## **What Is SonaRQube**

* **Code Analysis**
* **Test Reports**
* **Code Coverage**
* **Quality Gate**
* SonarQube is a Code Quality Assurance tool that collects and analyzes source code, and provides reports for the code quality of your project.
* It combines static and dynamic analysis tools and enables quality to be measured continually over time. Everything from minor styling choices, to design errors are inspected and evaluated by SonarQube. This provides users with a rich searchable history of the code to analyze where the code is messing up and determine whether or not it is styling issues, code defeats, code duplication, lack of test coverage, or excessively complex code.
* The software will analyze source code from different aspects and drills down the code layer by layer, moving module level down to the class level, with each level producing metric values and statistics that should reveal problematic areas in the source code that needs improvement.
* Sonarqube also ensures code reliability, Application security, and reduces technical debt by making your code base clean and maintainable. Sonarqube also provides support for 27 different languages, including C, C++, Java, Javascript, PHP, GO, Python, and much more.SonarQube also provides Ci/CD integration, and gives feedback during code review with branch analysis and pull request decoration.

**When creating software, the code should have the following characteristics:**

* The code should follow a specific convention
* The code should be following established good practices and have been followed
* Checked for potential bugs and performance, security, or vulnerabilities issues
* Is the code duplicated anywhere
* Does the code make logical sense, or is it too complex
* Does the public API have good documentation and comments
* Does the code have unit tests
* Does the code follow good software design and architecture principles.

**Without SonarQube**

* Make sure to achieve Quality of the code.
* Manage Analysis report separately.
* Difficult in maintaining pool of logical errors etc
* Have to install multiple plugins in Ide to get the analysis and report.
* Apply Rules on code while running [Dynamic]
* ¼ th time has to given time for testing and to make quality product.

**MeasureMents of Quality Product**

Quality

Internal : Coding && Best Practices

Variuos Plugins to install

**UcDetector | CheckStyle | FindBugs | Emma**

We can enforce these coding standards automatically by two methods: **Static code analysis or Dynamic code analysis.** To explain them quickly:

## **Dynamic code analysis**

**During Executing of Code :: Selenium | CodeCoverage | CasperJS | Sahi**

Dynamic Code Analysis relies on studying how the code behaves during execution. The objective is to find errors in a program while it is running, rather than by repeatedly examining the code offline. Some things that Dynamic code analysis does are:

* Code Coverage: Computing how much a piece of code gets tested by test suites
* Memory error detection: Checking whether or not memory leaks or errors occur
* Fault localization: Locating the buggy code to a specific location
* Invariant Inference: Observes the values that the program computes, and then report properties that were true over the observed executions, and this likely true over all executions.
* Security Analysis: Detect security problems.
* Concurrency errors: Dynamic Uses runtime error detection to expose defects such as race conditions, exceptions, resource and memory leaks, and security attack vulnerabilities
* Program slicing: Consists of reducing the program to the minimum form that still produces the selected behavior.
* Performance Analysis: dynamically tracing software applications at runtime and captures data that can be used to analyze and identify the causes of poor performance.

## **Static Code Analysis**

Test the code without the Execution

**SonarQube | Apache-Yatus | CodeDx | CodeSense**

Static code analysis is done without executing any of the code. It is a collection of algorithms and techniques to analyze source code to automatically find potential errors and poor coding practices. This is done with compiler errors and run-time debugging techniques such as white box testing. **Static code analysis is also considered a way to automate code review process**. The tasks involved in static code analysis can be divided as such:

