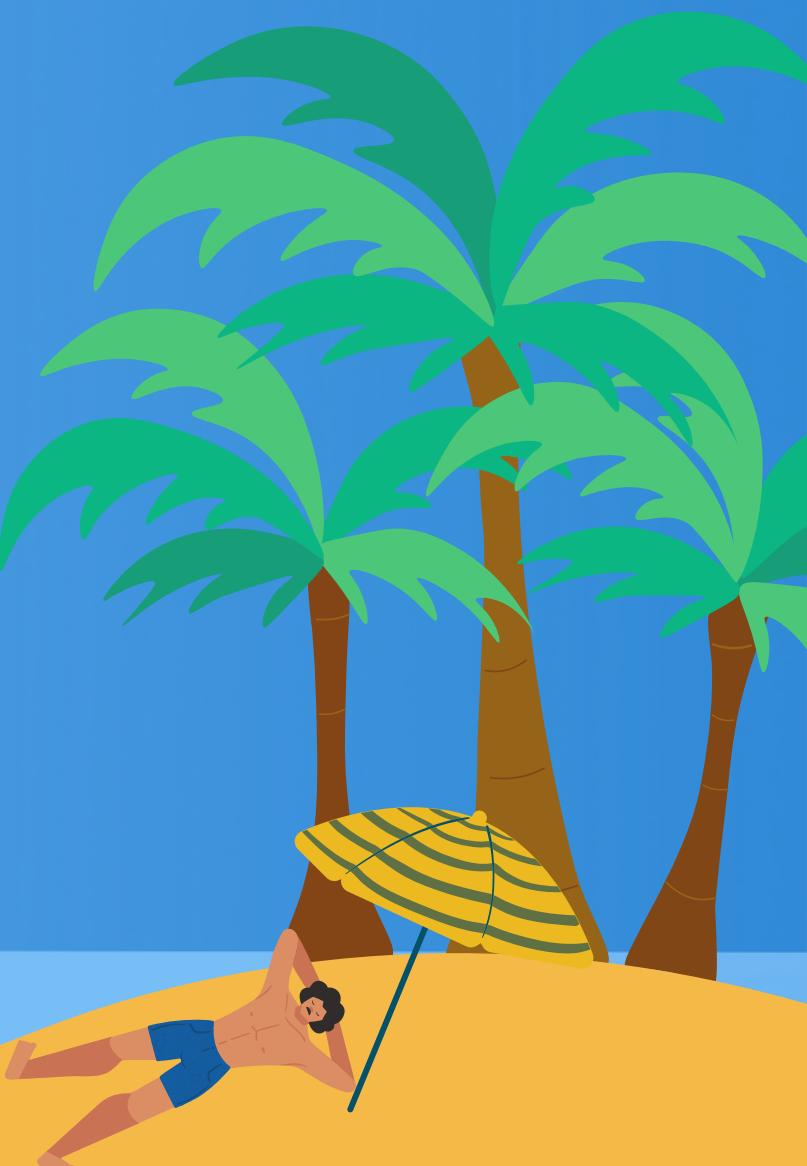
SECURITY ANALYSIS

- Security issues should not be considered the de facto realm of security teams
- Make development teams deeply understand security issues and improve clean coding abilities.
- Early security test and early fix.
- Clear security issues, clear actions
- Maximum protection with detection of injection flaws
- Chase down the bad actors
- Dedicated UI to track untrusted user input
- More security features at enterprise level



PART-05

Usage





1.WORKFLOW

Steps to install a local instance of SonarQube and analyze a project are as follows:

- Install Java (Oracle JRE or OpenJDK) on the machine where you plan to run SonarQube (prerequisite). However, different versions of SonarQube require matching versions of JDK and database. More details are available on the official website document: Prerequisites and overview
- Download the packet of SonarQube from the official website and deploy it on your local machine. Once the instance is up and running, log in to <http://127.0.0.1:9000> to use the system.
- Create a new project in the system and configure SonarQube in your local IDE (my IDE is IDEA). When configured successfully, execute SonarQube in your project.
- After reading the code, SonarQube will automatically perform code analysis and generate result reports.





