Exp No :14	Integration testing and quality gates

Date   :

Aim:

To verify the integration of application components using JUnit and enforce code quality standards using SonarQube.

Procedure:

1. Environment Setup:

1. Install SonarQube locally and set it up for code quality analysis.

2. Ensure Git is installed and properly configured for version control.

3. Install JUnit for writing and executing test cases.

2. Create the Application:

1. Write a simple Java program:

   o Create a HelloWorldProject containing an App class with a main method.

   o The method should print "Hello World!" to the console.

2. Write JUnit test cases to validate the functionality of the App class.

3. Integration Testing:

1. Verify the integration of components by:

   o Running JUnit test cases locally.

   o Ensuring all tests pass successfully.

4. Code Quality Check with SonarQube:

1. Use SonarQube to analyze the codebase for issues such as bugs, vulnerabilities, and code smells.

2. Define quality thresholds (e.g., no critical bugs, minimum 80% test coverage).

3. Ensure the project meets the defined quality standards.

Modify the Pom.XML :

```xml
<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <sonar.projectKey>myproject</sonar.projectKey>
  <sonar.host.url>http://localhost:9000</sonar.host.url>
  <sonar.login>sqa_c793f4432e1a4df9f1798d191534d50c10838d7c</sonar.login>
</properties>
<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>3.8.1</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.13.2</version>
    <scope>test</scope>
  </dependency>
</dependencies>
<build>
<plugins>
  <plugin>
    <groupId>org.sonarsource.scanner.maven</groupId>
    <artifactId>sonar-maven-plugin</artifactId>
    <version>3.9.1.2184</version>
  </plugin>
</plugins>
</plugins>
```

Program :

m pom.xml (HelloWorldProject)　　AppTest.java ×　　App.java ×

```java
1    package com.Ost;
2
3
4    public class App    new *
5    {
6        public static void main( String[] args )    new *
7        {
8            System.out.println( "Hello World!" );
9        }
10   }
11
```

Unit test class

Testing Program :

```
m pom.xml (HelloWorldProject)    ⓒ AppTest.java ×    ⓒ App.java

1    package com.0st;// src/test/java/HelloWorldTest.java
2    import org.junit.Test;
3
4    import static junit.framework.Assert.assertEquals;
5    import static org.junit.Assert.assertEquals;
6
7    public class AppTest {  new *
8        @Test  new *
9        public void testHelloWorldOutput() {
10           assertEquals( expected: "Hello, World!",  actual: "Hello, World!");
11       }
12   }
13
```

Preparation	
Observation	
Output	
Viva	
Record	
Total	

Result:

Thus the verification of the integration of application components using JUnit and enforce code quality standards using SonarQube has been completed.

Output:

```
[INFO] ---------------------< com.Ost:HelloWorldProject >---------------------
[INFO] Building HelloWorldProject 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ HelloWorldProject ---
[INFO] Deleting C:\Users\Rajkumar\Desktop\Projects\HelloWorldProject\target
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ HelloWorldProject ---
[INFO] skip non existing resourceDirectory C:\Users\Rajkumar\Desktop\Projects\HelloWorldProject\src\main\resources
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ HelloWorldProject ---
[INFO] Recompiling the module because of changed source code.
[INFO] Compiling 1 source file with javac [debug target 1.8] to target\classes
[WARNING] bootstrap class path not set in conjunction with -source 8
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ HelloWorldProject ---
[INFO] skip non existing resourceDirectory C:\Users\Rajkumar\Desktop\Projects\HelloWorldProject\src\test\resources
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ HelloWorldProject ---
[INFO] Recompiling the module because of changed dependency.
[INFO] Compiling 1 source file with javac [debug target 1.8] to target\test-classes
[WARNING] bootstrap class path not set in conjunction with -source 8
[INFO] /C:/Users/Rajkumar/Desktop/Projects/HelloWorldProject/src/test/java/com/Ost/AppTest.java: C:\Users\Rajkumar\Desktop\Projects\HelloWorldProject\src\test\java\com\Ost\AppTest.
java uses or overrides a deprecated API.
[INFO] /C:/Users/Rajkumar/Desktop/Projects/HelloWorldProject/src/test/java/com/Ost/AppTest.java: Recompile with -Xlint:deprecation for details.
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running com.Ost.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.080 s -- in com.Ost.AppTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jar:3.4.1:jar (default-jar) @ HelloWorldProject ---
[INFO] Building jar: C:\Users\Rajkumar\Desktop\Projects\HelloWorldProject\target\HelloWorldProject-1.0-SNAPSHOT.jar
[INFO]
[INFO] ---------------------< com.Ost:HelloWorldProject >---------------------
[INFO] Building HelloWorldProject 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- sonar:3.9.1.2184:sonar (default-cli) @ HelloWorldProject ---
[INFO] User cache: C:\Users\Rajkumar\.sonar\cache
[INFO] SonarQube version: 10.7.0.96327
[INFO] Default locale: "en_IN", source code encoding: "UTF-8"
[INFO] Load global settings
[INFO] Load global settings (done) | time=263ms
[INFO] Analysis report generated in 124ms, dir size=227.1 kB
[INFO] Analysis report compressed in 37ms, zip size=25.9 kB
[INFO] Analysis report uploaded in 59ms
[INFO] ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=myproject
[INFO] Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
[INFO] More about the report processing at http://localhost:9000/api/ce/task?id=63521cdd-e349-4a61-815c-ad10cf5a6eee
[INFO] Analysis total time: 21.719 s
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  28.369 s
[INFO] Finished at: 2024-11-23T09:16:42+05:30
[INFO] Finished at: 2024-11-23T09:16:42+05:30
[INFO] ------------------------------------------------------------------------
```

SonarQube Output: