A 3. QUALITY ASSURANCE

Static Analysis

Practical Exercise

Understanding and ensuring software quality are fundamental aspects of software engineering. This includes the practice of static analysis, which examines code without executing it to identify potential flaws early in the development cycle. Embracing static analysis helps improve code safety, maintainability, and compliance with standards, which are crucial for delivering reliable and efficient software products in any industry.

In this exercise, you will gain hands-on experience with *Sonarqube*, a leading tool for enhancing code quality and security. Your task is to install *Sonarqube* on your local machine and use it to analyze a software project. This practical assignment is designed to deepen your understanding of static code analysis and its impact on software development processes. This is an essential skill for any software engineer, especially as you approach professional environments.

Practical Exercise

1) Install Sonarqube in a local server. Please follow the instructions provided on the Sonarqube documentation page.

2) Conduct a static code análisis of the Project you've developed.

3) Compile a report detailing the findings of your analysis.

4) Your analysis will be especially valued if it includes a comparison between the default settings in the Sonarway profile and its quality gates, and results obtained from a new profile and quality gate you create by modifying some rules.

NOTE: While Sonar Cloud may be used, installing and utilizing a local server for this exercise is highly encouraged and will be regarded favorably.

Practical Exercise

Documentation:

Link: Sonarqube / Link: Installation Guide

Sonarqube is a powerful static code analysis tool that helps developers detect bugs, vulnerabilities, and code smells in their software. It supports multiple programming languages and integrates seamlessly into development workflows, enhancing code quality and security..

Link: Quality Profiles

Quality Profiles in *Sonarqube* are essential for configuring and enforcing coding rules and standards. They allow teams to tailor the analysis to specific project needs, ensuring that code adheres to agreed-upon quality and security benchmarks.

Link: Quality Gates

Quality Gates in *Sonarqube* serve as checkpoints that assess whether a software project meets specific quality criteria before it can be released. This feature ensures that only code that passes these rigorous standards progresses through the development pipeline.