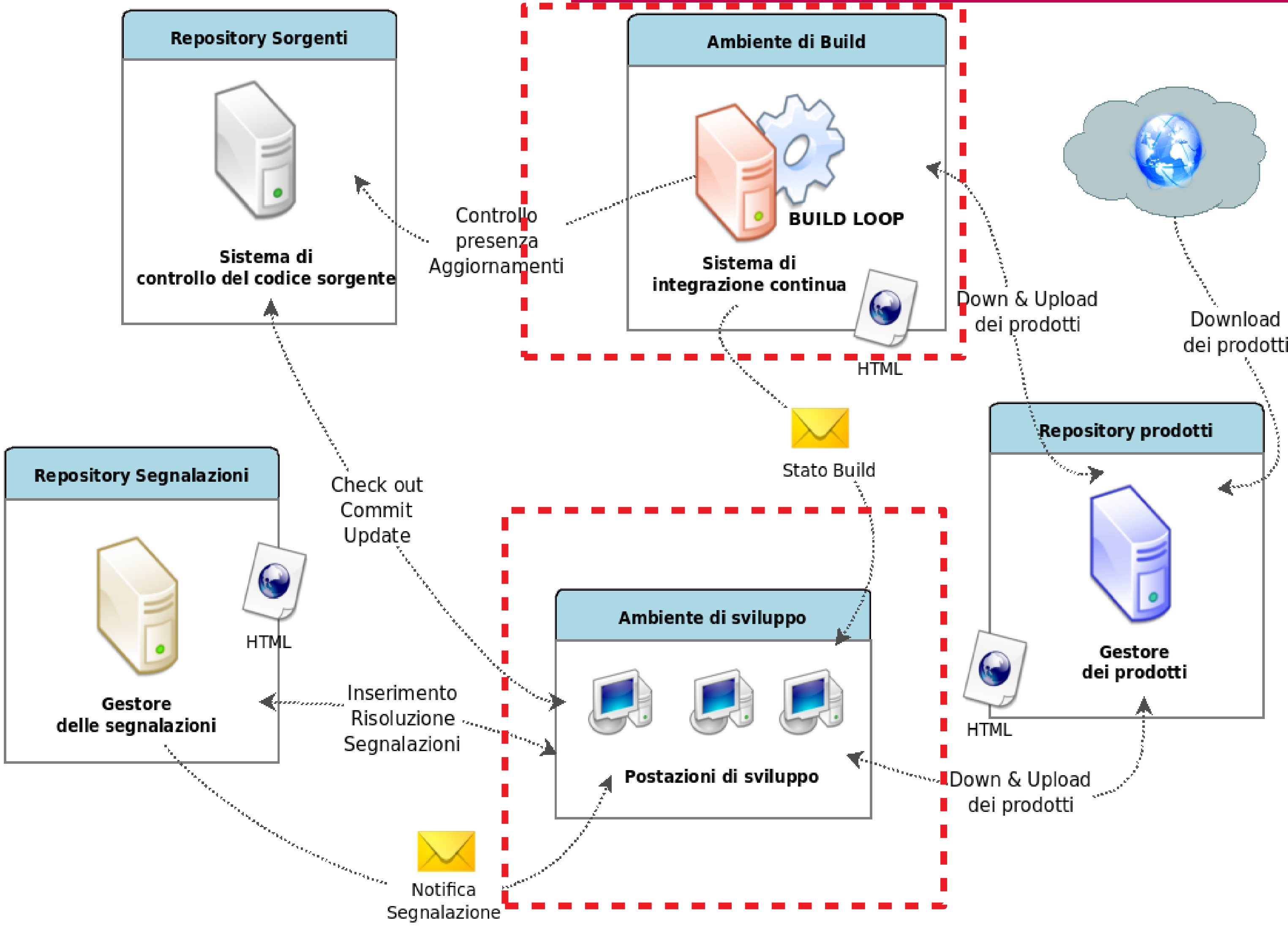
## Analisi statica del codice

# METODI E TECNOLOGIE PER LO SVILUPPO SOFTWARE

Nicola Bertazzo
nicola.bertazzo [at] unipd.it

Università degli Studi di Padova
Dipartimento di Matematica
Corso di Laurea in Informatica, A.A. 2022 – 2023

## Visione Generale



## Static program/code analysis

### Definizione

**Static program analysis/static code analysis** is the **analysis of computer software** that is performed **without actually executing programs**, in contrast with dynamic analysis, which is analysis performed on programs while they are executing. In most cases the analysis is performed on some version of the source code, and in the other cases, some form of the object code.

