



FACULTY  
OF INFORMATICS

Masaryk University

# SonarQube

---

**Seminar group Java-Lasaris**

Xuan Linh Phamová

# Introduction

What is SonarQube?

## SonarQube

SonarQube<sup>1</sup> is an open-source automatic code review tool to detect bugs, vulnerabilities, and code smells in some code.

## SonarQube supports:

- integration with **Maven, Ant, Gradle, MSBuild**
- integration with **continuous integration tools** (*Atlassian Bamboo, Jenkins, Hudson, etc.*)
- **27** programming languages (*Java, C#, C/C++, JavaScript, TypeScript, Python, etc.*)

---

<sup>1</sup>Website: <https://www.sonarqube.org/>

# Overview

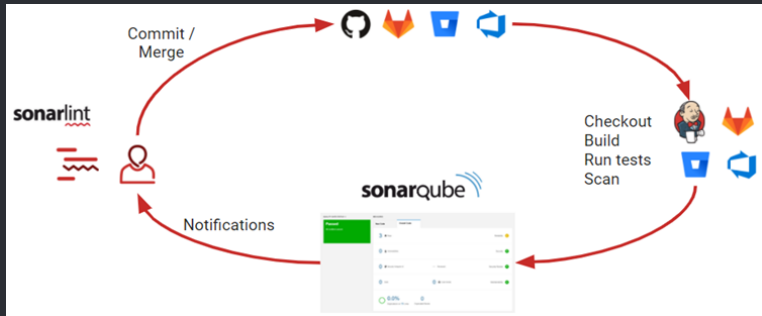


Figure: Development cycle with SonarQube

# User Interface

## Main page

The screenshot displays the SonarQube main page. At the top, there is a navigation bar with the SonarQube logo and menu items: Projects, Issues, Rules, Quality Profiles, Quality Gates, and Administration. A search bar on the right contains the text 'Search for projects...' and a green 'A' icon.

Below the navigation bar, the main content area is divided into two columns. The left column contains filters for 'My Favorites' (set to 'All') and 'Filters'. Under 'Quality Gate', there are two buttons: 'Passed' (with a progress bar at 1) and 'Failed' (with a progress bar at 0). Under 'Reliability ( Bugs )', there are five radio buttons labeled A, B, C, D, and E, each with a corresponding progress bar. Under 'Security ( Vulnerabilities )', there are five radio buttons labeled A, B, C, D, and E, each with a corresponding progress bar. Under 'Security Review ( Security Hotspots )', there are five radio buttons labeled A, B, C, D, and E, each with a corresponding progress bar.

The right column features a search bar 'Search by project name or key' and a blue 'Add project' button with a star icon. Below this, it shows '1 projects' and a 'Perspective' dropdown set to 'Overall Status'. A 'Sort by' dropdown is set to 'Name'. The main project card is titled 'Example of basic Maven project' with a green 'Passed' status and 'Last analysis: 25 minutes ago'. The card displays several metrics: 'Bugs' (0 A), 'Vulnerabilities' (0 A), 'Hotspots Reviewed' (1 A), 'Code Smells' (3 A), 'Coverage' (60.0% with a red progress bar), 'Duplications' (0.0% with a green progress bar), and 'Lines' (89 XS XML, Java). At the bottom of the card, it says '1 of 1 shown'.

At the bottom of the page, there is a footer with the text: 'SonarQube™ technology is powered by SonarSource SA. Community Edition - Version 8.7 (build 41497) - LGPL v3 - Community - Documentation - Plugins - Web API - About'.

Figure: UI of the main page

# User Interface

## Project analysis

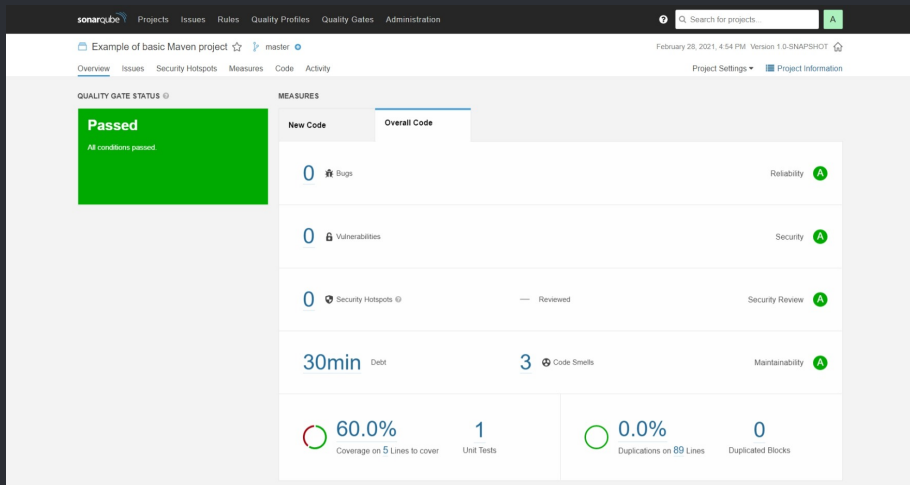


Figure: UI of the page containing project analysis results

# User Interface

## *Project analysis*

1. **Issues:** SonarQube raises issues whenever a piece of the code breaks a coding rule – whether it's a
  - **bug:** an error that will break the code
  - **vulnerability:** a point in the code open to attack
  - **code smell:** a maintainability issue
2. **Security Hotspots** SonarQube highlights security-sensitive pieces of code that need to be reviewed
3. **Measures:** a dedicated sub-space where all project measures can be seen
4. **Code:** an outline of the project structure
5. **Activity:** contains a full list of code scans performed on the project since it was created in SonarQube

