

### Lab Exercise – SonarQube

IT3040 – IT Project Management

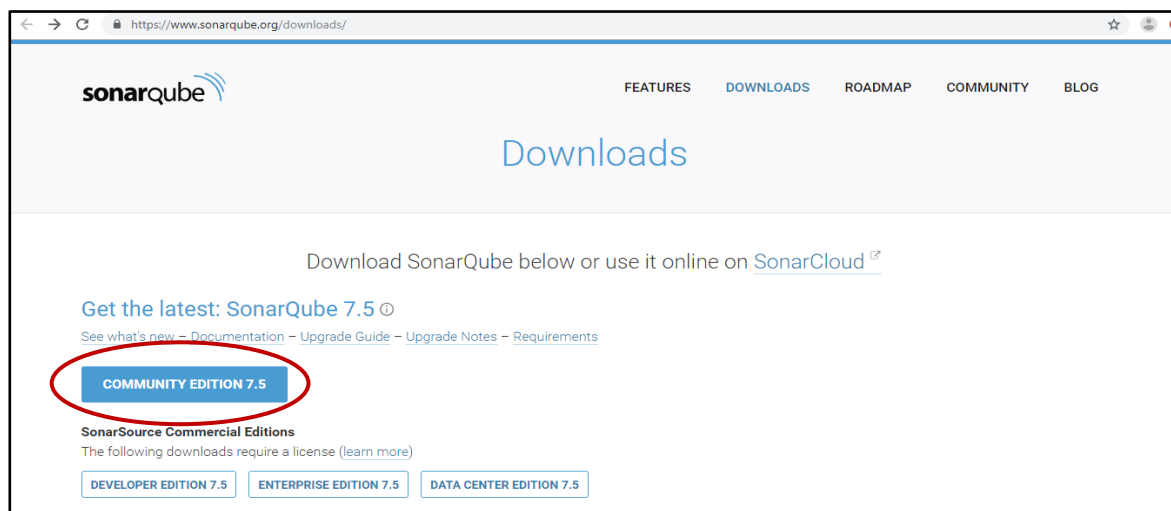
Semester 1

SonarQube is an open-source platform used for continuous inspection of code quality to perform automatic reviews with static analysis of code to detect bugs, code smells, and security vulnerabilities.

#### Installation:

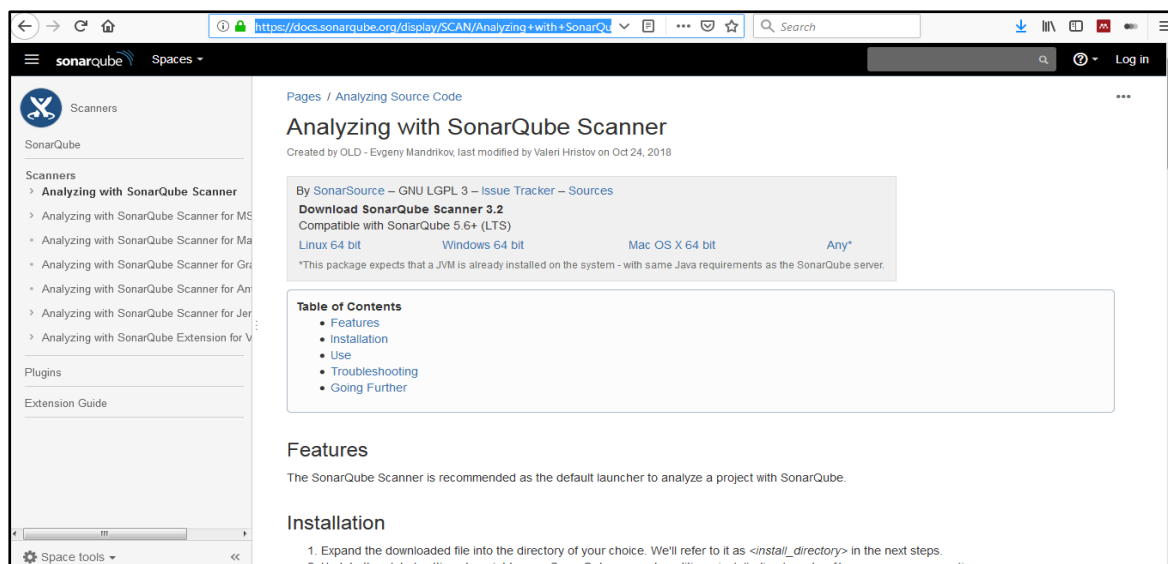
1. Go to SonarQube website and download the SonarQube server. Extract the downloaded package on C: drive.

