What is code quality?

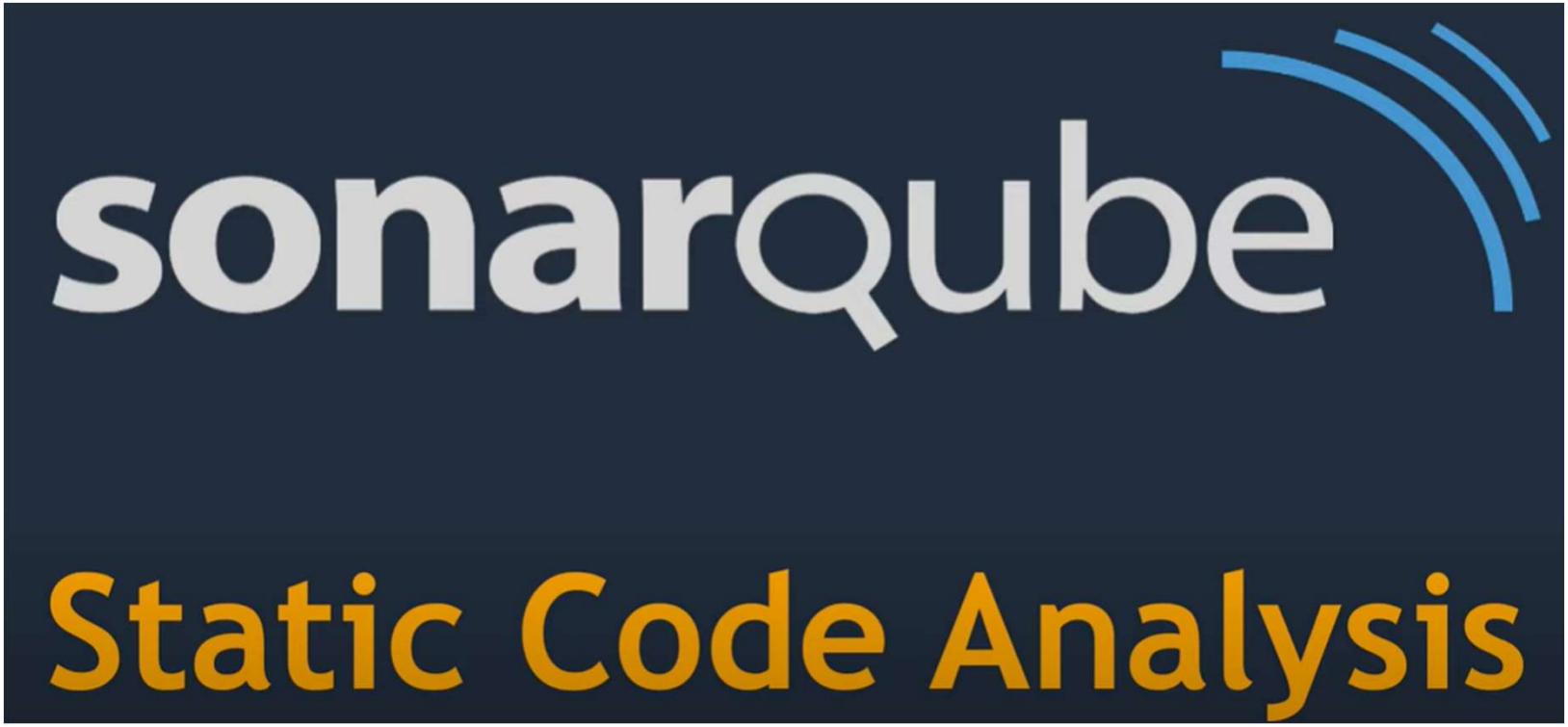
- Is it a bug free?
- Is it secure?
- Duplications avoided?
- Tested properly?
- Complex code?
- Easy to integrate with others code?