PART-03

2. CONFIGURATION ---- STEP#1

Check matching version of JDK and viable database in guidance document. SonarQube9.9 requires JDK17 and Oracle 19c/21c to run. Version of JDK and variation are strictly regulated. If the current environment variable version is not JDK17, you need to change it.

	A	B	C
1	Java	Server	Scanners
2	Oracle JRE	17√	17√
3		11×	11√
4	OpenJDK	17√	17√
5		11×	11√





PART-03

2. CONFIGURATION ---- STEP#2

Configure your SonarQube initially.

Amend document in \sonarqube-9.9.0.65466\sonarqube-9.9.0.65466\conf, adding:

- **sonar.jdbc.url=jdbc:oracle:thin:@localhost:1521/orcl**
- **sonar.jdbc.username=*****
- **sonar.jdbc.password=*****
- **sonar.jdbc.driverClassName=oracle.jdbc.driver.OracleDriver**
- **sonar.web.port=9000**
- **sonar.login=admin**
- **sonar.password=admin**

Amend document in \sonarqube-9.9.0.65466\sonarqube-9.9.0.65466\elasticsearch\config\elasticsearch.yml, adding:

- **network.host: 127.0.0.1**

Copy your JDBC to \sonarqube-9.9.0.65466\sonarqube-9.9.0.65466\extensions\jdbc-driver\oracle, or it will go wrong.