The term is usually applied to the **analysis performed by an automated tool**, with **human analysis** being called program understanding, program comprehension, or code review. Software inspections and software walkthroughs are also used in the latter case.

Appartiene alle seguenti categorie di test:

- **Test statico**: non richiede l'esecuzione
- **White Box**: è presente il codice sorgente
- **Test Non Funzionale**

WIKIPEDIA
The Free Encyclopedia

## Automated code review

### Definizione

**Automated code review** software **checks source code for compliance** with a **predefined set of rules or best practices**. The use of analytical methods to inspect and review source code to detect bugs has been a **standard development practice**. This process can be accomplished both manually and in an automated fashion. With automation, software tools provide assistance with the code review and inspection process. **The review program or tool typically displays a list of warnings** (violations of programming standards). A review program can also provide an automated or a programmer-assisted way to correct the issues found. This is a component for mastering easily software. This is contributing to the Software Intelligence practice.

Some static code analysis tools can be used to assist with automated code review. They do not compare favorably to manual reviews, however **they can be done faster** and more efficiently. These tools also **encapsulate deep knowledge of underlying rules and semantics** required to perform this type analysis such that it does not require the human code reviewer to have the same level of expertise as an expert human auditor.

Wikipedia
The Free Encyclopedia

## Teoria delle finestre rotte

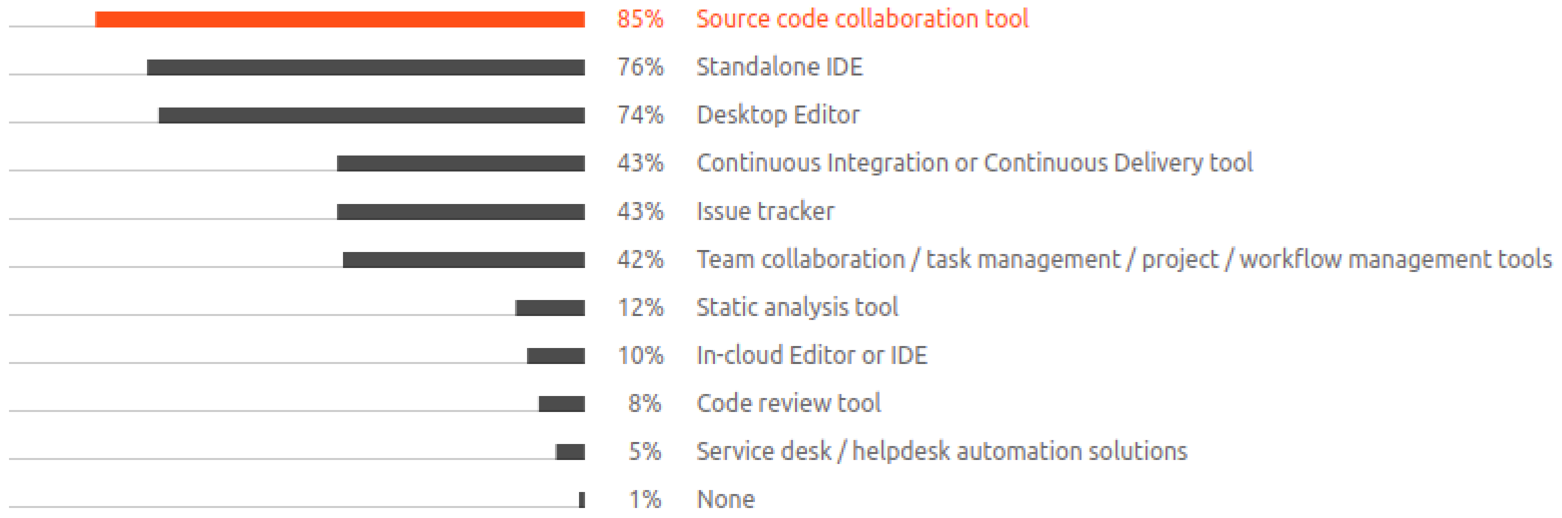
**La teoria delle finestre rotte** è una teoria criminologica sulla **capacità del disordine** urbano e del vandalismo **di generare criminalità aggiuntiva** e comportamenti anti-sociali. La teoria afferma che mantenere e controllare ambienti urbani **reprimendo i piccoli reati**, gli atti vandalici, la deturpazione dei luoghi, il bere in pubblico, la sosta selvaggia o l'evasione nel pagamento di parcheggi, mezzi pubblici o pedaggi, contribuisce a creare un clima di ordine e legalità e **riduce il rischio di crimini più gravi.**

Ad esempio l'esistenza di una finestra rotta (da cui il nome della teoria) potrebbe generare fenomeni di emulazione, portando qualcun altro a rompere un lampione o un idrante, dando così inizio a una spirale di degrado urbano e sociale.

