**SonarQube**

**Q1. What is Software quality measurement?**

Ans. Software quality measurement is a quantitative process summing up weighted attribute values, which in part describe specific software characteristics. For each characteristic, a set of such measurable attributes is defined.

Well it could be:

* Whether the coding has been done following a specific convention.
* Whether well-known/established good practices have been followed and well-known/established bad practices have been avoided.
* Are there any potential bugs and performance issues, security vulnerabilities?
* Is there any duplicate code?
* Is the code logic very complex?
* Whether the public API has good documentation and comments.
* Whether the code has unit tests.
* Whether the code follows good design and architecture principles.

After having the software characteristics defined, the next question which comes to our mind is how do we enforce it automatically? The answer lies in static code analysis.

2. **Static Code Analysis:**

Static code analysis is a collection of algorithms and techniques used to analyze source code in order to automatically find potential errors or poor coding practices.

**2.1. Static code analysis tools in market**

* Conventions (Checkstyle)
* Bad practices (PMD)
* Potential bugs (FindBugs)
* HP Fortify (Security Vulnerabilities)

3. **What is PMD (Programming Mistake Detector)?**

* An extensible cross-language static code analyzer.
* It finds common programming flaws like unused variables, empty catch blocks, unnecessary object creation, and so forth. It includes CPD, the copy-paste-detector.

**Some of the features offered by PMD are:**

* Supports multiple languages.
* Enforce a coding standard for your codebase.

**4. What is SonarQube?**

* Continuous Code Quality tool.
* SonarQube collects and analyzes source code, measuring quality and providing reports for your projects.
* It combines static and dynamic analysis tools and enables quality to be measured continuously over time.
* Everything that affects our code base, from minor styling details to critical design errors, is inspected and evaluated by SonarQube.
* it enables the developer to access and track code analysis data ranging from styling errors, potential bugs, and code defects to design inefficiencies, code duplication, lack of test coverage, and excess complexity.
* The Sonar platform analyzes source code from different aspects and hence it drills down to your code layer by layer, moving from the module level down to the class level.
* At each level, SonarQube produces metric values and statistics, revealing problematic areas in the source that require inspection or improvement.

**Features of SonarQube->**

* SonarQube addresses not just bugs but also coding rules, test coverage, duplications, API documentation, complexity, and architecture, providing all these details in a dashboard.
* It gives you a moment-in-time snapshot of your code quality today, as well as trends of lagging (what's already gone wrong) and leading (what's likely to go wrong in the future) quality indicators.
* It provides you with metrics to help you take right decision. In nearly every industry, serious leaders track metrics. Whether it's manufacturing defects and waste, sales and revenue, or baseball hits and RBIs, there are metrics that tell you how you're doing: if you're doing well overall, or whether you're getting better or worse.
* It adds values to real business by provide you risks and technical debt of application.
* SonarQube not only addresses core developers and programmers but, project managers and even higher managerial levels due to the management aspect it offers.
* a managerial perspective, transparent and continuous access on historical data enables the manager to ask the right questions.
* Multi-language.
* Detect tricky issues.
* Security analysis.
* A biggest advantage of Sonar is, it generates visual reports in, "Dashboard" form.

**Sonar Include plugins:**

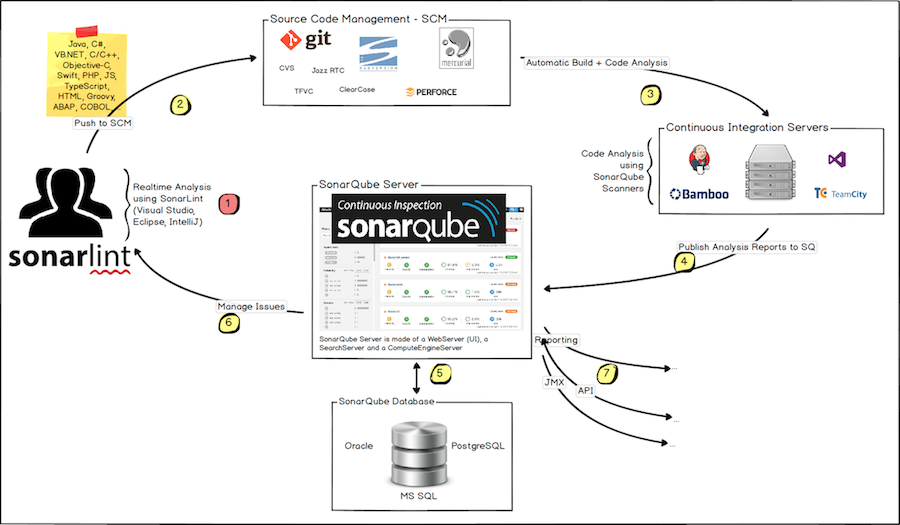
* Conventions (Checkstyle)
* Bad practices (PMD)
* Potential bugs (FindBugs)
* Cobertura (code coverage)
* HP Fortify (Security Vulnerabilities)

**Plugin Available of Sonar for Different IDE**

* Radar-netbeans (NetBean Plugin)
* Sonarlint (For Eclipse)

**5. Integration**

The following schema shows how SonarQube integrates with other **Application lifecycle management** (ALM) tools and where the various components of SonarQube are used.