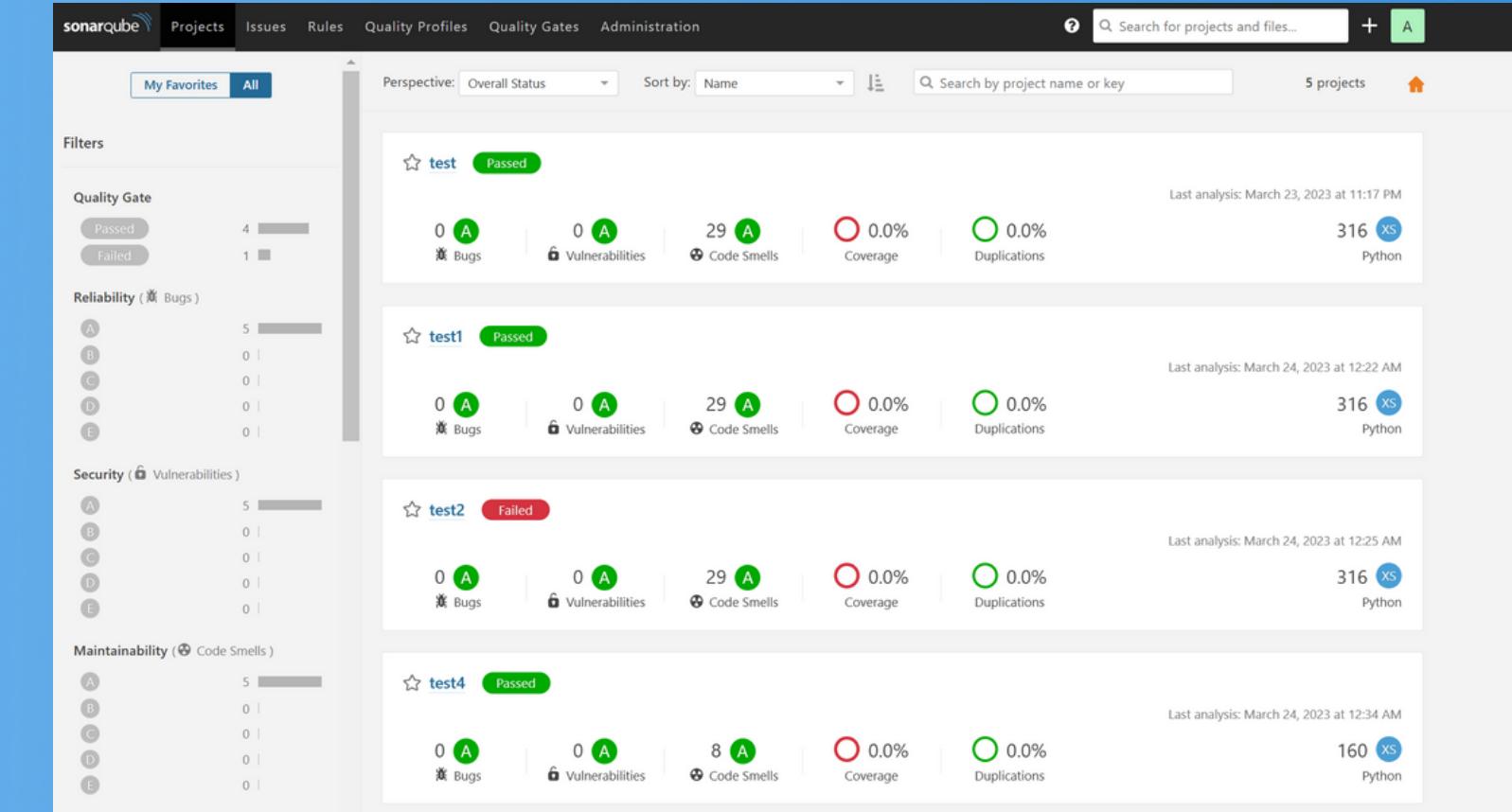


PART-05

2. CONFIGURATION

```
(wrapper) --> Wrapper Started as Console
(wrapper) Launching a JVM...
(jvm 1) Wrapper (Version 3.2.3) http://wrapper.tanukisoftware.org
(jvm 1) Copyright 1999-2006 Tanuki Software, Inc. All Rights Reserved.

[jvm 1] 2023.03.27 20:53:55 INFO app[] [o.s.a.AppFileSystem] Cleaning or creating temp directory C:\Users\zhang\Desktop\sonarqube-7.9.6\temp
[jvm 1] 2023.03.27 20:53:55 INFO app[] [o.s.a.EsSettings] Elasticsearch listening on /127.0.0.1:9001
[jvm 1] 2023.03.27 20:53:55 INFO app[] [o.s.a.ProcessLauncherImpl] Launch process[[key='es', ipcIndex=1, logFilenamePrefix=es]] from [C:\Users\zhang\Desktop\sonarqube-7.9.6\elasticsearch]: C:\Program Files\Java\jdk-11.0.16.1\bin\java -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitiatingOccupancyOnly -Des.networkaddress.cache.ttl=60 -Des.networkaddress.cache.negative.ttl=10 -XX:+AlwaysPreTouch -Xss1m -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djna.nosys=true -XX:-OmitStackTraceInFastThrow -Dio.netty.noUnsafe=true -Dio.netty.noKeySetOptimization=true -Dio.netty.recycler.maxCapacityPerThread=0 -Dlog4j.shutdownHookEnabled=false -Dlog4j2.disable.jmx=true -Djava.io.tmpdir=C:\Users\zhang\Desktop\sonarqube-7.9.6\temp -XX:ErrorFile=../logs/es_hs_err_pid%.log -Des.enforce.bootstrap.checks=true -Xms512m -Xmx512m -XX:+HeapDumpOnOutOfMemoryError -Delasticsearch -Des.path.home=C:\Users\zhang\Desktop\sonarqube-7.9.6\elasticsearch -Des.path.conf=C:\Users\zhang\Desktop\sonarqube-7.9.6\temp\conf\es -cp lib/* org.elasticsearch.bootstrap.Elasticsearch
[jvm 1] 2023.03.27 20:53:55 INFO app[] [o.s.a.SchedulerImpl] Waiting for Elasticsearch to be up and running
[jvm 1] Java HotSpot(TM) 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future release.
[jvm 1] 2023.03.27 20:53:56 INFO app[] [o.e.p.PluginsService] no modules loaded
[jvm 1] 2023.03.27 20:53:56 INFO app[] [o.e.p.PluginsService] loaded plugin [org.elasticsearch.transport.Netty4Plugin]
```





PART-05

3. ADD PROJECT AND INTEGRATE LOCAL IDE (IDEA)

1. ADD PROJECT: CLICK ADD PROJECT ON THE MAIN SCREEN TO CREATE A NEW PROJECT. IT WILL OFFER SOME CONFIGURATION MESSAGES TO YOU AS FOLLOWS:

MVN CLEAN VERIFY SONAR:SONAR

-DSONAR.PROJECTKEY=ONSITEAPI

-DSONAR.HOST.URL=HTTP://127.0.0.1:9000

-DSONAR.LOGIN=SQP_D42F10B92092F9ACF9B3E86720561F4EC825316F

2. CONFIGURE THE LOCAL PROJECT, ADDING THE FOLLOWING CODES TO YOUR SPRINGBOOT PROJECT (MODIFYING POM.XML):

<PROFILES>

<PROFILE>

<ID>SONAR</ID>

<ACTIVATION>

<ACTIVEBYDEFAULT>TRUE</ACTIVEBYDEFAULT>

</ACTIVATION>

<PROPERTIES>

<SONAR.PROJECTKEY>ONSITEAPI</SONAR.PROJECTKEY>

<SONAR.HOST.URL>HTTP://LOCALHOST:9000</SONAR.HOST.URL>

<SONAR.LOGIN>SQP_D42F10B92092F9ACF9B3E86720561F4EC825316F</SONAR.LOGIN>

</PROPERTIES>

</PROFILE>

</PROFILES>

3. CLICK SONAR:SONAR USING SONARQUBE VIA MEAVEN, LIKE:

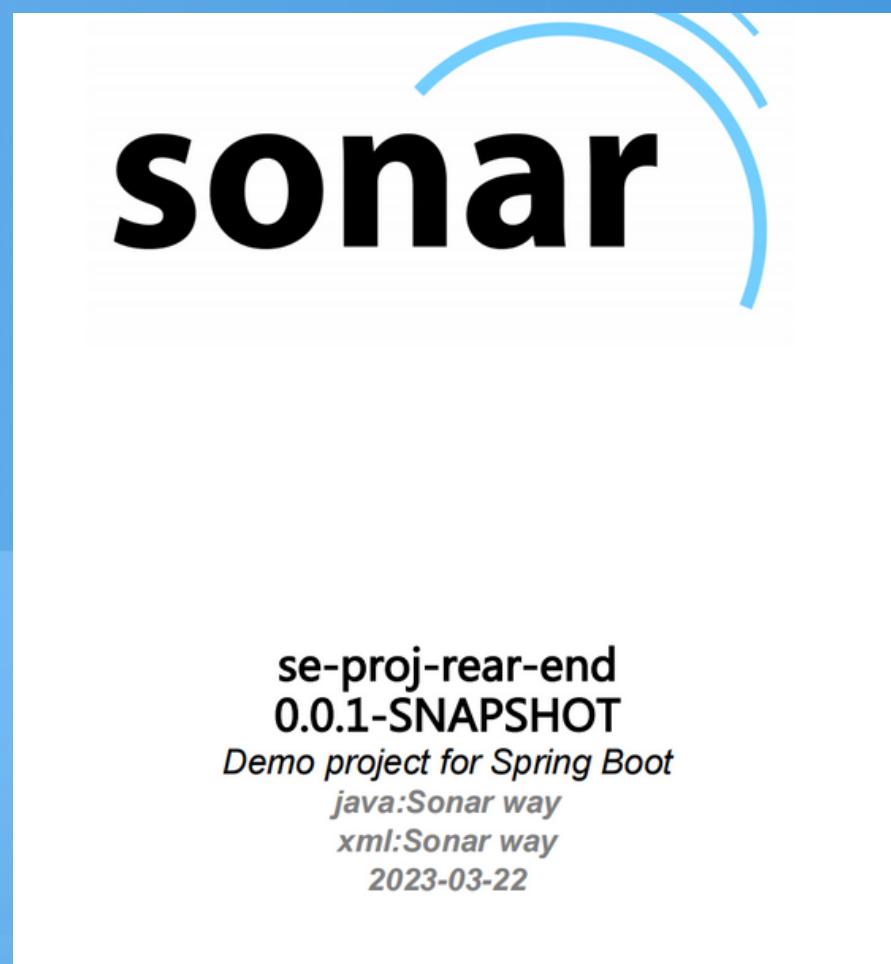
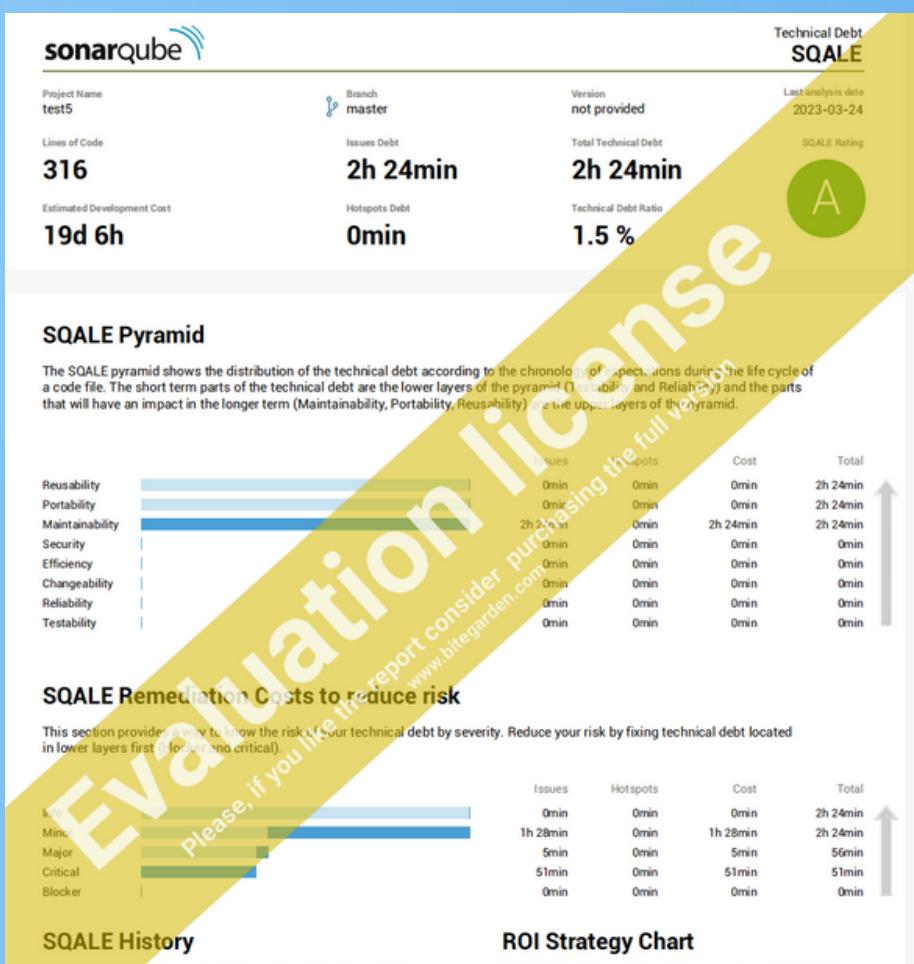




PART-05

4. GET YOUR REPORT

- THEN YOU WILL GET YOUR REPORT ON `HTTP://127.0.0.1:9000/PROJECTS`. THE PDF REPORT IS ALSO AVAILABLE, YOU CAN DOWNLOAD THE REPORT ON THE WEBSITE. (`ONSITEAPIPDF`)



