OCORA

Open CCS On-board Reference Architecture

Requirement Structure
Gamma Release

Document ID: OCORA-30-009-Gamma

Version: 1.00

Date: 04.12.2020

Status: Final

Revision history

Version	Change Description	Name (Initials)	Date of change
0.01	Started document for Gamma Release	RM	2020-11-04
0.10	First Version for Gamma Release	MM	2020-11-13
0.60	Comments of SBB integrated	MM	2020-11-26
0.70	Further input from core team	MM	2020-11-27
1.00	final version for gamma release	RM	2020-12-04

Table of contents

References

The following references are used in this document:

[1] OCORA-10-001-Gamma – Release Notes

[2] OCORA-30-001-Gamma – Introduction to OCORA

[3] OCORA-30-002-Gamma – Problem Statements

[4] OCORA-30-004-Gamma – Economic Model

[5] OCORA-30-006-Gamma – High Level Methodology

[6] OCORA-30-007-Gamma – High Level Tooling

[7] OCORA-40-001-Gamma – System Architecture

[8] OCORA-40-004-Gamma – Computing Platform – Whitepaper

[9] OCORA-90-001-Gamma – Question and Answers

[10] OCORA-90-002-Gamma – Glossary

1 Introduction

1.1 Document context and purpose

This document is published as part of the OCORA Gamma release, together with the documents listed in the Release Notes [1]. It is the second release of this document and it is still in a preliminary state. It is to be understood as a state-of-the-art update of the OCORA Beta release Set-of-Requirements document.

It focuses on the process of managing the requirements within OCORA, providing better orientation for the workstreams (i.e. a clear focus on the business objectives) and transparency for the discussion with internal and external stakeholders.
The implementation of the process will be described at a later stage.

1.2 Why should I read this document?

This document is addressed to experts in the CCS domain and to any other person, interested in the OCORA concepts for CCS on-board. The reader will gain insights how the requirements will be managed.

Before reading this document, it is recommended to read the Release Notes [1] and the Introduction to OCORA [2]. The reader should also be aware of the Glossary [10].

The content provided in this document can be seen as the explanation how the Problem Statement [3] will be related to other OCORA deliverables like the Computing Platform [8] and the System Architecture [7]. It is the foundation for a more detailed requirement engineering, foreseen for the next phases of the OCORA program.

2 Purpose of Requirements Management in OCORA

Requirements and specifications are an important deliverable of OCORA. As it is a 'brown-field' approach in a regulated environment, many requirements are already given and experience with different solutions exists. To avoid a 'piling' of requirements, which may stop the innovation power for better solutions and do not contribute to the problem solving, a more structured requirements management (RM) approach is proposed. As it may add complexity at the beginning, it will be more robust and transparent during the subsequent phases.

It has the following characteristics:

- Full traceability of external (i.e. from outside OCORA) and internal requirements to anticipate their changes but also to trace back inherited problems during implementation
- Strict focus on business objectives for all high-level requirements and design decisions
- Separation in Requirement Blocks, representing the different perspectives on requirements during lifecycle
- Clustering to building blocks to support modularity and the discussion with potential suppliers

3 Requirements Structure

The following picture provides the source and its structural embedding into the requirement engineering process.