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

**6.** **Installation Guide:**

Prerequisite

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

**Installing from a zip file**

By Following are the steps, to setup SonarQube.

1. [Download](https://www.sonarqube.org/downloads/) the SonarQube Community Edition.
2. As a **non-root user**, unzip it, let's say in *C:\sonarqube* or */opt/sonarqube*.
3. As a **non-root user**, start the SonarQube Server:

# On Windows, execute:

1. C:\sonarqube\bin\windows-x86-xx\StartSonar.bat
2. Log in to [http://localhost:9000](http://localhost:9000/) with System Administrator credentials (login=admin, password=admin).
3. **Integrate SonarQube with Cunify:**

Follow the below steps to setup:

* Download sonarqube-ant-task-2.7.0.1612.jar
* Install Ant on local machine.
* Add ANT path in system.
* Edit build-impl.xml file of CUNify project. Add below line.

<project xmlns:webproject1="http://www.netbeans.org/ns/web-project/1" xmlns:webproject2="http://www.netbeans.org/ns/web-project/2" xmlns:webproject3="http://www.netbeans.org/ns/web-project/3" basedir=".." default="default" name="MainDev-impl" **xmlns:sonar="antlib:org.sonar.ant"** >

--------------

-------------------

-------------

<!-- sonar configuration here -->

<property name="sonar.host.url" value="http://localhost:9000" />

<!-- Define the SonarQube project properties -->

<property name="sonar.projectKey" value="Cunify" />

<property name="sonar.projectName" value="Cunify" />

<property name="sonar.projectVersion" value="1.0" />

<!- - Add the path of source code for the project -->

<property name="sonar.sources" value="C:\\Development\\myprojects\\MainDev\\CUnify\\TCCUSsme\\WEB-INF\\classes" />

<!- - Add the path of .class file for the project -->

<property name="sonar.java.binaries" value="C:\\Development\\myprojects\\MainDev\\CUnify\\build\\web\\WEB-INF\\classes" />

<!- - Add the path of lib folder for the project -->

<property name="sonar.java.libraries" value="C:\\Development\\myprojects\\MainDev\\CUnify\\build\\web\\WEB-INF\\lib\\\*.jar" />

<!-- Define SonarScanner for Ant Target -->

<target name="sonar">

<taskdef uri="antlib:org.sonar.ant" resource="org/sonar/ant/antlib.xml">

<!—Add the Path where the sonar scanner jar located -->

<classpath path="C:\\Development\\myprojects\\MainDev\\CUnify\\nbproject\\sonarqube-ant-task-2.7.0.1612.jar" />

