**What is SonarQube and How it works? An Overview and Its Use Cases**

**History & Origin of SonarQube?**

**Simon Brandhof starts developing the Sonar platform by integrating best-of-breed open source tools for Java**. The two of them are joined in September 2007 by Olivier Gaudin, who was enthused by the Sonar platform’s vision and starts contributing to it.

Back in **2007**, when first lines of code were created, the founders of SonarQube (originally called Sonar) had a dream to one day provide every developer the ability to measure the code quality of his projects. Their motto: “Continuous Inspection must become mainstream as Continuous Integration”

Covering [27 programming languages](https://www.sonarqube.org/features/multi-languages/), while pairing-up with your existing software pipeline, SonarQube provides clear remediation guidance for developers to understand and fix issues and for teams overall to deliver better, safer software. With over 170,000 deployments, helping small development teams as well as global organizations, SonarQube provides the means for all teams and companies, around the world, to own and impact their Code Quality and Security.

Back in 2007, when first lines of code were created, the founders of SonarQube (originally called Sonar) had a dream to one day provide every developer the ability to measure the code quality of his projects. Their motto: “Continuous Inspection must become mainstream as Continuous Integration”.

**What is SonarQube?**

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.

**Overview**

Once the SonarQube platform has been installed, you’re ready to install a scanner and begin creating projects. To do that, you must install and configure the scanner that is most appropriate for your needs. Do you build with:

* Gradle – [SonarScanner for Gradle](https://docs.sonarqube.org/latest/analysis/scan/sonarscanner-for-gradle/)
* .NET – [SonarScanner for .NET](https://docs.sonarqube.org/latest/analysis/scan/sonarscanner-for-msbuild/)
* Maven – use the [SonarScanner for Maven](https://docs.sonarqube.org/latest/analysis/scan/sonarscanner-for-maven/)
* Jenkins – [SonarScanner for Jenkins](https://docs.sonarqube.org/latest/analysis/scan/sonarscanner-for-jenkins/)
* Azure DevOps – [SonarQube Extension for Azure DevOps](https://docs.sonarqube.org/latest/analysis/scan/sonarscanner-for-azure-devops/)
* Ant – [SonarScanner for Ant](https://docs.sonarqube.org/latest/analysis/scan/sonarscanner-for-ant/)
* anything else (CLI) – [SonarScanner](https://docs.sonarqube.org/latest/analysis/scan/sonarscanner/)

**Why should we use SonarQube?**

SonarQube **reduces the risk of software development within a very short amount of time**. It detects bugs in the code automatically and alerts developers to fix them before rolling it out for production. SonarQube also highlights the complex areas of code that are less covered by unit tests

**How SonarQube works aka SonarQube architecture?**

1. One SonarQube Server starting 3 main processes:
   * Web Server for developers, managers to browse quality snapshots and configure the SonarQube instance
   * Search Server based on Elasticsearch to back searches from the UI
   * Compute Engine Server in charge of processing code analysis reports and saving them in the SonarQube Database
2. One SonarQube Database to store:
   * the configuration of the SonarQube instance (security, plugins settings, etc.)
   * the quality snapshots of projects, views, etc.
3. Multiple SonarQube Plugins installed on the server, possibly including language, SCM, integration, authentication, and governance plugins
4. One or more SonarScanners running on your Build / Continuous Integration Servers to analyze projects

**Integration**

The following schema shows how SonarQube integrates with other ALM tools and where the various components of SonarQube are used.

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

**About Machines and Locations**

* The SonarQube Platform cannot have more than one SonarQube Server (although the Server can be installed [as a cluster](https://docs.sonarqube.org/8.1/setup/install-cluster/)) and one SonarQube Database.
* For optimal performance, each component (server, database, scanners) should be installed on a separate machine, and the server machine(s) should be dedicated.
* SonarScanners scale by adding machines.
* All machines must be time synchronized.
* The SonarQube Server and the SonarQube Database must be located in the same network
* SonarScanners don’t need to be on the same network as the SonarQube Server.
* There is **no communication** between **SonarScanners** and the **SonarQube Database**.

**Use case of  SonarQube?**

SonarQube is used **as part of the build process (Continuous Integration and Continuous Delivery) in all Java services to ensure a high quality of code and remove bugs that can be found during static analysis**.

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.

**How to Use SonarQube Tool For Code Quality:**

1. Step 1: Download and Unzip SonarQube. Prerequisites: Java (Oracle JRE11 or OpenJDK 11 minimum) …
2. Step 2: Run the SonarQube local server. …
3. Step 3: Start a new SonarQube project. …
4. Step 4: Setup Project properties and SonarScanner. …
5. Step 5: View your analysis report on Sonar Dashboard.

How do you write test cases in SonarQube?

**Importing .** **NET reports**

1. Run the SonarScanner. …
2. Build your project using MSBuild.
3. Run your test tool, instructing it to produce a report at the same location specified earlier to the MSBuild SonarQube Runner (How to generate reports with different tools)
4. Run the SonarScanner.

**Feature and Advantage of using SonarQube**

SonarQube platform **significantly increases the lifetime of applications by reducing complexities, duplications and potential bugs in the code, by keeping neat and clean code architecture and increased unit tests**. SonarQube increases maintainability of the software. It also has the ability to handle changes.

**Benefits of SonarQube**

* Sustainability – Reduces complexity, possible vulnerabilities, and code duplications, optimising the life of applications.
* Increase productivity – Reduces the scale, cost of maintenance, and risk of the application; as such, it removes the need to spend more time changing the code.

**What is SonarQube and its features?**

SonarQube is a **Code Quality Assurance tool that collects and analyzes source code**, and provides reports for the code quality of your project. … Sonarqube also ensures code reliability, Application security, and reduces technical debt by making your code base clean and maintainable.

**Best Alternative of SonarQube**

Browse options below. Based on reviewer data you can see how SonarQube stacks up to the competition, check reviews from current & previous users in industries like Information Technology and Services, Computer Software, and Financial Services, and find the best product for your business.

**Top 10 Alternatives to SonarQube**

* Embold.
* GitHub.
* Coverity.
* Checkmarx.
* Klocwork.
* GitLab.
* Veracode Application Security Platform.
* Kiuwan Code Security & Insights.

[Please Click here more Top 10 **Alternatives to SonarQube**](https://www.g2.com/products/sonarqube/competitors/alternatives)

**Best Resources, Tutorials and Guide for SonarQu**

1. [devopsschool.com](https://www.devopsschool.com/courses/sonarqube/)
2. [udemy.com](https://www.udemy.com/course/sonarqube-the-complete-guide-2021-edition/)
3. [devopsuniversity.org](https://www.devopsuniversity.org/sonarqube-setup-guide/)