OETCS/WP2/D2.3a

ITEA 2
INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

ITEA2 Project
Call 6 11025
2012 – 2015

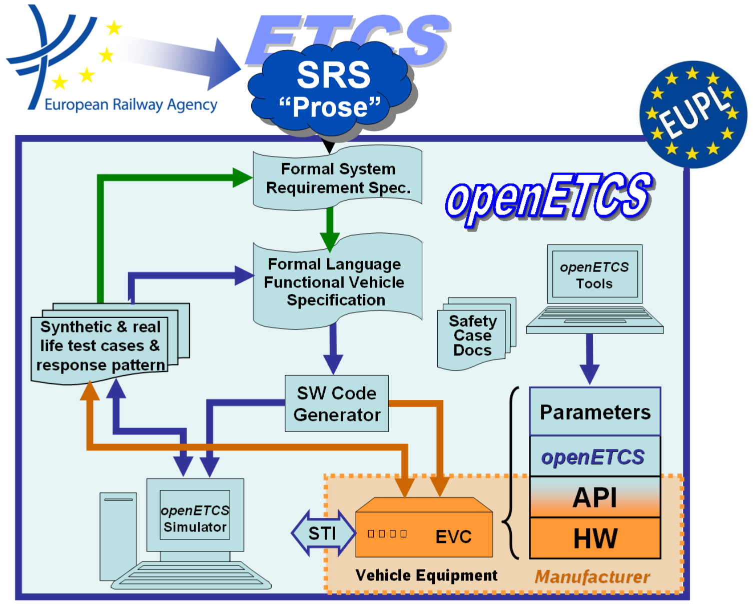
Work-Package 2: “Requirements”

Definition of the openETCS Development Process

A Standard-Compliant Process for the EVC Software Development

Hardi Hungar

June 2015
Revised September 2015
2nd Revision October 2015



Funded by:

Federal Ministry of Education and Research
République Française
Ministère de l'Enseignement Supérieur et de la Recherche
Investissements d'Avenir
Région de Bruxelles-Capitale
Gobierno de España
Ministerio de Industria, Energía y Turismo

openETCS

This page is intentionally left blank

Work-Package 2: “Requirements”

OETCS/WP2/D2.3a
June 2015
Revised September 2015
2nd Revision October 2015

Definition of the openETCS Development Process

A Standard-Compliant Process for the EVC Software Development

Document approbation

Lead author:	Technical assessor:	Quality assessor:	Project lead:
location / date	location / date	location / date	location / date
signature Hardi Hungar (German Aerospace Center)	signature Jan Welte (TU Brunswick)	signature Peter Mahlmann (DB netz)	signature Klaus-Rüdiger Hase (DB Netz)

Hardi Hungar

German Aerospace Center
Lilienthalplatz 7
38108 Brunswick, Germany

Technical Report

Prepared for openETCS@ITEA2 Project

Abstract: This documents describes a standard-compliant process suitable for the development of the software of the EVC (European Vital Computer) in the openETCS approach. It instantiates an illustrative life cycle from the EN 50128:2011 for the specific goal of openETCS. The EN 50128:2011 is the standard relevant for software development of safety-critical rail systems. The instantiation is complemented by additions concerning system aspects, i.e., issues beyond software development. These additions are in line with the EN 50126-1:1999, the standard addressing the development of safety-critical rail systems.

The main concretions and modifications to the software development as described in the EN 50128:2011 are:

- The definition of a system phase which generates the input necessary for the development of an EVC as a sub-system of the ETCS system from available background material.
- A sub-system (EVC) phase defining an interface of the HW to enable a SW development without considering the HW in detail.
- A high-level definition of the choice of implementation language, techniques and measures for the SW development and its verification.
- A validation concept for the SW without prior SW/HW integration based on simulation.
- A sketch of an iterative implementation of the process phases (in contrast to a sequential implementation where each phase is completed before the next begins).