URL: <https://www.sonarqube.org/downloads/>



2. Download and extract SonarQube scanner.

URL: <https://docs.sonarqube.org/display/SCAN/Analyzing+with+SonarQube+Scanner>



**Lab Exercise – SonarQube****IT3040 – IT Project Management****Semester 1****Adding a project to sonar-scanner properties**

1. Open sonar-scanner.properties file inside the conf folder.

sonar-scanner-cli-3.2.0.1227-windows → sonar-scanner-3.2.0.1227-windows → conf

Default content in sonar-scanner.properties file are as follows;

Default URL to access the reports.

```
#----- Default SonarQube server
#sonar.host.url=http://localhost:9000
```

Default source code encoding

```
#----- Default source code encoding
#sonar.sourceEncoding=UTF-8
```

2. Add the following lines in sonar-scanner.properties file to add a pure entry of ITPM\_SonarqubeSampleProject.

```
sonar.projectKey=ITPM_SonarqubeSampleProject
sonar.projectName=ITPM_SonarqubeSampleProject
sonar.projectVersion=1.0
sonar.sources=C:/ITPM_SonarqubeSampleProject/src/itpm_sonarqubesampleproject
```

**Note:** SonarQube report will be generated based on the content added in sonar-scanner.properties file.

- **Project Key** is a unique identification in sonar-scanner for the project.
- **Sources** is the path to the source codes that needs to be analyzed.

**Run sonar-scanner and generate the report for ITPM\_SonarqubeSampleProject**

1. Start SonarQube server.
  - i. Go to: C: → sonarqube-7.5 → sonarqube-7.5 → bin
  - ii. Go inside the relevant folder according to the operating system of the device and copy the folder path.
  - iii. Open command prompt and change the directory by execute the following command.

**cd C:\sonarqube-7.5\sonarqube-7.5\bin\windows-x86-64**

**Note:** C:\sonarqube-7.5\sonarqube-7.5\bin\windows-x86-64 is the folder path copied in above step (iii).

### Lab Exercise – SonarQube

IT3040 – IT Project Management

Semester 1

- iv. Start SonarQube server by executing the **startSonar** command.

```
C:\sonarqube-7.5\sonarqube-7.5\bin\windows-x86-64>startSonar
```

Once the server is up, following message will be displayed in the command prompt.

```
jvm 1      | 2019.01.07 22:07:56 INFO  app[[io.s.a.SchedulerImpl]] Process[cel] is u
jvm 1      | 2019.01.07 22:07:56 INFO  app[[io.s.a.SchedulerImpl]] SonarQube is up
```

**Note:** SonarQube server should be up to generate the report. In order to check whether the server is up, go to the URL <http://localhost:9000> from your browser and check whether it is working.

2. Open another command prompt window and change the directory using following command.

```
cd C:\Users\User\Desktop\ITPM_SonarqubeSampleProject
```

**Note:** *C:\Users\User\Desktop\ITPM\_SonarqubeSampleProject* is the folder path to *ITPM\_SonarqubeSampleProject* project folder.

3. Set the environmental variable and add the path to sonar-scanner.

Execute the following command in the command prompt:

```
set path=%PATH%; C:\sonar-scanner-cli-3.2.0.1227-windows\sonar-scanner-3.2.0.1227-windows\bin
```

**Note:** *C:\sonar-scanner-cli-3.2.0.1227-windows\sonar-scanner-3.2.0.1227-windows\bin* in above command is the path to bin folder inside sonar-scanner. You can also add sonar-scanner bin at to the environment variable by changing system properties.

4. Execute **sonar-scanner** command.

```
C:\Users\User\Desktop\ITPM_SonarqubeSampleProject>sonar-scanner
```

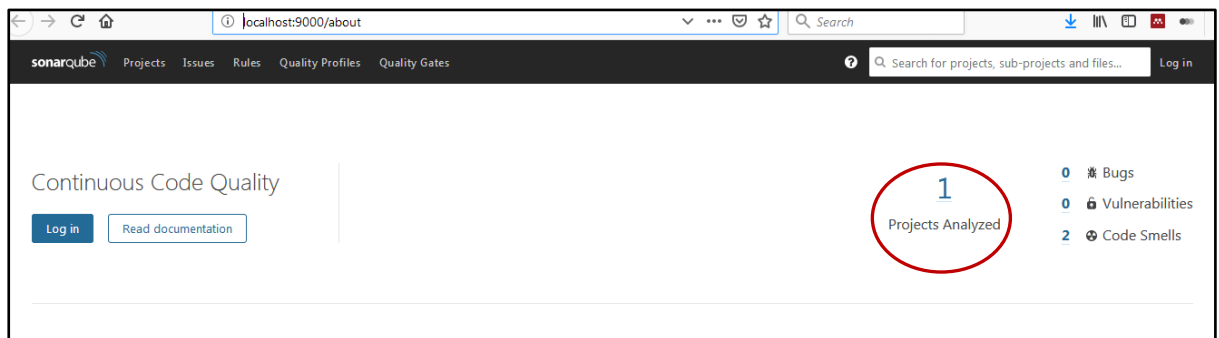
**Note:** Once the sonar-scanner is successfully completed following message will be displayed in the command prompt.

```
INFO: -----
INFO: EXECUTION SUCCESS
INFO: -----
INFO: Total time: 23.808s
INFO: Final Memory: 15M/191M
INFO: -----
```