Wikipedia
The Free Encyclopedia

## Team Tools

### Which of the following tools do you regularly use?



The use of CI / CD tools is most widespread among DevOps engineers, architects, team leads, and developer advocates.

https://www.jetbrains.com/lp/devecosystem-2021/team-tools/

## Static code analysis tool

### Funzionalità

Simili ad un correttore ortografico, permettono di:

- Imporre il rispetto di convenzioni e stili
- Verificare la congruità della documentazione
- Controllare metriche ed indicatori (complessità ciclomatica, grafo delle dipendenze, numerosità delle linee di codice)
- Ricercare codice copiato in più punti
- Ricercare errori comuni nel codice
- Misurare la percentuale di codice testato
- Ricercare indicatori di parti incomplete (p. es. tag)

CheckStyle

PMD & CPD

Find Bugs

## Checkstyle

Checkstyle is a development tool to help programmers write Java code that adheres to a **coding standard**. It automates the process of checking Java code to spare humans of this boring (but important) task. This makes it ideal for projects that want to **enforce a coding standard**.

Checkstyle can check many aspects of your source code. It can find class **design problems**, method design problems. It also has the ability to check code layout and formatting issues.

Per una lista dei controlli vedi qui:
http://checkstyle.sourceforge.net/checks.html

Esiste un plugin Maven per eseguire checkstyle:
https://maven.apache.org/plugins/maven-checkstyle-plugin/

## FindBugs / SpotBugs

FindBugs is a program which uses static analysis to **look for bugs in Java code.**

SpotBugs is the spiritual successor of FindBugs, carrying on from the point where it left off with support of its community.

Una lista completa dei controlli di FindBugs:
http://findbugs.sourceforge.net/bugDescriptions.html
Una lista completa dei controlli di SpotBugs:
https://spotbugs.readthedocs.io/en/latest/bugDescriptions.html

Esiste un plugin maven:
https://spotbugs.readthedocs.io/en/latest/maven.html

Per un esempio di possibili bug che possono essere trovati con FindBugs vedi:
https://www.slideshare.net/caroljmcdonald/finding-bugs-that-matter-with-findbugs

## PMD

PMD is a static source code analyzer. It **finds common programming flaws** like unused variables, empty catch blocks, unnecessary object creation, copy and paste, and so forth. It’s mainly concerned with Java and Apex, but supports six other languages.

PMD features many built-in checks (in PMD lingo, rules), which are documented for each language.
https://docs.pmd-code.org/latest/pmd_release_notes_pmd7.html

Extensive API to write your own rules, which you can do either in Java or as a self-contained XPath query.

Esiste un plugin maven per pmd:
https://maven.apache.org/plugins/maven-pmd-plugin/

## SonarQube

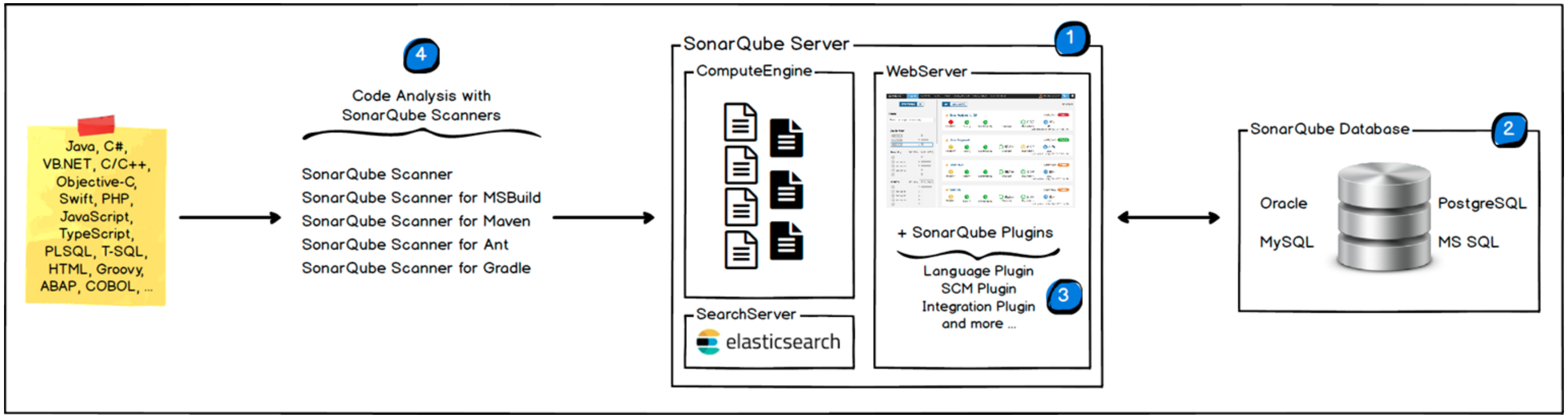
SonarQube is an automatic code review tool to detect bugs, vulnerabilities and code smells in your code. It can integrate with your existing workflow to enable **continuous code inspection** across your project branches and pull requests.