Disclaimer: This work is licensed under the "openETCS Open License Terms" (oOLT) dual Licensing: European Union Public Licence (EUPL v.1.1+) AND Creative Commons Attribution-ShareAlike 3.0 – (cc by-sa 3.0)

Modification History

Version / Date	Section	Modification / Description	Author
01 / June 23, 2015	all	initial	Hungar
02 / September 02, 2015	all	incorporated comments by Jan Welte and completed the coverage of D2.3 according to the analysis by Hardi Hungar **Additionally:** Changed responsibilities for component test.	Hungar
03 / October 26, 2015	2, 3.7	Corrected Fig. 1, added verification to Phase 7	Hungar

Table of Contents

Figures and Tables

Figures

Tables

1 Introduction

1.1 Document Context

1.1.1 openETCS Project and openETCS Activity

This document is a contribution to the ITEA 2 project within the context of the openETCS activity.

Here, it is distinguished between the current *openETCS project*, funded as part of the EUREKA cluster programme ITEA 2, and the *openETCS activity* as a whole, which encompasses the project.

The openETCS activity pursues the vision of a full CENELEC compliant development of open-source software for the *European Vital Computer* (EVC). It is intended to produce a software kernel which can be used in commercial EVCs. Also, a development method with suitable tool support shall be defined. The project is a major first step in realizing the goals of the activity. Though the project cannot realize all of the vision, it shall produce partial results which can subsequently be taken up to achieve that goal.

1.1.2 Scope of the openETCS Development

The starting point is the ETCS specification, mainly Subset 026 [1]. From that, and a hardware interface description, the SW requirements are to be derived in a (logically) first development phase. The result shall be a generic SW covering the full functionality described in [1], but without additional features which are usually required for a product. Such additional features include functions addressing national specifics or customer extensions. Since there are no plans yet to deploy the SW to a dedicated HW, the SW cannot be validated on the target HW as foreseen by the CENELEC standards. Instead, the development ends with a SW validation via simulation.

1.1.3 Role of the Document

This document provides a high-level view of a suitable process for the openETCS activity. The project, which does only part of the development, shall follow the description given here. The results of the project shall be artifacts of this process or parts of such artifacts, and they shall be labeled accordingly. What is not done completely shall be delineated. All of this shall result in a set of artifacts which forms a useful basis for subsequent activities. These activities shall be able, by addressing the missing points, to complete a documented software development suitable for integration in a system which can be certified.

This document updates and extends the definition of the openETCS process given in [2]. That specification dates from June 2013 and does not cover completely the current views. D2.3 is extended in providing more detail on how to structure the development compliant to the CENELEC standards, and on the process steps themselves.

Some development aspects described in D2.3 are not covered here. They concern mainly:

- A semi-formal model of the ETCS system. This is not done in the project and not necessary for developing the EVC software.

- Formal sub-models of the ETCS system, in particular of the EVC software functions. Some such activities are pursued in the project, but this is considered to be merely a specific kind of implementing process steps and thus need not be included in an overall process definition.
- Applications of formal methods, in particular for safety-related parts. Again, this is not done on a large scale and merely one of several approaches pursued. Thus, it is not necessary to be detailed here.

These items may be covered in a supplementary document describing ambitious activities of openETCS following the current project.

1.1.4 Reference Material

The main reference for the openETCS process is the standard for developing software for railway control and protection systems, the EN 50128:2011 [3]. For the system context, the EN 50126-1:1999 [4] is taken into account, which regulates the specification and demonstration of reliability, availability, maintainability and safety of railway applications in general (of which SW is a part). In particular, this standard is relevant for the phases before the SW development. The standard EN 50129:2010 [5], which concerns the safety case, is not relevant for this description. It has to be considered for the implementation of all safety-related activities listed here.

From within the project, the definition of the openETCS requirements from D2.6 [6] are relevant for an elaboration of Sec. 4, which sketches the implementation of the openETCS process.

1.2 Scope of the Document

1.2.1 Items Covered

This document defines the structure of the SW development process of openETCS and its stages. It references and summarizes the main requirements originating from the standards. It lists artifacts to be produced and activities to be performed. It includes an instantiation of the system phases necessary to derive the software requirements. This instantiation is done in a form tailored to openETCS where several specifications are already available (mainly the UNISIG documents).