**Lab Exercise – SonarQube****IT3040 – IT Project Management****Semester 1**

*Once the sonar-scanner is successfully executed, report will be sent to the SonarQube server.*

*Go to URL: <http://localhost:9000> . Projects Analyzed count of this page is changed.*



*Click the Projects Analyzed to view the report of projects analyzed.*

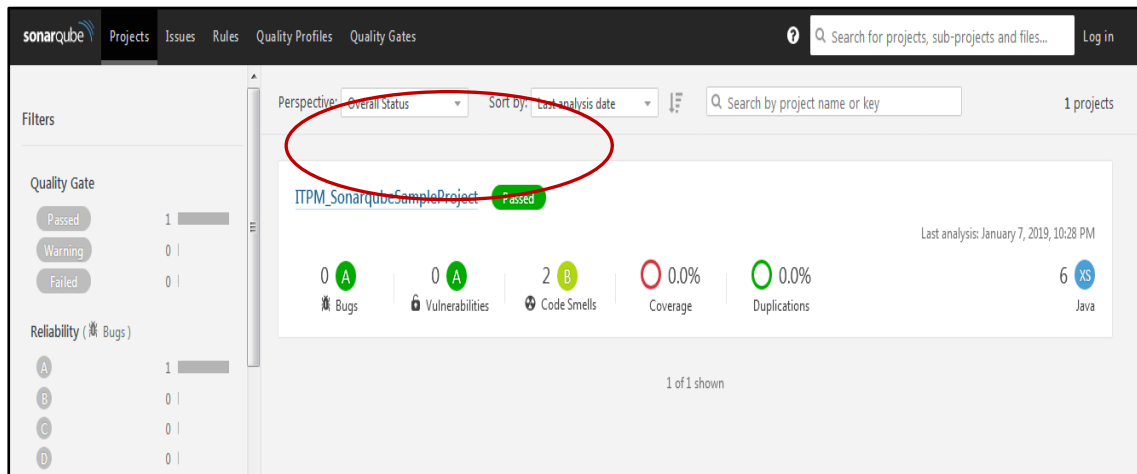
### Lab Exercise – SonarQube

IT3040 – IT Project Management

Semester 1

#### SonarQube Report

1. Select the project to view the report.



In SonarQube report, analyzers contribute rules which are executed on source code to generate issues. There are four types of rules:

- **Code Smell (Maintainability domain)**  
A maintainability-related issue in the code. Leaving it as-is means that at best maintainers will have a harder time than they should making changes to the code. At worst, they'll be so confused by the state of the code that they'll introduce additional errors as they make changes.
- **Bug (Reliability domain)**  
An issue that represents something wrong in the code. If this has not broken yet, it will, and probably at the worst possible moment. This needs to be fixed.
- **Vulnerability (Security domain)**  
A security-related issue which represents a backdoor for attackers. A special type of issue that identify sensitive areas of code that should be reviewed by a Security Auditor to determine if they are truly Vulnerabilities.
- **Security Hotspot (Security domain)**  
A security-related issue highlighting a piece of code that uses a security-sensitive API (E.G. use of a weak algorithm, connection to a database without a password, ...). Security hotspots must be reviewed by a security auditor who may determine that the APIs are used in ways that introduce Vulnerabilities.

