

Quality Manual, Policy and Objectives

Table of Contents

Summary	1
1. Scope	1
Role of Project	1
Context Map	2
Relation to openEHR/java-lib.	2
2. Quality Policy	3
3. Objectives	3
Up-to-dateness	3
Follow best practices	3
Zero defects	3
4. Roles	3
Maintainer	3
Programmer	4
Technical Writer	4
Client	4
5. Processes	4
Glossary	5

Summary

The Quality Manual describes the scope of the Quality Management System (QMS), its documented procedures and a description of their interactions.

1. Scope

The QMS described in this Quality Manual applies to all software products of [openEHR 4J](#).

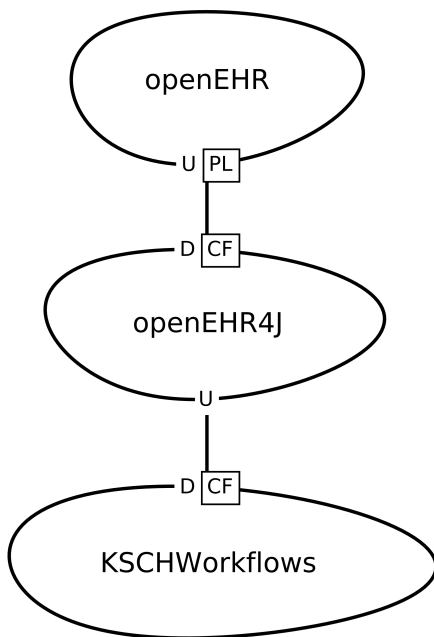
Role of Project

[openEHR4J](#) is an open source project based on volunteer work. It is a private initiative that is not affiliated nor endorsed by [openEHR International](#).

Context Map

The [openEHR specifications](#) are an abstract, programming language independent model for healthcare data. The openEHR4J project attempts to provide organizations that implement openEHR with the Java programming language, e.g. the [KSCH Workflows project](#), tools which verify the compliance the data models defined in the openEHR specification.

The following Context Map diagram provides an overview over the projects related with openEHR4J in the terminology of the Domain-Driven Design methodology^[1]:



CF

Conformist - a relationship between external projects in which the downstream project "slavishly" adheres to the upstream project

D

Downstream - a project that uses services from a upstream project

PL

Published Language - a data exchange format

U

Upstream - a project that provides services to a downstream project

Relation to openEHR/java-libs

The GitHub project [openEHR/java-libs](#) provides a reference implementation for openEHR in the Java programming language^[2]. In contrast to openEHR/java-libs, openEHR4J does not provide any components for production use but only helps with the verification of third-party components.

2. Quality Policy

The goal of this project is to facilitate the implementation of openEHR for healthcare-related information systems.

Being a non-commercial open source project, the terms of the [Apache License, Version 2.0](#) apply in terms of warranty and liability for the provided software products.

3. Objectives

Up-to-dateness

- Given an openEHR specification is supported, when a new version is released, then a new openEHR4j package is released within 30 days.

Follow best practices

- Open source best practice compliance will be measured with a badge from the [OpenSSF Best Practices Badge Program](#).
- The code style will be checked with the default configuration of [SonarCloud](#) and by using the [Google Java code style](#), validated with [Checkstyle](#).

Zero defects

- A search on [GitHub](#) with the query `org:openehr4j label:bug` should yield 0 results in the "Issues" tab^[3].

4. Roles

Maintainer

Responsibilities

- Triage feedback in the form of GitHub issues (add labels for type and priority, complete issue description, request further information, etc.)
- Accept Pull Requests for the source code or request changes
- Create new releases

Required qualification

- Basic knowledge about open source project governance^[4]
- Advanced knowledge of the Java programming language
- Basic knowledge of [Open Source Software Supply Chain Security](#), [Supply-chain Levels for](#)

- Advanced knowledge of the version control system Git

Programmer

Responsibilities

- Implement features and fix defects in the code generator
- Participate in discussions regarding the style of the generated code

Required qualification

- Advanced knowledge of the Java programming language
- Basic knowledge of the version control system Git

Technical Writer

Responsibilities

- Create concepts for documentation structure
- Write usage guides for project website
- Write README files for Git repositories

Required qualification

- Fluent in English language
- Basic knowledge of the Java programming language
- Basic knowledge of the version control system Git

Client

Responsibilities

- Provide feedback about defects and usability issues

Required qualification

- Clear communication
- Advanced knowledge of the Java programming language

5. Processes

The following Standard Operating Procedures (SOP) documents describe key development and maintenance activities:

- [SOP Deployment](#)
- [SOP Integrated Software Development](#)
- [SOP Feedback Management](#)
- [SOP Problem Resolution](#)

Glossary

SDK

Software Development Kit. A package developed by a third party that can be used within the source code of the source code of a software product.

SOP

Standard Operating Procedure. step-by-step instructions for routing operations (also see wikipedia.org and sweetprocess.com)

Legal notice

Template Copyright openregulatory.com. See [template license](#).

Please don't remove this notice even if you've modified contents of this template.

[1] see [Domain-Driven Design: Tackling Complexity in the Heart of Software](#), Eric Evans, 2003, chapter 14

[2] see [Chen, Rong & Klein, Gunnar. \(2007\). The openEHR Java reference implementation project. Studies in health technology and informatics. 129. 58-62. 10.3233/978-1-58603-774-1-58.](#)

[3] <https://github.com/search?q=org%3Aopenehr4j+label%3Abug&type=issues>

[4] <https://learning.oreilly.com/library/view/open-source-projects/9781837636884>