1.2.2 Items not Covered

The document does not include full detail on the means (methods and tools) to be employed. This is described (at least for the concerns of the ITEA 2 project) elsewhere:

1. The *primary openETCS toolchain* for software design given by [7]
2. the *secondary openETCS toolchain* for verification and validation described in [8]

It also does not address organizational issues of the SW development, i.e. how roles like Designer, Verifier etc. are to be filled, as defined in the EN 50128. These organizational issues will have to be addressed during activities following the ITEA 2 project. This will be necessary for integrating the resulting SW into an actual product and having the result approved.

The description of activities and requirements on artifact content given here does not provide full detail. In planning and performing any of the process steps defined here, it is recommended to consider the respectively relevant standard(s) to ensure full standard compliance. References to the standards are provided.

The document does not specify in which way the results of the current openETCS project will not cover the full functionality of the EVC, and how it is planned to deal with this incompleteness. Each design artifact produced shall define its scope, including a statement of what is missing for a full development.

1.3 General Description of the openETCS Development Process

The openETCS SW development process follows the Illustrative Development Lifecycle 2 [3, Fig. 4] of the EN 50128 and the detailed description for "Generic software development" in Sec. 7 of that standard. It deviates mainly in start and end phases from that lifecycle. These phases are adapted to the specifics of the openETCS goals, as not a full system will be developed (no hardware) and a substantial body of background material concerning the EVC and its environment is available. Additionally, as s a minor deviation, the SW component design phase has been made part of the SW architecture and design phase. This deviation reflects the form of the development within the openETCS project, but it is of not much significance and could easily be undone.

1.3.1 Terminology

System refers to the ETCS system as described in Subset 026 [1]. This includes the onboard unit, the European Vital Computer (EVC). The term *sub-system* means this onboard unit. This terminology has been introduced in the first description of the openETCS process [2]. A list of abbreviations and terms is given in Sec. 5.

1.3.2 Presentation of the Process

The presentation starts with an overview of the process. For each of the phases of the process, the following aspects are described. These same aspects are defined in the standard.

Objectives The goals of the phase

Activities Description of what has to be done

Input Lists artifacts and external source material

Output Process artifacts which are either produced or substantially altered by the activities

Requirements Short recapture of the requirements of the EN 50126 or EN 50128 for that phase, and how they are to be met by the deliverables.[1]

The process is described as a top-down, waterfall procedure. The implementation within the openETCS project deviates from that, as the phases of the process are performed in an overlapping, iterative style. How that is done is indicated in a separate section (Sec. 4). The documentation of the development which is to be produced shall not reflect the iterative implementation of the process, but describe its results as if they had been achieved in the top-down, waterfall way of the description in Sec. 3 (as foreseen in [3, 7.1.2.2]). Then, a completed set of artifacts would be in accordance with the main requirements of the CENELEC on the documentation.[2]

[1] Usually, the recapture does not provide full detail. Thus, one has to refer to the standards when planning the activities of a phase.

[2] Additionally, it will be necessary to address organizational requirements adequately.

2 List of Artifacts and Phases

This section lists all main artifacts which are to be produced to document the development and provide the basis for a safety case of a product using the openETCS project results. Further detail on the content and role of these artifacts is given in the text describing the phases where these artifacts are to be produced or updated.

The artifacts in the list are numbered in the format "*m-n*", where "*m*" is the number of the phase where the first version of the artifact is produced, and "*n*" is a unique number (roughly the sequence number the artifact would get in a linear implementation of the process). For each artifact, the role mainly responsible for its creation is defined in accordance with [3, Tab. C.1]. The roles are taken from [3, Sec. 5.1 / Fig. 2] and listed below. These roles have also been assigned to the artifacts related to the system phases, i.e. the scope of the [4] which does not mention roles. If two roles are given for one artifact, they are responsible for different parts.

PM Project Manager

RQM Requirements Manager

DES Designer

IMP Implementer

INT Integrator

TST Tester

VER Verifier

VAL Validator

CM Configuration Manager

Note that implementing independence requirements according to [3, 5.1.2.10] is necessary for a SIL-4 development. As this is not done in the project, subsequent activities with the goal of producing code usable in a SIL-4 system will have to care for that.