### Lab Exercise – SonarQube

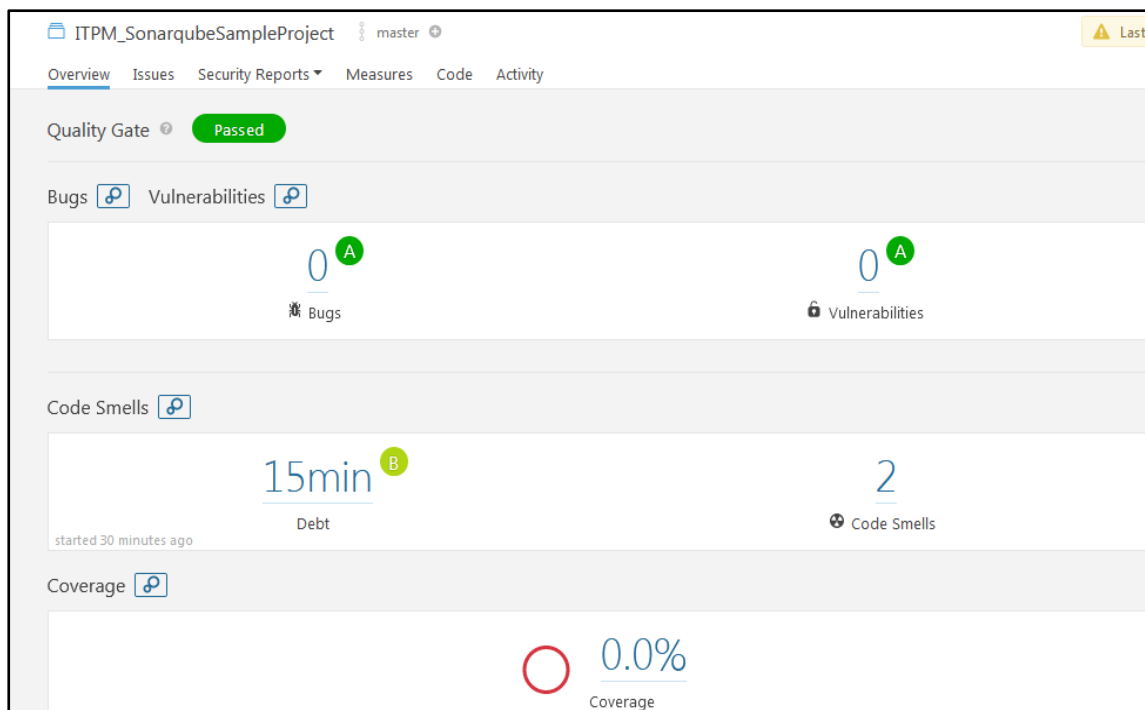
#### IT3040 – IT Project Management

#### Semester 1

For Code Smells and Bugs, zero false-positives are expected. At least this is the target so that developers don't have to wonder if a fix is required.

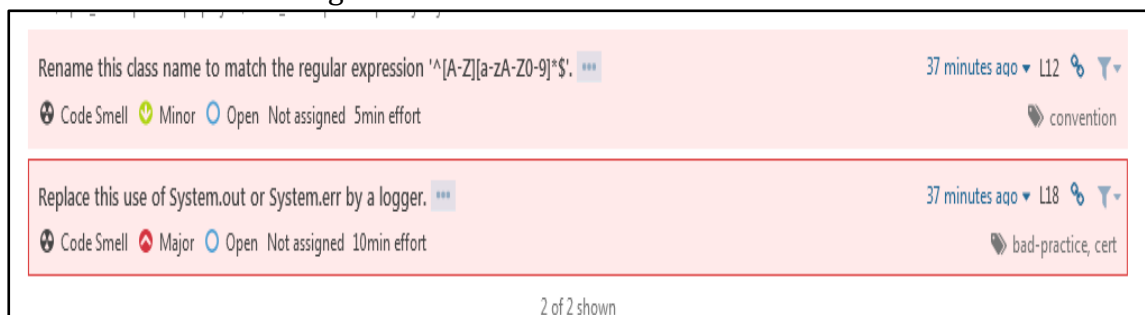
For Vulnerabilities, the target is to have more than 80% of the issues to be true-positives.

Security Hotspot rules are purposefully designed to draw attention to code is security-sensitive. It is expected that more than 80% of the issues will be quickly resolved as "Won't Fix" after review by a Security Auditor.



**Note:** Debt in the output indicates the time to fix the issues in code.

- Click on Code Smells to get information about the errors.



This block shows a detailed view of two Code Smells:

- Smell 1:** 'Rename this class name to match the regular expression `^[A-Z][a-zA-Z0-9]*$`.' It is a Minor issue, Open, Not assigned, with 5min effort. It was detected 37 minutes ago (L12) and is associated with the 'convention' rule.
- Smell 2:** 'Replace this use of System.out or System.err by a logger.' It is a Major issue, Open, Not assigned, with 10min effort. It was detected 37 minutes ago (L18) and is associated with the 'bad-practice, cert' rule.

At the bottom, it indicates '2 of 2 shown'.