SonarQube (formerly Sonar) is an open source platform developed by SonarSource for continuous inspection of code quality to perform automatic reviews with static analysis of code to detect bugs, code smells, and security vulnerabilities on 20+ programming languages. SonarQube offers reports on duplicated code, coding standards, unit tests, code coverage, code complexity, comments, bugs, and security vulnerabilities.

SonarQube can **record metrics history and provides evolution graphs**. SonarQube's provides fully automated analysis and integration with Maven, Ant, Gradle, MSBuild and continuous integration tools (Atlassian Bamboo, Jenkins, Hudson, etc.).

WIKIPEDIA
The Free Encyclopedia

## Architettura

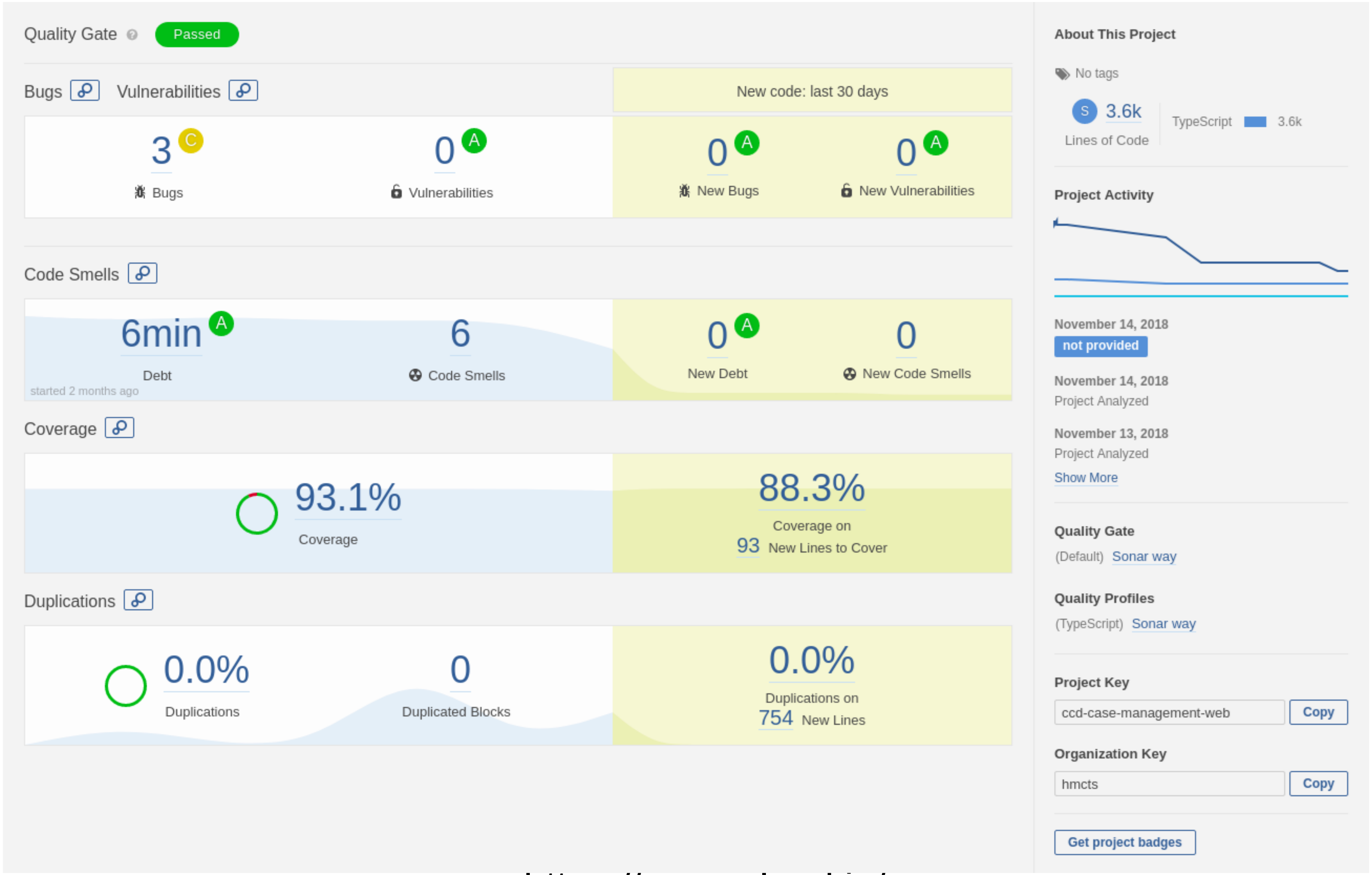

Java, C#, VB.NET, C/C++, Objective-C, Swift, PHP, JavaScript, TypeScript, PLSQL, T-SQL, HTML, Groovy, ABAP, COBOL, ...
4
Code Analysis with SonarQube Scanners
SonarQube Scanner
SonarQube Scanner for MSBuild
SonarQube Scanner for Maven
SonarQube Scanner for Ant
SonarQube Scanner for Gradle
SonarQube Server
1
ComputeEngine
WebServer
+ SonarQube Plugins
Language Plugin
SCM Plugin
Integration Plugin
and more ...
3
SearchServer
elasticsearch
SonarQube Database
2
Oracle
MySQL
PostgreSQL
MS SQL

https://docs.sonarqube.org/7.9/architecture/architecture-integration/

## Funzionalità

- **Storicizza** l’andamento della qualità
- Permette di verificare se c’è un miglioramento o un deterioramento del progetto nel tempo
- Permette di stabilire un **quality profile** (un insieme di regole) da applicare al progetto
- Permette di stabilire un **quality gate** per verificare se la qualità del progetto rispetta determinati standard
- Le **issue** segnalate vengono classificate in base alla gravità (Blocker, Critical, Major, Minor, info)
- Le issue vengono classificate in:
- **Vulnerabilità** → Permette di valutare il livello di sicurezza del progetto (security)
- **Bug** → Permette di valutare l’affidabilità del progetto (Reliability)
- **Code Smell** → Permette di valutare la mantenibilità del progetto (Maintainability)
- Permette di revisionare le issue segnalate e segnare i falsi positivi