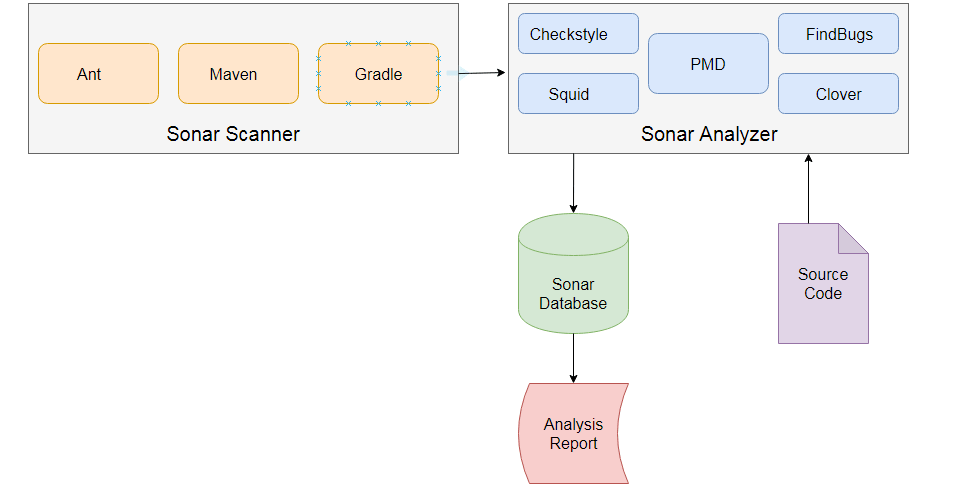
**Drills down from top to bottom ie module to package level, package to class level, class to function level.**

* Detecting errors in programs
* Recommendations on code formatting with a formatter
* Metrics computation, which gives you back a rating on how well your code is.
* Early detection of code vulnerabilities reduces the cost of debugging and fixing at a later stage in the SDLC.
* It is conducted by trained security engineers who have strong knowledge about secure coding practices.
* It is a highly scalable method which means it can run on multiple code bases and can be run repeatedly.
* It is a fast and efficient way to identify vulnerabilities in the code such as SQL Injection, buffer overflow, etc.
* Automated tools can scan the entire application source code with minimal effort.
* The use of automated tools helps provide mitigation recommendations, thereby reducing research time.

Popular tools for static Code Analysis are Checkstyle, PMD, and FindBugs.

## **SonarQube Architecture**

[**https://www.codeusingjava.com/sonar/sonarqube-architecture**](https://www.codeusingjava.com/sonar/sonarqube-architecture)[**https://www.partech.nl/nl/publicaties/2020/11/introduction-to-sonarqube#**](https://www.partech.nl/nl/publicaties/2020/11/introduction-to-sonarqube)



**1. Sonar Scanner**

* Previously known as sonar runner.
* Ideally you need to use any of the built-in tools such as ant, maven or gradle that invokes the Sonar Scanner to fetch the source code.
* Ideally you need to use any of the built-in tools such as ant, maven or gradle that invokes the Sonar Scanner to fetch the source code.
  + - It takes the source code from the repository or a configuration parameter is used to set the place from which you want the source code to be taken. SonarQube cube supports plugins like SVN and some of the other version control system such as git etc. You can go straight ahead and incorporate these stuff with any build tools that are the integration tool, so you don't have to offer the direction every time, because the build tool automatically pulls the software from the database.
    - It is a tool for performing code analysis. It is executed in the root directory of Project. We also need to [configure SonarQube](javascript:void()) under Project , which specifies the relevant information of the project and the address of SonarQube Server. Report the results to the server.

**2. Source Code**

* The source code is the code written by developer or manager.
* The code is then pushed to repository.
* Pull Requests (PRs) are visible in SonarQube

**3. Sonar Analyzer**

* The sonar analyzer takes the source code you would like to analyze
* Sonar analyzer takes the source code and go through all the code and gives you technical problems
* Sonar receives the request and starts to examine the source code of the project. The analysis is based on the project's Sonar profile activating any additional plugins or reporting capabilities, if any.
* You simply need to integrate these plugins into your sonar qube and the sonarqube will analyze the source code.
* You need to add these plugins to the sonar analyzer based on the language you want to analyze.
* When the analysis is finished, the results will be stored in database for future reference
* Sonar analyzers come with thousands of rules on 25+ programming languages

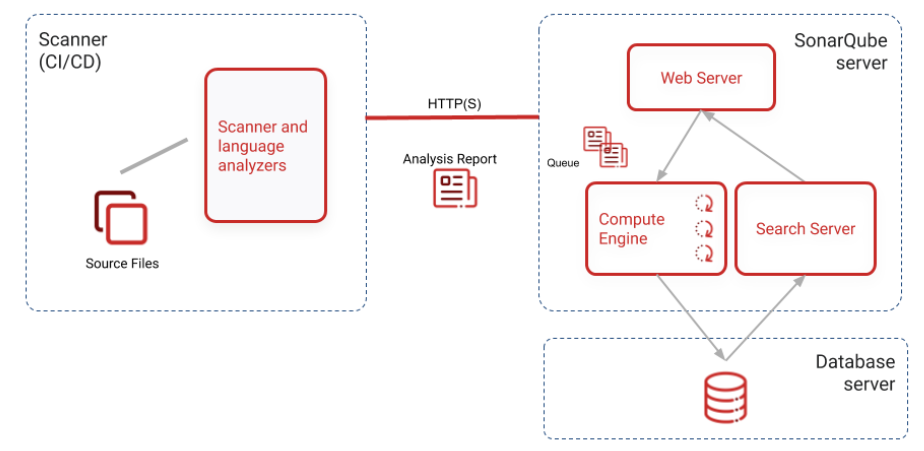
**4. SonarQube Database**

* It will be sent to the database once the report is generated so that it can be stored in the database and subsequently used for reporting purposes
* There is a caching server which is used to hold the temporary reports in the cache
* The main advantage of caching server is, it is going to maintain the reports in the cache to reduce the calls to the database as the result you can see high performance
* Once the sonar analyzer completes the analysis of the source code then it is going to send the this report to the queue where the server needs to be process those results
* After the results are processed then the report or the data will be stored in the database. SonarQube will have a default database
* You can integrate your own database our to your sonarqube
* Sonarqube supports various databases like a ms SQL Server, Oracle etc.

**WorkFlow**Developers code in their IDEs and use [SonarLint](https://sonarlint.org/" \t "_blank) to run local analysis.

1. Developers push their code into their favourite SCM : git, SVN, TFVC, ...
2. The Continuous Integration Server triggers an automatic build, and the execution of the SonarScanner required to run the SonarQube analysis.
3. The analysis report is sent to the SonarQube Server for processing.
4. SonarQube Server processes and stores the analysis report results in the SonarQube Database, and displays the results in the UI.
5. Developers review, comment, challenge their Issues to manage and reduce their Technical Debt through the SonarQube UI.
6. Managers receive Reports from the analysis. Ops use APIs to automate configuration and extract data from SonarQube. Ops use JMX to monitor SonarQube Server.

**Sonar component**



1. The SonarQube server running the following processes:
   * a web server that serves the SonarQube user interface.
   * a search server based on Elasticsearch.
   * the compute engine in charge of processing code analysis reports and saving them in the SonarQube database.
2. The database to store the following:
   * Metrics and issues for code quality and security generated during code scans.
   * The SonarQube instance configuration.
3. One or more scanners running on your build or continuous integration servers to analyze projects.

**WorkFlow2**

The moment you run the sonar scanner on the source code , scanner download the rules from sonarqube , apply on the source code and you get the report , these report scanner will send to the database , and it publish the report to the sonardasboard.

## **Installation**

**PreRequisites**

The only prerequisite for running SonarQube is to have Java (Oracle JRE 11 or OpenJDK 11) installed on your machine.

**To run sonarqube as a service [Windows]**

Execute ths command in cmd  
Sonar 9 installation

sc create SonarNew binPath= "\"C:\Program Files\sonarqube-9.1\bin\windows-x86-64\wrapper.exe\" -s \"C:\Program Files\sonarqube-9.1\conf\wrapper.conf\""

## **What Is SonaRQube**