Document ID: PLAN-0020 ●**Title**: Core Library Development Plan

Revision: 1.0 ● **Date of effectiveness**: 2021-09-17 08:13:56 (UTC)

	Name	Function	Digitally Signed
Author	Simon Glashoff	Software Developer	2021-09-09 10:48:47 (UTC)
Reviewer 1	Anders Lorentzen	Software Developer	2021-09-09 11:47:45 (UTC)
Reviewer 2	Steffen Brummer	Senior SW Engineering Specialist	2021-09-10 11:11:17 (UTC)
Approver 1	Karen Hvid Ipsen	QA/RA Director	2021-09-10 11:56:37 (UTC)
Approver 2	Steffen Brummer	Senior SW Engineering Specialist	2021-09-13 06:05:41 (UTC)
Approver 3	N/A	N/A	N/A
QA	Mathilde Madsen	QA Engineer	2021-09-17 08:13:40 (UTC)

Content

1 Purpose

Plan for developing software for the Core Library project. This plan describes deliverables, in the form of design documentation, software implementation, verification and validation. The plan ensures conformity with [SOP-009]. This plan will be revisited and updated continuously during the development phase. The document will be created and be ready for an initial release during the planning phase.

2 Scope

The scope of this plan is the development of the Core Library project. RSP Core Library is a C++ library, developed for use in a medical device, as an open-source project under the MPL-2.0 license.

The library contains C++ classes and functions for common low-level operations, including:

- Networking
- GUI
- Logging
- Configuration
- Storage Handling

The library is intended to be reused for other medical device projects.

The entire development is performed according to IEC 62304 and 60601-1 chapter 14.

At least one released version of the library will be submitted for FDA approval with the rest of the TGM project.

3 Definitions and acronyms

- SOUP – Software of unknown provenance
- GUI – Graphical User Interface
- QA – Quality Assurance
- QC – Quality Control
- RA – Regulatory Affairs
- IQC – Internal quality control

4 Goals and objectives

By following this plan, the goal is an open-source library of reusable components for embedded Linux, designed and developed in accordance with IEC 62304.
Final documentation for regulatory submission will be contained within the software repository.

4.1 Processes

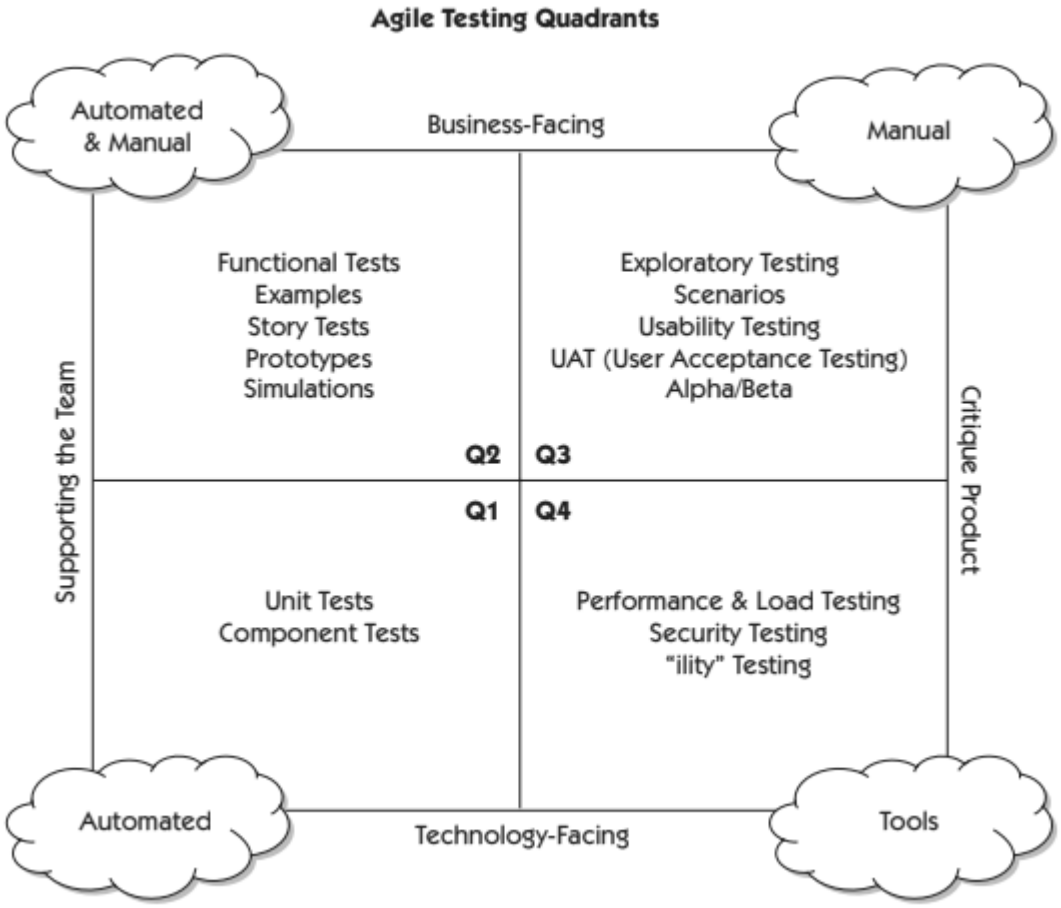
All embedded device software developed following this plan, makes use of [SOP-009 Software Development Procedure], which makes sure the design and development process is in conformance with IEC-62304.
Design control is performed using Redmine, following instructions in [SOP-0004 Redmine Usage].
All embedded software must comply with the coding style and guidelines found in Redmine project Wiki.