PART-06

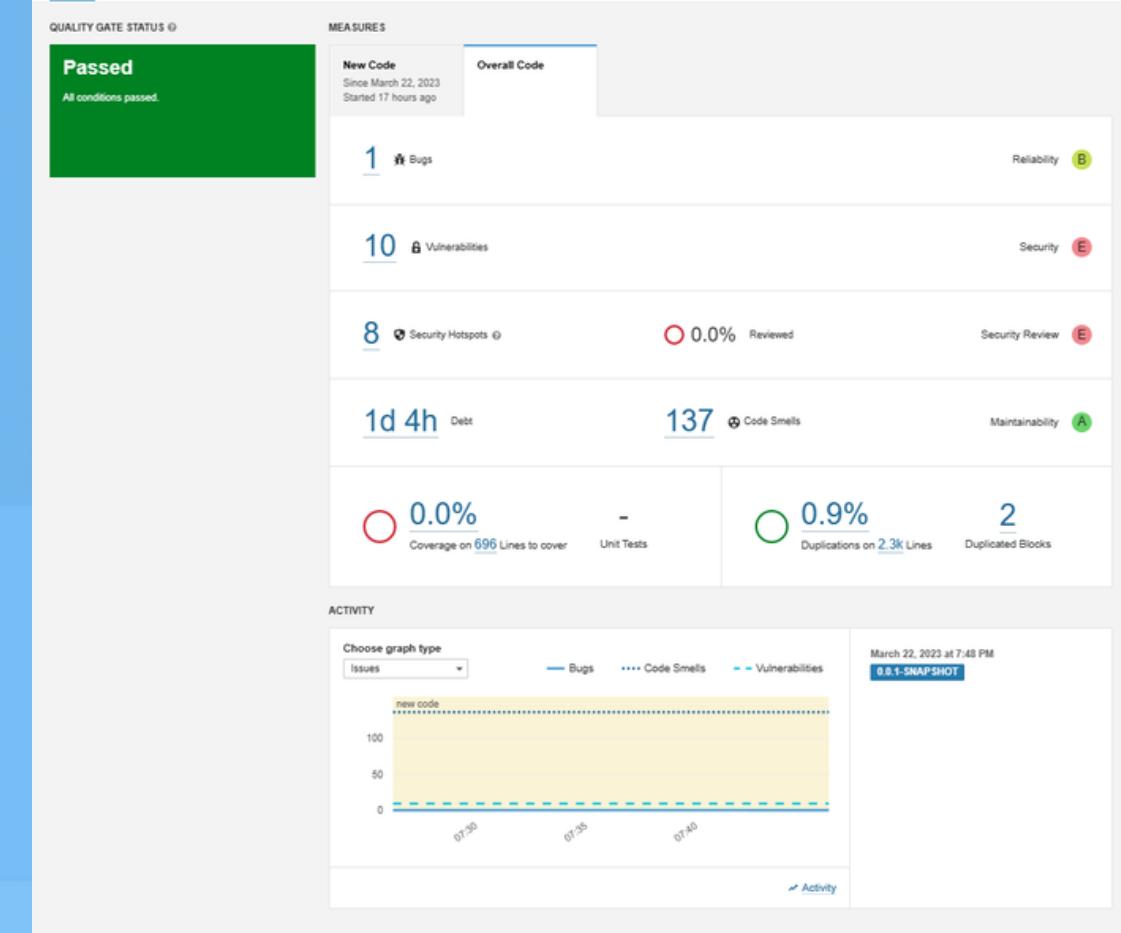
Experimentations and Result Analysis





TESTING PROJECT #1

A SPRINGBOOT PROJECT(JAVA) OFFERS API FOR DATABASE TO THE FRONT-END PROJECT. OVERALL ANALYSIS IS AS FOLLOWS:





PART-06

TESTING PROJECT #1

sonar | se-proj-rear-end | Sonar Report

1. se-proj-rear-end
报告提供了项目指标的概要，显示了与项目质量相关的最重要的指标。如果需要获取更详细的信息，请登陆网站进一步查询。
报告的项目为se-proj-rear-end，生成时间为2023-03-22，使用的质量配置为 java:Sonar way
xml:Sonar way，共计 503条规则。

1.1. 概述

编码问题

类别	数量	修复工作	时间
Bug	1	可靠性修复工作	5min
漏洞	10	安全修复工作	9h0min
坏味道	137	技术债务	12h53min

148 问题

状态	数量
开启问题	148
重开问题	0
确认问题	0
误判问题	0
不修复的问题	0
已解决的问题	0
已删除的问题	0
阻断	11
严重	10
主要	69
次要	57
提示	1