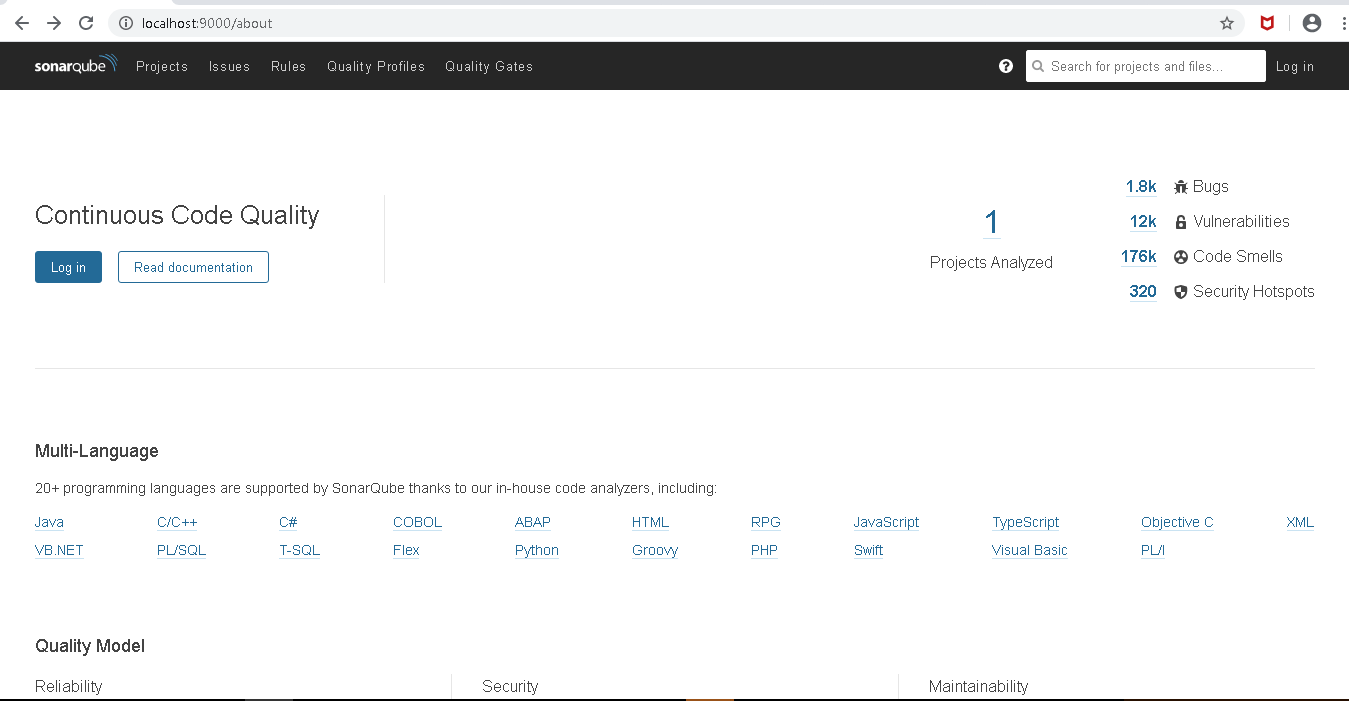
</taskdef>

<!-- Execute SonarScanner for Ant Analysis -->

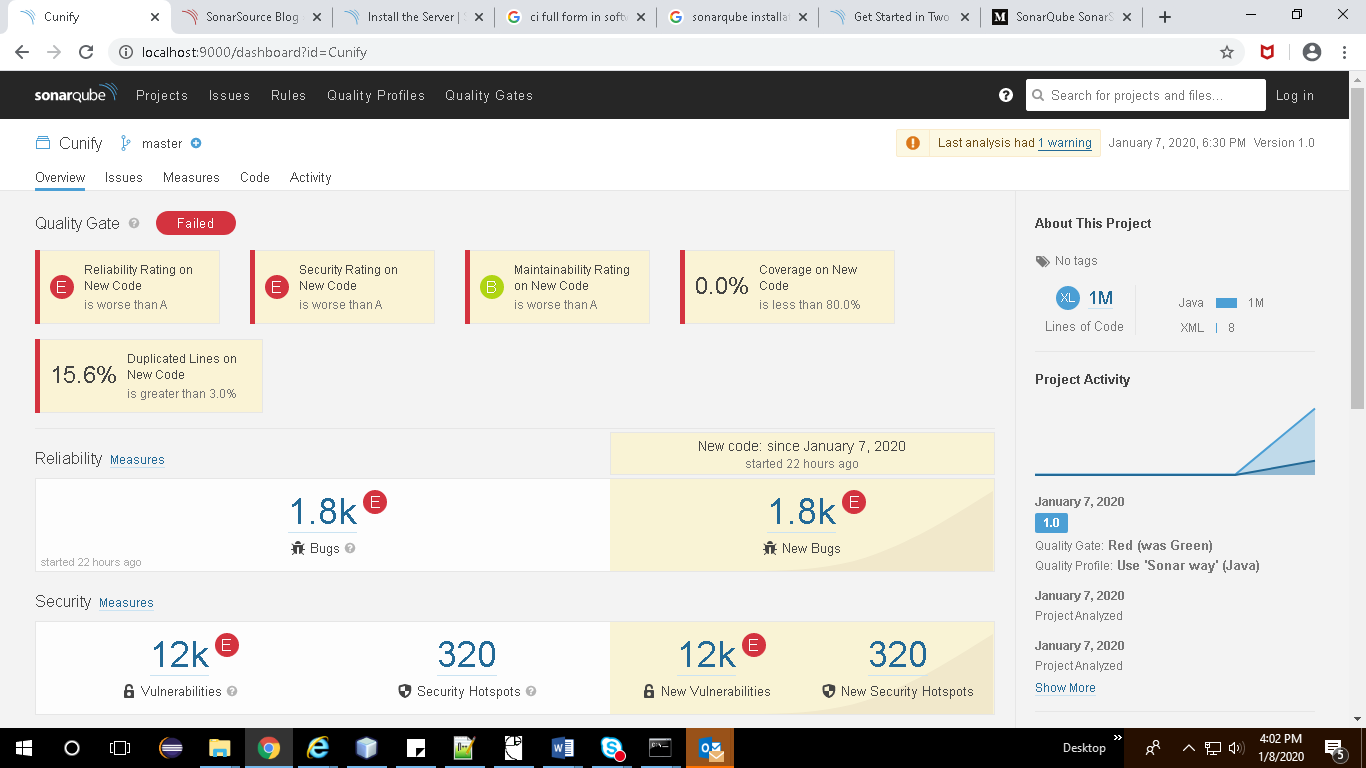
<sonar:sonar />

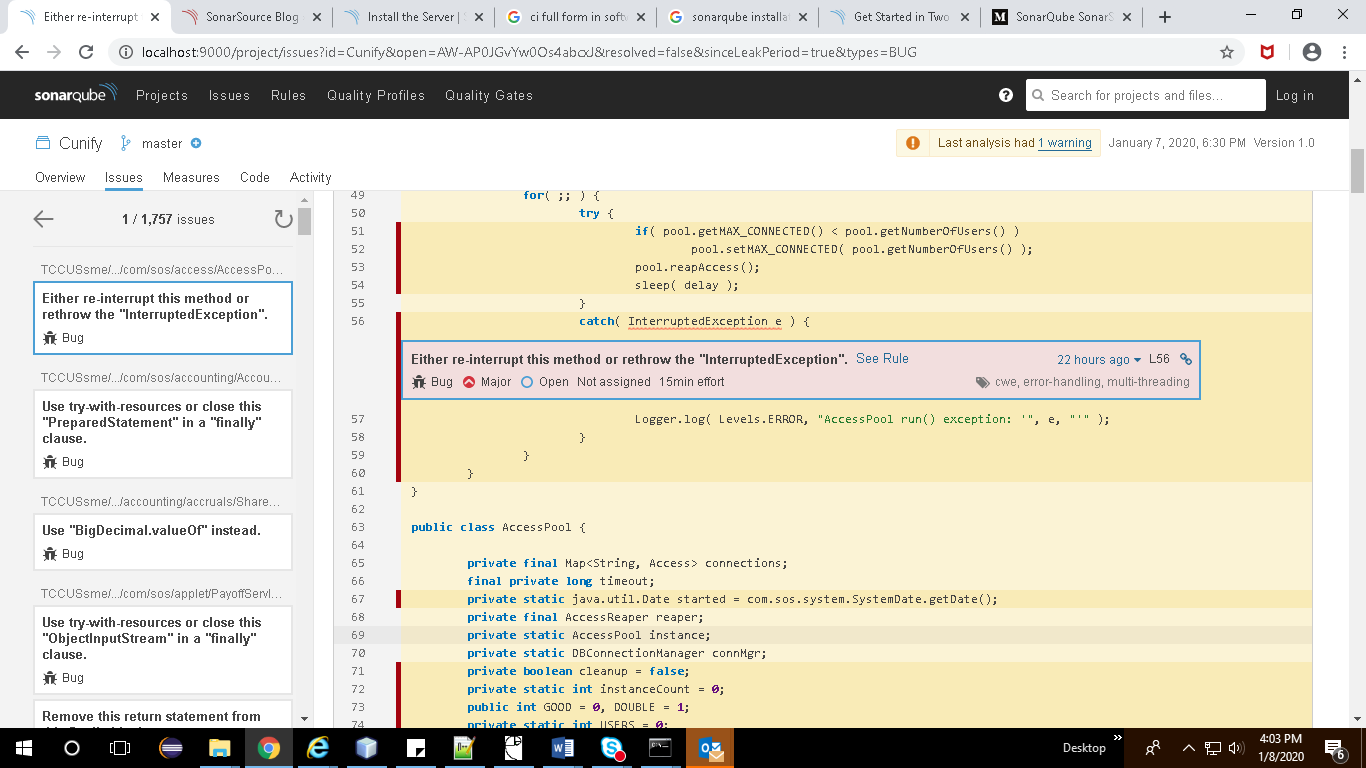
</target>

* clean and build the project.
* Go to project root folder (Example C:\Development\myprojects\MainDev) and run Command in CMD.
* ant sonar
* Open [http://localhost:9000](http://localhost:9000/) and refresh the browser. The Project will be listed under project.



* Click on Project Analyzed, the report will show.





References:

* <https://docs.sonarqube.org/>
* SonarQube in Action (Book).
* <https://medium.com/@amitvermaa93/sonarqube-setup-windows-e6a6c01be025>
* <https://www.baeldung.com/sonar-qube>
* <https://dzone.com/articles/how-quickly-get-started-sonar>

Prepared By:

Anuj Pachauri

Amit Kumar Singh