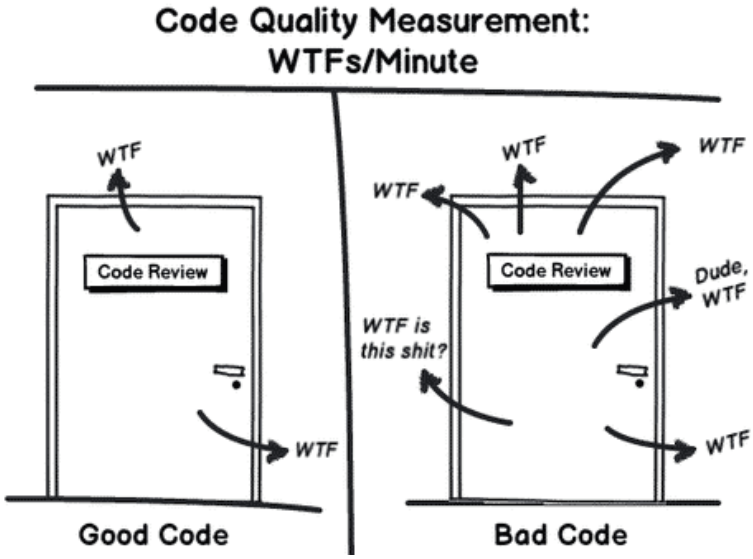
SonarQube Tutorial:

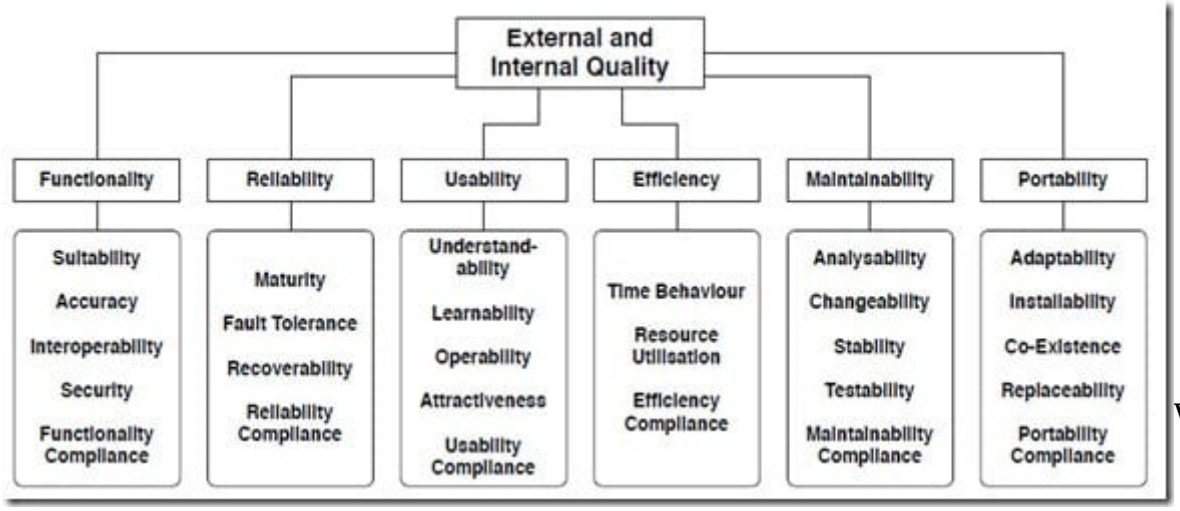
==========

What is code quality ?
What is SonarQube?
Why SonarQube?
How Sonarqube works ?
Sonar Structure & CI
Sonarqube Feature
Installation of SonarQube

What is Code Quality ?

"Code quality is an indicator about how quickly developers can add business value to software system"





