**Software Requirements Specification (SRS)**

# **1. Introduction**

## **1.1 Purpose**

The purpose of this document is to define the Software Requirements Specification (SRS) for the "Software Quality Metrics in an Agile Context" project. This project aims to implement software quality metrics as encapsulated services and develop a UI-based application to utilize these services effectively within an Agile/Lean process.

## **1.2 Scope**

The system will allow users to:

* Select and configure projects for which to apply software quality metrics.
* Select and configure metrics for evaluation.
* Gather necessary data and compute metrics.
* Visualize the metrics through charts and graphs.
* Benchmark metrics over time against a predefined baseline.
* Follow Scrum methodologies for software development.

## **1.3 Definitions, Acronyms, and Abbreviations**

* **Agile/Lean Process:** An iterative approach to software development.
* **Encapsulated Service:** A self-contained unit providing specific functionality.
* **UI:** User Interface.
* **CI/CD:** Continuous Integration and Continuous Deployment.
* **Scrum:** Agile development framework.

## **1.4 References**

* Agile Manifesto
* IEEE Standard for Software Requirements Specifications

# **2. Overall Description**

## **2.1 Product Perspective**

This system will be a standalone application that interacts with encapsulated services through APIs. It will follow a modular architecture to ensure loose coupling and scalability.

## **2.2 Product Functions**

The key functionalities of the system include:

1. **Project Configuration**: Users can select and configure projects.
2. **Metric Selection**: Users can choose which quality metrics to apply.
3. **Data Collection**: The system gathers necessary data for metric calculations.
4. **Metric Computation**: The system computes raw metric values.
5. **Visualization**: The system displays computed metrics in a user-friendly manner.
6. **Benchmarking**: The system allows comparison of metrics over time.

## **2.3 User Characteristics**

The primary users will be software developers, project managers, and quality analysts familiar with Agile methodologies and software quality metrics.

# **3. Specific Requirements**

## **3.1 Functional Requirements**

* The system shall allow users to select projects.
* The system shall provide an interface to select and configure metrics.
* The system shall gather data from repositories and external sources.
* The system shall compute and display metric values in tabular and graphical formats.
* The system shall support benchmarking of metrics over different time periods.

## **3.2 Non-Functional Requirements**

* **Scalability:** The system should support multiple projects and users.
* **Performance:** Metrics should be computed within an acceptable time frame.
* **Usability:** The UI should be intuitive and easy to navigate.
* **Security:** User data should be protected with authentication mechanisms.
* **Maintainability:** The system should follow best coding practices to ensure easy maintenance.

# **4. External Interface Requirements**

## **4.1 User Interfaces**

* A web-based dashboard to interact with the encapsulated services.

## **4.2 Hardware Interfaces**

* The system should be deployable on standard cloud or local servers.

## **4.3 Software Interfaces**

* REST APIs for encapsulated services.
* Integration with GitHub, Taiga, and Slack.

# **5. Other Requirements**

* The system should be buildable from GitHub without an IDE.
* The system should include unit tests and peer reviews.
* The development process should follow Scrum best practices.