## SonarQube



Quality Gate Passed

Bugs Vulnerabilities

New code: last 30 days

3 C Bugs | 0 A Vulnerabilities | 0 A New Bugs | 0 A New Vulnerabilities

Code Smells

6min A Debt | 6 Code Smells | 0 A New Debt | 0 New Code Smells

started 2 months ago

Coverage

93.1% Coverage | 88.3% Coverage on 93 New Lines to Cover

Duplications

0.0% Duplications | 0 Duplicated Blocks | 0.0% Duplications on 754 New Lines

About This Project

No tags

S 3.6k Lines of Code | TypeScript 3.6k

Project Activity

November 14, 2018
not provided

November 14, 2018
Project Analyzed

November 13, 2018
Project Analyzed

Show More

Quality Gate
(Default) Sonar way

Quality Profiles
(TypeScript) Sonar way

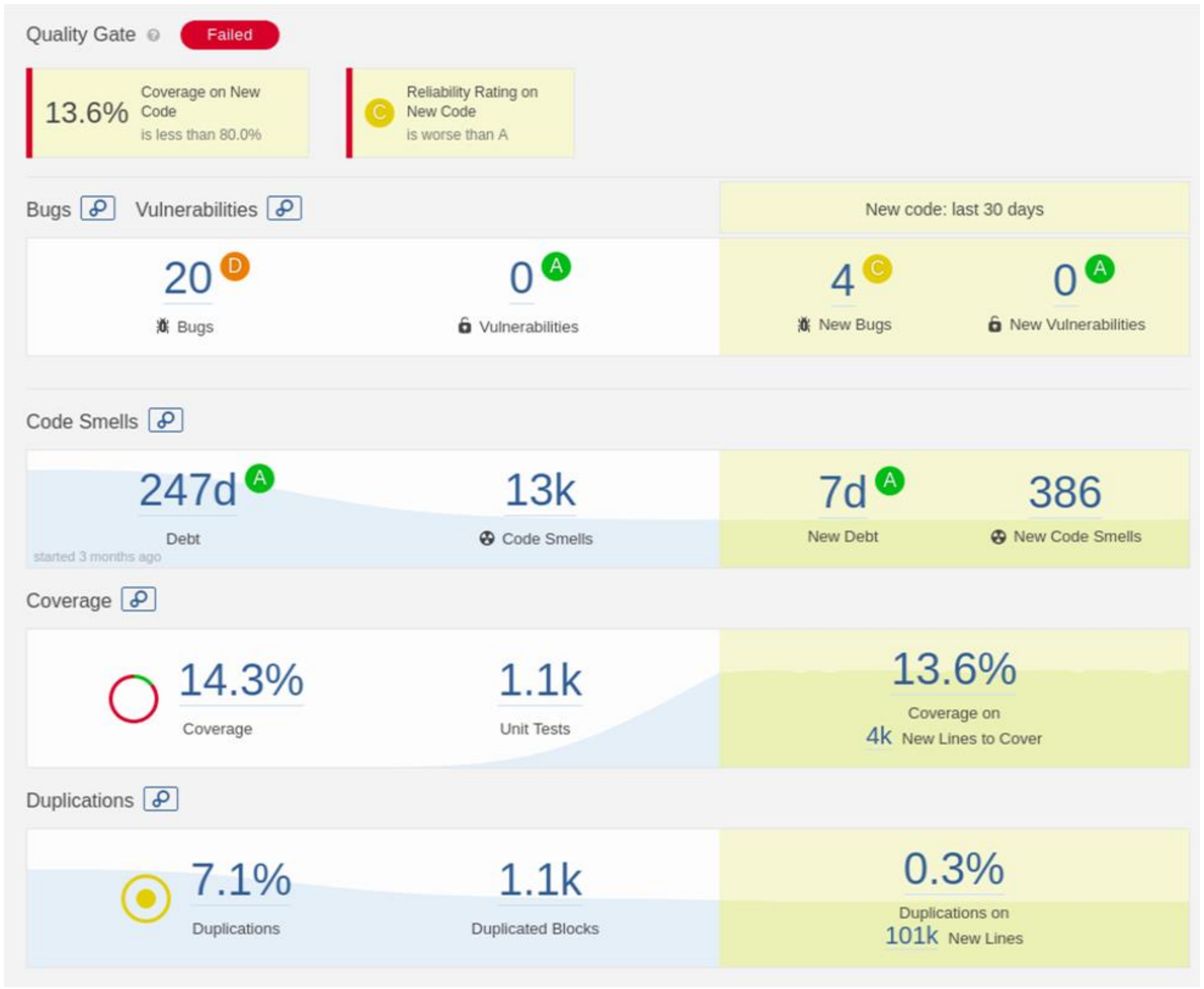
Project Key
ccd-case-management-web Copy

Organization Key
hmcts Copy

Get project badges

https://sonarcloud.io/

## SonarQube



Quality Gate ? Failed

13.6% Coverage on New Code is less than 80.0%

C Reliability Rating on New Code is worse than A

Bugs Vulnerabilities

New code: last 30 days

| | | | |
|---|---|---|---|
| 20 D Bugs | 0 A Vulnerabilities | 4 C New Bugs | 0 A New Vulnerabilities |

Code Smells

| | | | |
|---|---|---|---|
| 247d A Debt | 13k Code Smells | 7d A New Debt | 386 New Code Smells |