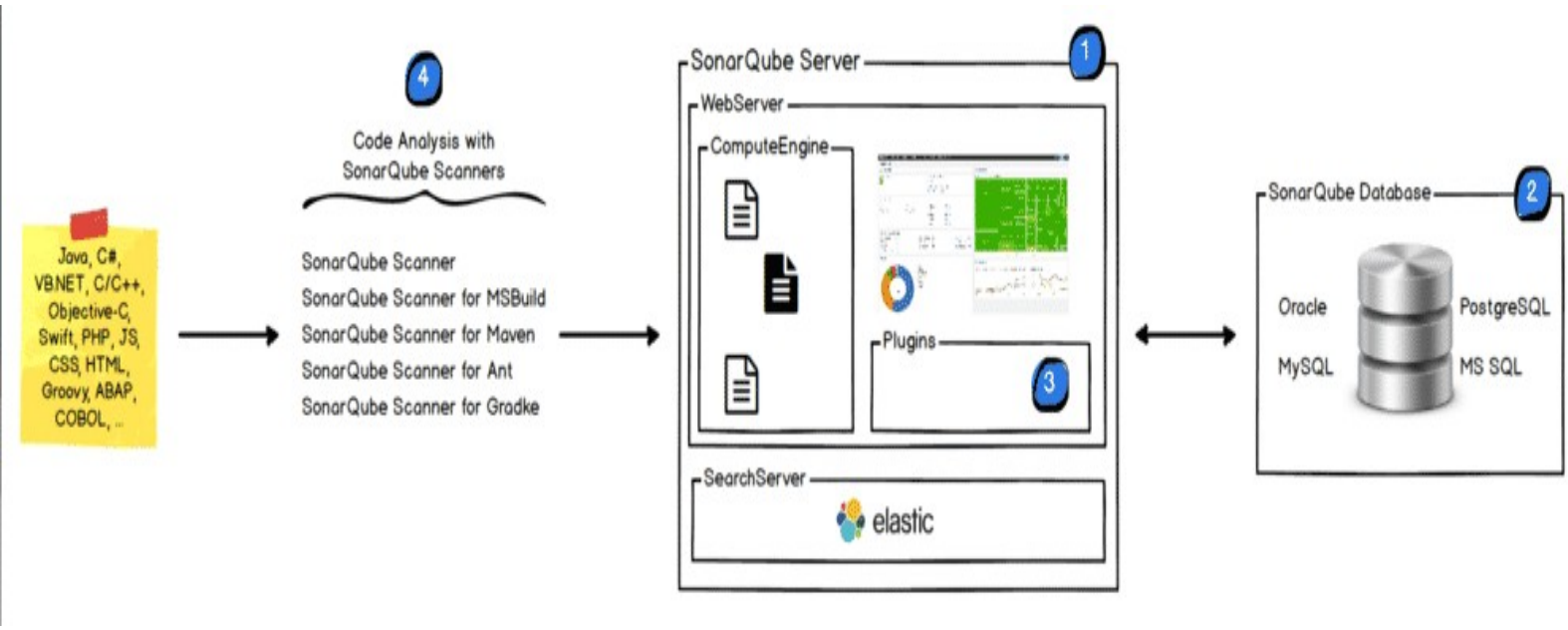
What is SonarQube ?

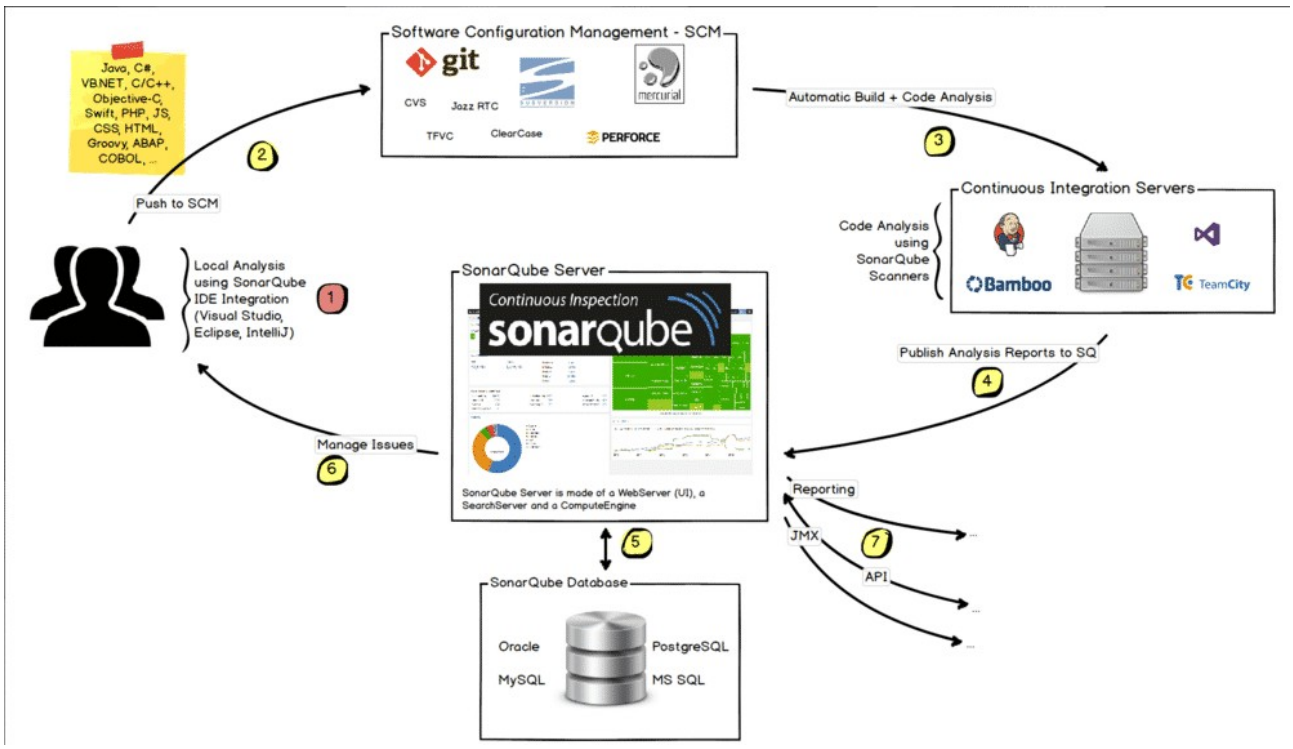
Sonar is an open source software quality platform.

