

P1 - Technical debt calculation/evaluation

Due date: 15/10/2023 at 23.55

The goal of this deliverable is twofold:

1. Help you to familiarize with the project
2. Use a state-of-the-art technical debt analysis tool and apply it on the source code.

Computing technical Debt with SonarQube – Community Edition

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

Important: Do not spend more than a **maximum of 6 hours**.

Helpful link:

<https://duecode.io/blog/what-is-technical-debt-in-sonarqube/>

Deliverable:

1. Cover page with Group # and list of the group members.
2. **A table, indicating the contribution of each group member.**
Name, ID, Github login, Contribution (towards the group and individual part)
3. Analyze our fork (depending on which project you are assigned to)

[rilling/OpenTracksConcordia](#) or [OSMDashboardConcordia](#)

with SonarQube . – can be done as a group

SonarQube will analyze the whole project. For the detailed analysis, focus on the code which was assigned to your group (See document on Moodle).

4. Report the technical debt for the overall code of your group.
 - Screen captures of technical debt reported for your components (below are some examples). If no technical debt analysis can be performed on the code assigned to your group or there is no technical debt reported for your assigned code, report on the overall analysis.
 - Discuss briefly hotspots (area with the most technical debt) in your assigned code base and also mention the top three types of technical debt found in your analyzed code.

5. **Each group member** should then take on **at least one instance** of a technical debt reported by SonarQube. **Each group member should do the following**
- Create an issue on the Issue tracker in GitHub – Use the “Technical Debt” template and make sure you follow the standards for creating good issues. Also select the milestone (only one available) and labels (group # and “technical debt”), assign the issue to a group member.
 - Update the issue status to close once you completed the issue and include a copy of the code changes after removing the technical debt.
 - Commit the changed code – make sure to provide a complete commit message with a reference to the issue.

As part of the individual documentation

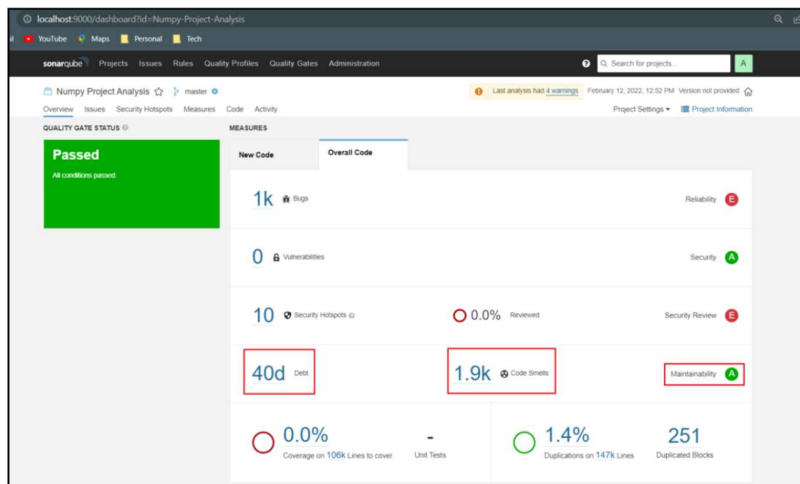
- Include a screen capture of the issue you worked on + a link to the issue
- include a screen capture of the committed code + link to the commit

6. **Note:** Combine all group and individual parts in **one PDF document**. **Important:** only **one submission** per group.

In case **you were unable to run SonarQube** or ran into a problem with analyzing the software – mention where you got stuck.

Example views – these views might differ from the version of SonarQube you install and what you are able to generate.

1. Overview



4. Technical Debt Graph



sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration

Numpy Project Analysis

Overview Issues Security Hotspots Measures Code Activity

Search for files

Lines of Code Bugs Vulnerabilities Code Smells Security Hotspots Coverage Duplications

1	__init__	56	3	0	0	0	0.0%	0.0%
1	array_api	3,479	0	0	20	0	0.0%	0.0%
1	compat	491	0	0	1	0	0.0%	0.0%
1	core	14,137	80	0	614	0	0.0%	0.4%
1	distutils	14,044	1	0	221	1	0.0%	1.2%
1	doc	393	0	0	1	0	0.0%	0.0%
1	Obj	12,486	16	0	219	0	0.0%	1.1%
1	os	624	0	0	6	0	0.0%	0.0%
1	os	22,196	8	0	250	3	0.0%	1.5%
1	testing	3,117	0	0	82	0	0.0%	0.0%
1	ma	11,896	26	0	215	0	0.0%	1.7%
1	matrix	1,144	10	0	24	0	0.0%	0.0%
1	polynomial	5,779	4	0	80	0	0.0%	3.1%
1	random	7,699	0	0	36	0	0.0%	11.9%
1	testing	3,272	2	0	85	0	0.0%	1.8%
1	utils	842	0	0	10	0	0.0%	0.0%
1	typing	5,275	895	0	7	0	0.0%	0.0%
1	_util.py	202	0	0	0	0	0.0%	0.0%