started 3 months ago

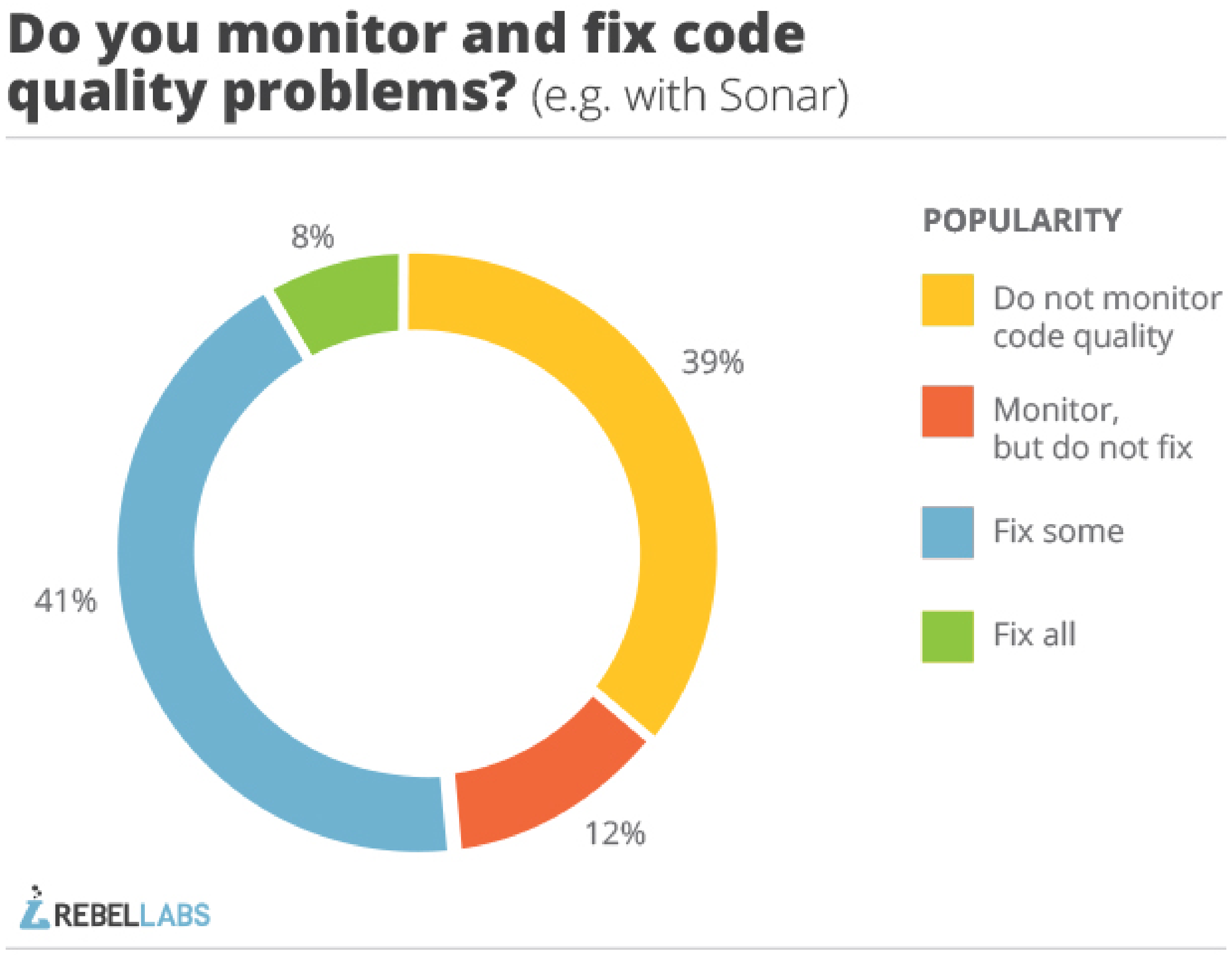
Coverage

| | | |
|---|---|---|
| 14.3% Coverage | 1.1k Unit Tests | 13.6% Coverage on 4k New Lines to Cover |