静态分析

项目规模

1

sonar | se-proj-rear-end | Sonar Report

代码行数	行数	方法	类	文件	目录	重复行(%)
2335	3320	218	30	37	N/A	0.9

复杂度

文件	复杂度	注释(%)
248	8.3	14.0
		注释行数 380

1.2. 问题分析

违反最多的规则TOP10

规则	数量
Unnecessary imports should be removed	39
Sections of code should not be commented out	26
Standard outputs should not be used directly to log anything	22
Restricted Identifiers should not be used as Identifiers	12
Credentials should not be hardcoded	8
String literals should not be duplicated	7
Public constants and fields initialized at declaration should be "static final" rather than merely "final"	6
Raw types should not be used	4
URIs should not be hardcoded	3
"static" base class members should not be accessed via derived types	2

2

sonar | se-proj-rear-end | Sonar Report

违规最多的文件TOP5

文件	数量
UserController.java	19
OssTemplate.java	16
SubmitController.java	14
SwaggerConfig.java	13
SubmitService.java	12

复杂度最高的文件TOP5

文件	复杂度
User.java	44
Scene.java	38
Competition.java	34
Submit.java	27
UserService.java	17

重复行最多的文件TOP5

文件	数量
SubmitController.java	16
EntryController.java	15

1.3. 问题详情

规则 Unnecessary imports should be removed

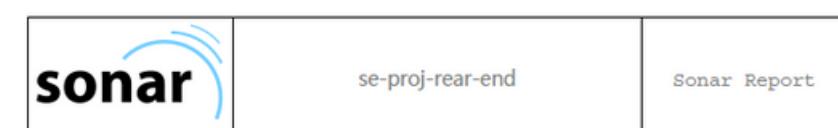
3





PART-06

TESTING PROJECT #1

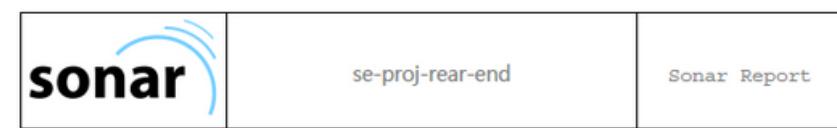


UserController.java	22, 23, 27, 30, 32
CompetitionMapper.java	4
SeProjRearEndApplication.java	6

规则	Sections of code should not be commented out
规则描述	Programmers should not comment out code as it bloats programs and reduces readability. Unused code should be deleted and can be retrieved from source control history if required.

文件名称	违规行
SceneController.java	75, 91
SubmitController.java	77
SubmitService.java	61, 69
UserController.java	193, 212, 129, 174
EntryController.java	85, 88
EntryService.java	55
SubmitController.java	100
SubmitMapper.java	20
UserController.java	150
SubmitMapper.java	22, 28
UserController.java	228
UserService.java	87
UserController.java	104
WebConfiguration.java	83
UserController.java	57
OssTemplate.java	50, 92, 138, 145

规则	Standard outputs should not be used directly to log anything
----	--



规则描述	Utility classes, which are collections of static members, are not meant to be instantiated. Even abstract utility classes, which can be extended, should not have public constructors.
	Java adds an implicit public constructor to every class which does not define at least one explicitly. Hence, at least one non-public constructor should be defined.

Noncompliant Code Example

```
class StringUtils { // Noncompliant
    public static String concatenate(String s1, String s2) {
        return s1 + s2;
    }
}
```

Compliant Solution

```
class StringUtils { // Compliant
    private StringUtils() {
        throw new IllegalStateException("Utility class");
    }

    public static String concatenate(String s1, String s2) {
        return s1 + s2;
    }
}
```

Exceptions

When class contains `public static void main(String[] args)` method it is not considered as utility class and will be ignored by this rule.

文件名称	违规行
TokenUtils.java	20

1.4. 质量配置

质量配置	java:Sonar way	Bug:139	漏洞:31	坏味道:272
规则		类型	违规级别	
Methods should not call same-class methods with incompatible "@Transactional" values	Bug		阻断	
Methods "wait(..)", "notify()" and "notifyAll()" should not be called on Thread instances	Bug		阻断	
Files opened in append mode should not be used with ObjectOutputStream	Bug		阻断	
"PreparedStatement" and "ResultSet" methods should be called with valid indices	Bug		阻断	
"wait(..)" should be used instead of "Thread.sleep(..)" when a lock is held	Bug		阻断	