# How to create an instance of SonarQube?

You can create a SonarQube instance either from the:

- zip file (*simplified version is covered in these slides*)
- docker image

# How to create a local instance of SonarQube?

## *Setting up the environment*

### Download the recommended platforms:

1. Java 11 – download either Oracle JDK 11 or OpenJDK – versions beyond Java 11 are not officially supported!
  - if you have a newer version of JDK installed, download the .zip file instead
2. PostgreSQL 12 – download [here](#)

When you're done with setting up the environment, download the **SonarQube zip file** (*SonarQube Community Edition*).

We will refer to the directories into which you unzipped JDK and SonarQube as **\$SONARQUBE-HOME** and **\$JAVA-HOME**, respectively.



# How to create a local instance of SonarQube?

## *Installing the database*

1. Firstly, check the requirements [here](#) (*hardware requirements etc.*)
2. Install the server
  - 2.1 install PostgreSQL database – open `psql` and enter these commands:

```
CREATE USER [user] WITH PASSWORD '[password]';  
CREATE DATABASE [database] OWNER [user];
```

then enter `\connect [database]`

```
GRANT CREATE ON SCHEMA "public" TO [user];  
GRANT UPDATE ON ALL TABLES IN SCHEMA "public" TO [user];  
GRANT DELETE ON ALL TABLES IN SCHEMA "public" TO [user];
```

# How to create a local instance of SonarQube?

## *Installing the server*

### 2. cont.

#### 2.2 configure database settings

2.2.1 open `$SONARQUBE-HOME/conf/sonar.properties`

2.2.2 edit properties to:

```
sonar.jdbc.username=[user]
```

```
sonar.jdbc.password=[password]
```

```
sonar.jdbc.url=jdbc:postgresql://localhost/sonarqube
```

where `[user]` and `[password]` are the values you entered in the previous step 1

#### 2.3 adjust the Java installation

- » if you didn't have the JDK installed prior to this seminar: add the JDK folder into PATH
- » otherwise open `$SONARQUBE-HOME/conf/wrapper.conf` and edit the property `wrapper.java.command` – it must be the path to the `Java executable`

# How to create a local instance of SonarQube?

## *Starting the web server*

### 3. Start the web server

3.1 open a shell and based on your OS, execute:

» Linux: `$SONARQUBE-HOME/bin/linux-x86-64/sonar.sh`  
`start`

» macOS:

`$SONARQUBE-HOME/bin/macosx-universal-64/sonar.sh`  
`start`

» Windows:

`$SONARQUBE-HOME/bin/windows-x86-64/StartSonar.bat`

3.2 to check whether it is running correctly, the shell outputs  
`SonarQube is up` in case it is set up correctly

3.3 open `http://localhost:9000` in your browser

3.4 login with credentials `admin/admin`

If anything fails, please check the [tutorial](#) and follow all the steps carefully.

# How to integrate GitLab with SonarQube?

## *Creating GitLab OAuth application*

The full tutorial can be found [here](#).

1. open **GitLab**, log in, click on your profile photo in the upper-right corner → **Edit profile** → **Applications** → fill in **Name** (for example, *SonarQube*) → into **Redirect URI** fill `http://localhost:9000/oauth2/callback/gitlab` in → in **Scopes**, check **api** → click on **Save application**
2. GitLab redirects you to a page → copy **Application ID** and **Secret**

# How to integrate GitLab with SonarQube?

## *Changing global settings in SonarQube*

1. go to the website of your local SonarQube instance → **Administration** → **Configuration** → **ALM Integrations** → go to the **GitLab** tab → section **GitLab Authentication** → toggle **Enabled** to **true** → fill in **GitLab URL** (*in case you use your school account, fill `https://gitlab.fi.muni.cz` in*) → into **Application ID** and **Secret** paste the values obtained in the previous step → toggle **Synchronize user groups** to **true** → save the settings

# How to integrate GitLab with SonarQube?

## *Changing global settings in SonarQube*

2. go back to your GitLab profile → **Edit profile** → **Access Tokens** → fill in **Name** → in **Scopes**, check **api** → click on **Create personal access token** → copy the token (*be sure to keep it in a safe place, so you can return to the token later*)
3. go back to your SonarQube instance → above the section **GitLab Authentication**, click on **Create configuration** → fill in **Configuration name** → into **GitLab API URL** fill the API URL in (*in case you use `gitlab.fi.muni.cz`, fill `https://gitlab.fi.muni.cz/api/v4/` in*) → into **Personal Access token** paste the token generated in the previous step → **Save configuration** → check that the configuration is valid