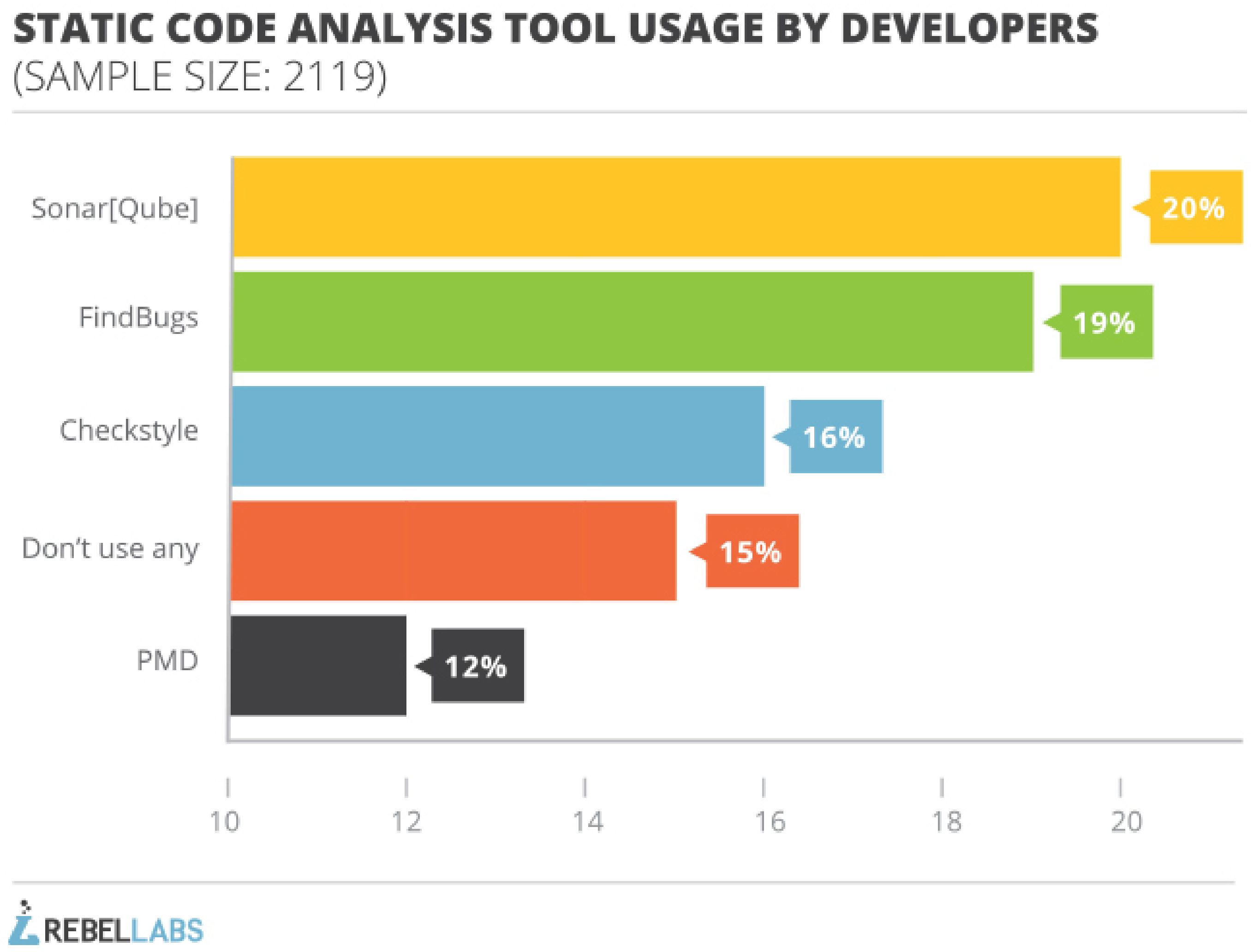
Duplications

| | | |
|---|---|---|
| 7.1% Duplications | 1.1k Duplicated Blocks | 0.3% Duplications on 101k New Lines |

https://sonarcloud.io/



https://www.jrebel.com/blog/static-code-analysis-in-java-guide


STATIC CODE ANALYSIS TOOL USAGE BY DEVELOPERS
(SAMPLE SIZE: 2119)
Sonar[Qube]
20%
FindBugs
19%
Checkstyle
16%
Don't use any
15%
PMD
12%
10
12
14
16
18
20
REBELLABS

## Riferimenti

https://en.wikipedia.org/wiki/Static_program_analysis
https://en.wikipedia.org/wiki/Automated_code_review
https://en.wikipedia.org/wiki/Broken_windows_theory
https://zeroturnaround.com/rebellabs/developers-guide-static-code-analysis-findbugs-checkstyle-pmd-coverity-sonarqube/2/
https://en.wikipedia.org/wiki/SonarQube
http://pages.zeroturnaround.com/DevelopersGuidetoStaticCode.html?utm_source=Developers%27%20Guide%20to%20Static%20Code&utm_medium=allreports&utm_campaign=rebellabs&utm_rebellabsid=87
http://checkstyle.sourceforge.net
http://findbugs.sourceforge.net
https://spotbugs.github.io
https://pmd.github.io
https://www.sonarqube.org
https://martinfowler.com/bliki/CodeSmell.html