### Lab Exercise – SonarQube

#### IT3040 – IT Project Management

#### Semester 1

In the SonarQube report, each issue has a severity type. The five severity types of issues in SonarQube are as follows:

- **Blocker**

Bug with a high probability to impact the behavior of the application in production: memory leak, unclosed JDBC connection, .... The code **MUST** be immediately fixed.

- **Critical**

Either a bug with a low probability to impact the behavior of the application in production or an issue which represents a security flaw: empty catch block, SQL injection, ... The code **MUST** be immediately reviewed.

- **Major**

Quality flaw which can highly impact the developer productivity: uncovered piece of code, duplicated blocks, unused parameters, etc.


- **Minor**

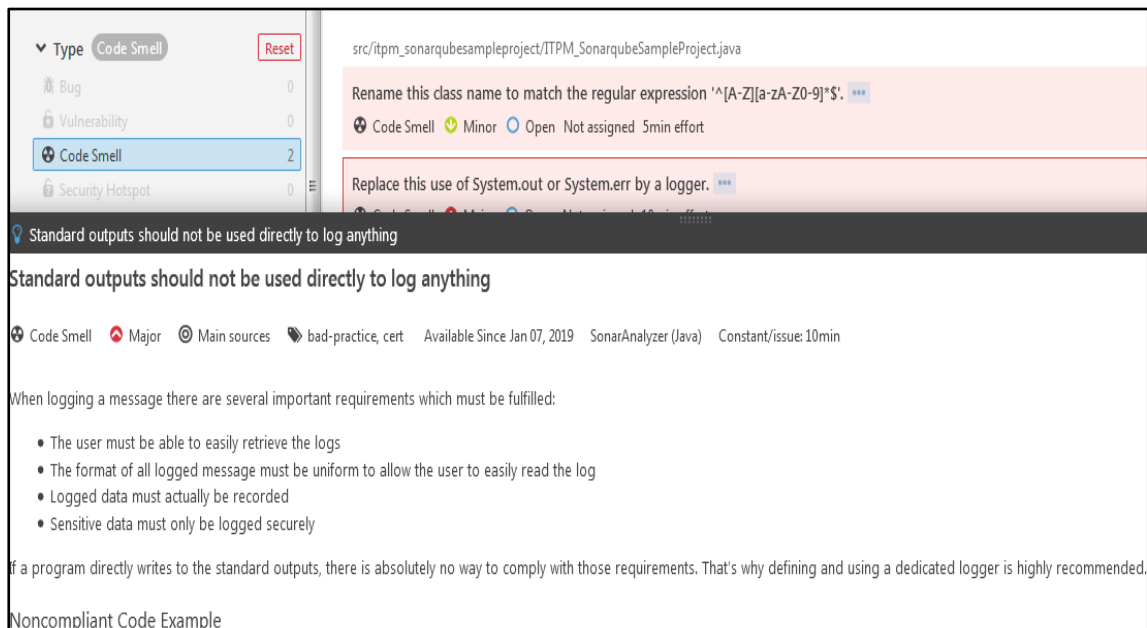
Quality flaw which can slightly impact the developer productivity: lines should not be too long, "switch" statements should have at least 3 cases, etc.

- **Info**

Neither a bug nor a quality flaw, just a finding.

#### 3. Fix the major error listed and generate the report again.

**Note:** Click the  to get more details on how to fix the error.



The screenshot shows the SonarQube web interface. On the left, a sidebar lists issue types: Bug (0), Vulnerability (0), Code Smell (2), and Security Hotspot (0). The 'Code Smell' item is selected. The main area displays a specific issue titled 'Standard outputs should not be used directly to log anything'. The issue is categorized as a 'Code Smell' with a 'Minor' severity. It includes a description: 'When logging a message there are several important requirements which must be fulfilled:' followed by a bulleted list of requirements. The requirements are: 'The user must be able to easily retrieve the logs', 'The format of all logged message must be uniform to allow the user to easily read the log', 'Logged data must actually be recorded', and 'Sensitive data must only be logged securely'. Below the list, it states: 'If a program directly writes to the standard outputs, there is absolutely no way to comply with those requirements. That's why defining and using a dedicated logger is highly recommended.' The issue also shows a 'Noncompliant Code Example' section.

**Lab Exercise – SonarQube****IT3040 – IT Project Management****Semester 1**

---

**References:**

1. <https://docs.sonarqube.org/latest/>
2. <https://docs.sonarqube.org/display/SCAN/Analyzing+with+SonarQube+Scanner>
3. <https://www.youtube.com/watch?v=Gu9skLzRSao>
4. [https://www.youtube.com/watch?v=puP59\\_PoeUc](https://www.youtube.com/watch?v=puP59_PoeUc)