4.2 Deliverables

- System Architectural Design (SAD).
- Software Classification.
- List of SOUP.
- Requirement Traceability Matrix (RTM).
- Software Risk Analysis.
- Hazard Traceability Matrix.
- Software verification plans.
- Software verification reports.
- Software validation plans.
- Software validation reports.
- Pre-releases of software to support early testing of hardware solutions during development.
- Instructions for setup of automated test environment.
- Instructions for setup of development environments.
- Release report.

4.3 Verification

The agile testing quadrant is used to ensure coverage of all relevant types of verification.



4.3.1 Unit Testing

All class B and C software must have unit-tests.

Class A software can suffice with integration tests (component test).

4.3.2 Simulation Testing

As feasible the different hardware interactions and timed workflows shall be simulated via the Doctest testing framework. This improves coverage for automated regression testing.

4.3.3 Usability Testing

The software components are targeted for use by other software developers. Therefore, the final software components must be separately put through open-minded usability testing by any number of in-house developers before it is included in other software projects.
The testers must use and misuse the components, perform critical thinking, and analyses their findings with regards to product safety, quality, and customer satisfaction. Outcome from the evaluation can result in identification of new feature requirements or improved design solutions.

4.3.4 Performance and Security Testing

Performance tests will be implemented via the Doctest framework together with our unit tests and security is handled in [PLAN-0012 TGM Cybersecurity Management Plan].

4.4 Validation

The final software must be validated against the validation plans made from the top-level software requirements.

Deliverables are validated via reviews on completion, where it is validated if the output lives up to the described requirement.

All validation tests must be described as test cases in Redmine, so final test plan can be extracted.

5 Roles and responsibilities

The software department must maintain and adhere to this plan.

Developers are a limited resource; they can be assigned as a workforce on different tasks as development progresses. The individual assignment specifies which role the developer will have on a task. Developers working with implementation of a task cannot also be reviewer on the same task. Every developer has multiple roles to fill during development:

Name	Role	Responsibility
Steffen Brummer	Manager, developer, reviewer, tester, QA, QC.	Architecture, interface design, build & testing environment, embedded software, RA compliance, reviews, tests.

Simon Glashoff	Developer, reviewer, tester.	Architecture, interface design, build & testing environment, embedded software, reviews, tests.

6 Implementation

Source code repositories for all software are found on github.com/rsps/rsp-core-library. Necessary documentation will be packaged with the source code and be available in the repository directly.

The project will be developed using native C/C++ as far as is feasible, build with Make and CMake and meant to run on top of an embedded Linux distribution.
Implementation of tests are done via the Doctest framework, which is an open-source test framework and can be found at github.com/onqtam/doctest.

Tools	Version
GNU Make	4.1
CMake	3.20.2
Doctest	2.4.6
Visual Studio Code	1.59.0
GCC	7.5.0

Milestones are documented using the roadmaps feature as per [SOP-0009 Software Development Procedure] and named as following.

Milestones
Core-Library - 0.01 Library Development Planning
Core-Library - GUI Primitives & Environment
Core-Library - GUI Widgets

7 References

- SOP-0004 Redmine Usage
- SOP-009 Software Development Procedure
- PLAN-0012 TGM Cybersecurity Management Plan
- Project in Redmine: https://redmine.rspamw.dk/projects/rsp-core-lib
- Software repository: github.com/rsps/rsp-core-library

8 Attachments

N/A

9 Version History

Version	Summary of change	Initials
1.0	The initial plan for the start of the project, all template text has been exchanged with relevant information following the SOP-0009 Software Development Procedure's requirements for a development plan.	SG