Printf-style format strings should not lead to unexpected behavior at runtime	Bug	阻断
"@SpringBootApplication" and "@ComponentScan" should not be used in the default package	Bug	阻断
"@Controller" classes that use "@SessionAttributes" must call "setComplete" on their "SessionStatus" objects	Bug	阻断
Loops should not be infinite	Bug	阻断
"wait" should not be called when multiple locks are held	Bug	阻断
Double-checked locking should not be used	Bug	阻断
Resources should be closed	Bug	阻断
Regular expressions should be syntactically valid	Bug	严重
Locks should be released on all paths	Bug	严重
Jump statements should not occur in "finally" blocks	Bug	严重
"Random" objects should be reused	Bug	严重
"super.finalize()" should be called at the end of "Object.finalize()" implementations	Bug	严重
Assertions comparing incompatible types should not be made	Bug	严重
Assertion methods should not be used within the try block of a try-catch catching an Error	Bug	严重
The signature of "finalize()" should match that of "Object.finalize()"	Bug	严重
Only one method invocation is expected when testing checked exceptions	Bug	严重
"runFinalizersOnExit" should not be called	Bug	严重
Regex boundaries should not be used in a way that can never be matched	Bug	严重
"ScheduledThreadPoolExecutor" should not have 0 core threads	Bug	严重
Regex patterns following a possessive quantifier should not always fail	Bug	严重
Zero should not be a possible denominator	Bug	严重
Back references in regular expressions should only refer to capturing groups that are matched before the reference	Bug	严重
Regex lookahead assertions should not be contradictory	Bug	严重
JUnit5 inner test classes should be annotated with @Nested	Bug	严重
Map "computeIfAbsent()" and "computeIfPresent()" should not be used to add "null" values	Bug	严重
Members ignored during record serialization should not be used	Bug	严重
Getters and setters should access the expected fields	Bug	严重



PART-06

TESTING PROJECT #2

PYTHON

test master

Last analysis had 1 warnings March 23, 2023 at 11:17 PM Version not provided

Overview Issues Measures Code Activity Administration More

1 / 29 issues

conway.py

Rename this parameter "N" to match the regular expression ^[a-z][a-z0-9]*\$. Code Smell

Rename this parameter "fastMode" to match the regular expression ^[a-z][a-z0-9]*\$. Code Smell

Rename this field "fastMode" to match the regular expression ^[a-z][a-z0-9]*\$. Code Smell

Rename this field "aliveValue" to match the regular expression ^[a-z][a-z0-9]*\$. Code Smell

Rename this field "deadValue" to match the regular expression ^[a-z][a-z0-9]*\$. Code Smell

Rename method "getStates" to match the regular expression ^[a-z][a-z0-9](2,\$). Code Smell

@author: shakes
import numpy as np
from scipy import signal
from rle import RunLengthEncodedParser

class GameOfLife:
 ...
 Object for computing Conway's Game of Life (GoL) cellular machine/automata
 ...

def __init__(self, N=256, finite=False, fastMode=False):

Rename this parameter "N" to match the regular expression ^[a-z][a-z0-9]*\$. See Rule 4 days ago L19 convention
Code Smell Minor Open Not assigned 2min effort Comment

Rename this parameter "fastMode" to match the regular expression ^[a-z][a-z0-9]*\$. See Rule 4 days ago L19 convention
Code Smell Minor Open Not assigned 2min effort Comment

self.grid = np.zeros((N,N), np.int)
self.neighborhood = np.ones((3,3), np.int) # 8 connected kernel
self.neighborhood[1,1] = 0 #do not count centre pixel
self.finite = finite
self.fastMode = fastMode

Rename this field "fastMode" to match the regular expression ^[a-z][a-z0-9]*\$. See Rule 4 days ago L24 convention
Code Smell Minor Open Not assigned 2min effort Comment

self.aliveValue = 1

Rename this field "aliveValue" to match the regular expression ^[a-z][a-z0-9]*\$. See Rule 4 days ago L25 convention
Code Smell Minor Open Not assigned 2min effort Comment

self.deadValue = 0

Thanks