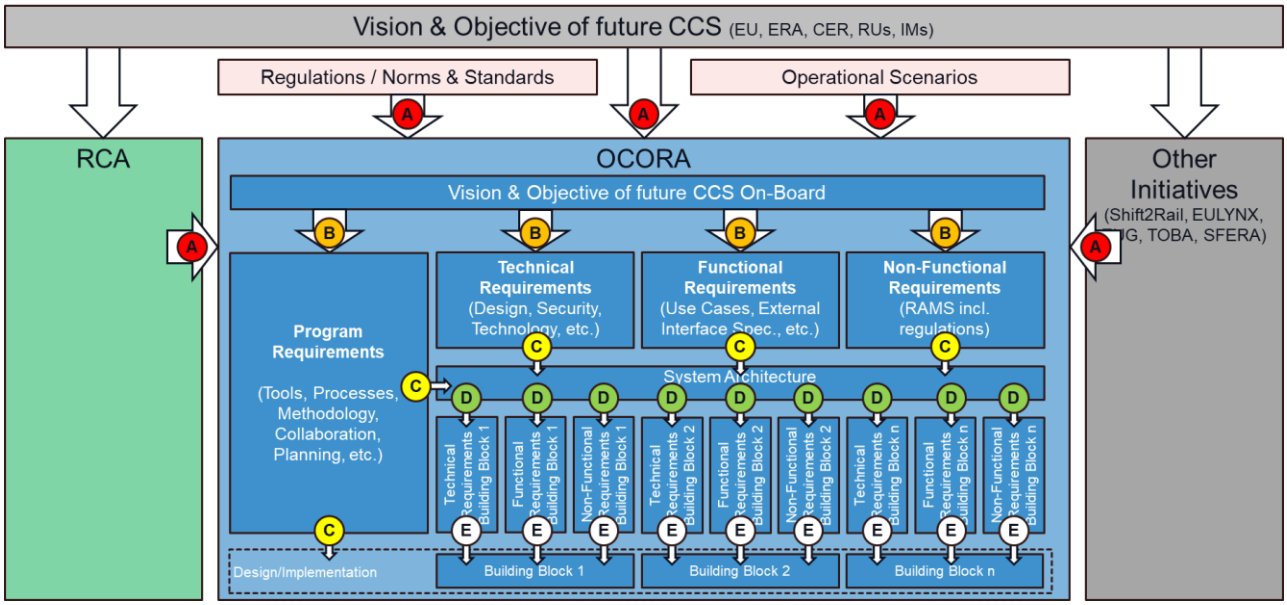

Figure 1 Requirements Structure

For a program like OCORA, the transparent management of requirements is essential. The structure of the requirements can be found in **Error! Reference source not found.**. In a nutshell, the external requirements (A) will flow traceable into the different requirement blocks of OCORA (program, technical, functional, and non-functional). Within OCORA, the Vision and Objective of future CCS on board as described in document [3] is the essential collection of pain points to be resolved. They will be converted into high level business objectives (B) in the upcoming phases of the OCORA program.

Those will then be further elaborated in the different perspectives of the major OCORA requirements blocks (program, technical, functional and non-functional) to solution principles and global requirements(C). They will also reference to selected external requirements (A) to ensure compliance.

The System Architecture and Design process will result in a topology of building blocks, where all the relevant requirements (D) can be sorted and assigned to. They will be completed with e.g. selected solutions and other information only relevant for the building block. They will form complete sets of requirements per building block (E) which will be managed in releases and will be used for tendering, design and implementation.

3.1 A - External Requirements

External requirements can be of very different nature, but they are somehow ‘given’ to OCORA.

OCORA can select them as valid source or may decide that they are not applicable. However, they need to be treated differently to the OCORA internal requirements.

Main consequence is the different change process. In case external requirements cause implementation, problems or are in contradiction with the goals of OCORA, they can’t simply be changed by OCORA. It needs some more information associated with the external requirements in the RM process to trigger and follow up this external change, including a traceability of the impact of the change and a decision what will be done for the time being. The status of that change requests needs to be transparent to the derived requirements as well.

Typical examples for changes to be managed are revisions on Subsets for the TSI, preliminary standards or technical requirements from other groups like S2R.

3.2 B - High level business objectives

As stated above, the business focus is transported via those essential requirements. They shall be limited to core business objectives and shall be derived from the pain points of the existing solution. They are giving key guidance within OCORA. Beside basic demarcations of OCORA (e.g. functional scope, vehicles targeted, roadmap) those Business Objectives (described in [3]) are converted to solution principles. There need to be a transparent mapping of the solutions to the objectives.

This mapping shall facilitate discussions within the solution principles (choice of alternatives) and shall provide stability in case solutions change or business objectives are not reached. It shall also relate the business workstreams to the technical and vice versa.

3.3 C - Solution Principles and Global Requirements

The mapped and derived solution principles and requirements will be assigned to a Requirements Block (Program, Technical, Functional, Non-Functional), depending on the character of the requirement. There may also be further elaboration of the requirements within the blocks for consistency and completeness after this sorting.

Program Requirements are of an organizational character, as there are timelines and sequences, collaboration principles or tools to be used. Technical Requirements are parameters like bandwidth, voltage levels, technologies to be used, design principles derived from the knowledge of domain experts. Functional requirements deal with use cases and given interfaces. Non-functional requirements deal with RAMSS goals.

The reason to do this separation is to understand the different dimensions of the specification process. It shall, as much as possible, separate different perspectives on the system. This opens the solution space and shall help in case technologies are changed or functionality is expanded.

However, the Requirement Blocks are handled from RM side as label. In case a requirement can't be assigned to a block, it shall be checked if it may need to be split.

As stated above, the separation is for long term lifecycle management, where underlying technologies may change (e.g. limitations of today GSM R capabilities, sensors or computation power).

3.4 D - Relevant Requirements

As the name says, the solution principles (C) and external requirements (A) are dispatched to the different parts of the system according to their relevance. There are maybe additional requirements created as consequence of the design decisions or the characteristics of the subsystem. Those created requirements must not be in contradiction. They can be a consequence of missing specification or ambiguity of external requirements and need an explanation. However, the principle of Requirement blocks is still in use and the subsystem is an additional information.

3.5 E - Sets of Requirements

This category is needed to decouple e.g. releases of requirements per building block. Before starting a development for a building block, a baseline of requirements needs to be generated. So those sets are a result of the RM process to decide which set of requirements are used for a certain project or procurement process. Those requirements must be under strict version control. Changes on Requirements over time must relate to the baselines and the changes must be visible in release notes and other materials.

4 Next Steps

The structure of the RM process depicted in this document needs to be translated in a tool-based RM process. The requirements need to be fed into this process by assigning them to the Requirement structure and by adding the requested traces and attributes. Also, the lifecycle management of RM needs a more detailed definition once enough experience is reached. The details on the methodology are described in the document [5] and the details on tooling is described in the document [6]. As described there, Polarion will be the tool for managing the requirements.