Solution: Static Code Analysis

Static Code Analysis Tools

Static Code Analysis

- Static analysis, also called static code analysis, is a method of computer program debugging that is done by examining the code without executing the program. The process provides an understanding of the code structure, and can help to ensure that the code adheres to industry standards.
- Automated tools can assist programmers and developers in carrying out static analysis. The process of scrutinizing code by visual inspection alone (by looking at a printout, for example), without the assistance of automated tools, is sometimes called program understanding or program comprehension.


sonarqube
Static Code Analysis

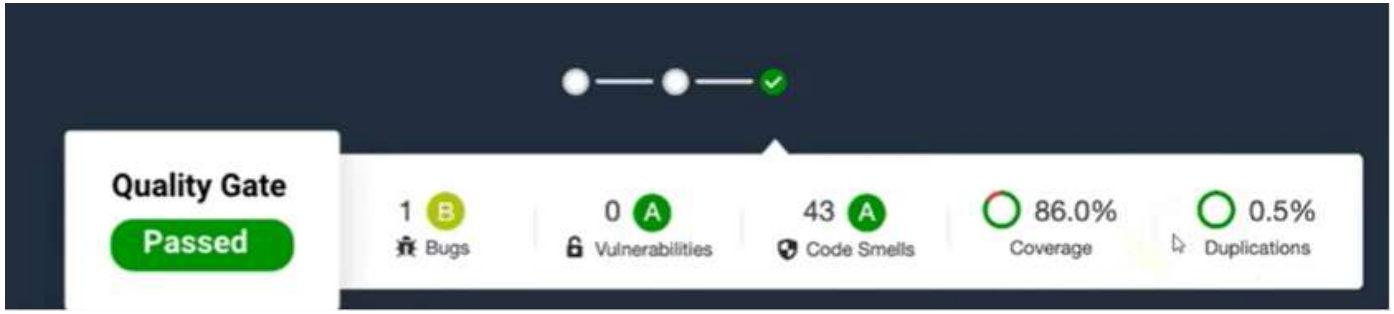
SonarQube

- It is a static testing open source tool developed by SonarSource for continuous inspection of code quality, perform automatic detection of static analysis of code to detect bugs, code smells, and security vulnerabilities on 25+ programming languages. SAST- Static Application Security Testing.
- SonarQube offers reports on duplicated code, coding standards, unit tests, code coverage, code complexity, comments, bugs, and security vulnerabilities.

WHY SonarQube?

Its act as "**Quality Management Tool**"

- Code Analysis
- Test Reports



Components of SonarQube

- SonarQuber Server
  - Rules
  - Database
  - Web Interface
- SonarScanner
  - SourceCode

Sonar Scanner

- Sonar Scanner is a separate client type application that in connection with the SonarQube server will run project analysis and then send the results to the SonarQube server to process it.

How SonarQube Works!

Developer

Sonarscanner collects required info from source code

SonarQube Server

How SonarQube Works!

Developer

Rules

Sonarscanner collects required info from source code

Gather applicable rules

Generate reports

SonarQube Server

How SonarQube Works!

Developer

Rules

Analysis Report

SonarQube Server

Sonarscanner collects required info from source code

Gather applicable rules

Generate reports

Stores reports on DB with help of SonarQube

Displays Graps on GUI

Technical Terms

- Quality Gate : Default or custom
- Reliability / Bugs : Error or Bugs in the code
- Security / Vulnerabilities :Security bugs or vulnerabilities detection
- Maintainability / Code Smells : Bad smell in code
- Coverage : coverage of the unit / integration test cases

Unit Testing test each part of the program and shows that the individual parts are correct, whereas Integration Testing combines different modules in the application and test as a group to see they are working fine. Unit Testing starts with the module specification, while Integration Testing starts with interface specification.

- Unit testing is performed first of all testing processes. Integration testing is performed after unit testing and before system testing. Unit testing is a white box testing. Integration testing is a black box testing. Unit testing is basically performed by the developer.

- Duplications : duplicate lines of code/ function/ block/ files
- Size: lines / classes / comments /files on code.
- Languages: programming languages
- Complexity : difficult to understand

Prerequisite :

1. JAVA 1.11 version or higher version
2. SonarQube latest version
3. Sonar Scanner latest version
4. Set path for SonarQube and Sonar Scanner