SonarQube saves the calculated measures in a database and showcases them in a rich web-based dashboard. Provides trends and leading indicators.

How Sonar Works ?

Sonar uses various static & dynamic code analysis tools such as Checkstyle, PMD, FindBugs , FxCop , Gendarme and many more to extract software metrics, which then can be used to improve software quality. Provides lots of plugins.





SonarQube Features:

Supports languages: Java, C/C++, Objective-C, C#, PHP, Flex, Groovy, JavaScript, Python, PL/SQL, COBOL, etc.

Can also be used in Android development.

Offers reports on duplicated code, coding standards, unit tests, code coverage, code complexity, potential bugs,comments and design and architecture.

Records metrics history and provides evolution graphs (“time machine”) and differential views.

Provides fully automated analyses: integrates with Maven, Ant, Gradle and continuous integration tools
(Atlassian Bamboo, Jenkins, Hudson, etc.).

Integrates with the Eclipse development environment

Integrates with external tools: JIRA, Mantis, LDAP, Fortify, etc.

Is expandable with the use of plugins.

What is static code analysis ?

Computer code that is performed without actually executing programs.
Source code will be checked for compliance with a predefined set of rules or best practices set by the organization.

Technical debt is caused by the 7 deadly sins of the developer:

- Duplications: SonarQube has a copy/paste detection engine for to find duplications
- Bad distribution of complexity: Cyclomatic complexity [wiki] (or McCabe metric)
- Spaghetti Design
- Lack of unit tests
- No coding standards
- Potential bugs
- Not enough or too many comments or incorrect comments
- Cyclomatic Complexity

Configuration hello world sonarcube:

1. download sonarqube-6.7.7.zip and extract (must have configure JAVA_HOME) run sonar.sh sonarcube start

2. Hello world

```
public class Cal {
	public int add(int a, int b) {
		return a+b;
	}
}

public class CalTest {

	private Cal cal;
	@Before
	public void setUp() throws Exception {
		cal=new Cal();
	}

	@Test
	public void testAdd() {
		int i=0;

		assertEquals(4, cal.add(2, 2));
		System.out.println("hello world cal running");
```

```
	}

	@After
	public void tearDown() throws Exception {
		cal=null;
	}

}
```

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
	xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
	<modelVersion>4.0.0</modelVersion>
	<groupId>com.demo</groupId>
	<artifactId>cal</artifactId>
	<version>0.0.1-SNAPSHOT</version>
	<properties>
		<maven.compiler.source>1.8</maven.compiler.source>
		<maven.compiler.target>1.8</maven.compiler.target>
	</properties>
	<dependencies>
		<!-- https://mvnrepository.com/artifact/junit/junit -->
		<dependency>
			<groupId>junit</groupId>
			<artifactId>junit</artifactId>
			<version>4.11</version>
			<scope>test</scope>
		</dependency>
		<!--
https://mvnrepository.com/artifact/org.sonarsource.scanner.maven/sonar-
maven-plugin -->
		<dependency>
			<groupId>org.sonarsource.scanner.maven</groupId>
			<artifactId>sonar-maven-plugin</artifactId>
			<version>3.6.0.1398</version>
		</dependency>


	</dependencies>

	<build>
		<plugins>

			<plugin>
				<groupId>org.jacoco</groupId>
				<artifactId>jacoco-maven-plugin</artifactId>
				<version>0.8.2</version>
				<executions>
					<execution>
						<id>prepare-agent</id>
						<goals>
							<goal>prepare-agent</goal>
						</goals>
						<phase>test-compile</phase>
					</execution>
```

```
                        <execution>
                              <id>report</id>
                              <goals>
                                    <goal>prepare-package</goal>
                              </goals>
                              <phase>report</phase>
                        </execution>

                        <execution>
                              <id>post-unit-test</id>
                              <phase>test</phase>
                              <goals>
                                    <goal>report</goal>
                              </goals>
                              <configuration>

      <destFile>target/jacoco.exec</destFile>
                                    <outputDirectory>target/jacoco-
ut</outputDirectory>
                              </configuration>
                        </execution>
                  </executions>

            </plugin>
      </plugins>
</build>

</project>
```

```
run clean test sonar:sonar
```

jacoco

JaCoCo is an open source toolkit for measuring code coverage in a code base and reporting it through visual reports