Generally, the Validator checks the results of the Verifier, and vice versa. This is made explicit for the planning phase, where the Validator has to check, besides the verification report, the "Verification Plan" written by the Verifier.

Phase 0: Planning

0-00 Project Plan (PM) A management structure for the development, including a documentation of the assignment of roles and personnel to activities.

0-01 Quality Assurance Plan (PM) Definition of all activities to ensure the quality of the openETCS development (RAMS, functionality etc.). This document, which defines the lifecycle, is a part of the Quality Assurance Plan.

0-02 Configuration Management Plan (CM) A document which details how to handle versions (documents, software, tools) in the development.

0-03 Verification Plan (VER) Definition of all verification activities and how they are to be documented, and the methods and tools for verification.

0-04 Validation Plan (VAL) Definition of how to demonstrate that the end result (EVC software) serves its purpose within the context of the ETCS system.

0-05 Planning Verification Report (VER, VAL) Documentation of the verification of the planning documents.

Phase 1: System Design

1-06 Documentation of Coverage by Background Material (PM,RQM) A listing of relevant available background material and a documentation of what is used for which purpose in openETCS. The RQM is concerned with the requirements, the PM handles all else.

1-07 Elaborated System Requirements (RQM) Corrections and additions to background requirement definitions, including

- System and sub-system functionality
- System RAMS policy

Together with requirements from the background material, this shall define all system requirements which are necessary to derive the sub-system software requirements.

1-08 Risk Assessment (PM) Documentation of the results of the analysis and evaluation of risks, including a hazard log for safety management.

1-09 Safety Plan (PM) Definition of activities to ensure safety of the openETCS development result (in the context of the ETCS system).

1-10 Sub-System Requirement Specification (RQM) The requirements on the EVC.

1-11 Sub-System Safety Specification (RQM) The safety requirements on the EVC.

1-12 System Design Verification Report (VER) Documentation of the verification of all activities and results of the first openETCS process phase.

Phase 2: Sub-System Architecture Design

2-13 Sub-System Architecture Design (DES) The SW/HW architecture of the EVC is described. The requirements and safety requirements are attributed to the components of the architecture.

2-14 SW Acceptance Plan (VAL) Definition of acceptance criteria and acceptance procedure for the EVC SW.

2-15 Sub-System Architecture and Design Verification Report (VER) Report on the verification of the results of Phase 2.

Phase 3: SW Specification

3-16 SW Requirements Specification (REQ) A complete and consistent specification of the SW requirements, including the safety requirements in a specific section.

3-17 Overall SW Test Specification (TST) Specification of tests checking the conformity of the sub-system software to its specification.

3-18 SW Specification Verification Report (VER) Report on the verification of the results of Phase 3.

Phase 4: SW Design

4-19 SW Architecture and Design Specification (DES) This document combines SW architecture and a detailed SW design. It contains a definition of the coding principles and rules.

4-20 SW Interface Specification (DES) A detailed specification of the interfaces of the overall SW and of the SW components. This complements the "SW Architecture and Design Specification".

4-21 SW Integration Test Specification (INT) Specification of tests and test procedure to check that the SW components interact correctly.

4-22 SW Component Test Specification (TST) Specification of white-box tests on module level and black-box tests on component level.

4-23 SW Design Verification Report (VER) Report on the verification of the results of Phase 4.

Phase 5: SW Component Implementation and Test

5-24 SW Components (IMP) The SW components, provided as SCADE models with code generated from them, and documented white-box tests.

5-25 SW Component Test Report (TST) Documentation of the component tests.

5-26 SW Component Verification Report (VER) Report on the verification of the results of Phase 5.

Phase 6: SW Integration

6-27 SW Integration Test Report (INT) Documentation of the integration test.

6-28 SW Integration Verification Report (VER) Report on the verification of the results of Phase 6.

Phase 7: SW Validation

7-29 Overall SW Test Report (TST) Report on the conformity test of the SW.

7-30 SW Validation Report (VAL, VER) Report on the validation of the SW, including the validation of the tools used in the process.

3 The openETCS Development Lifecycle

Fig. 1 gives an overview of the openETCS development lifecycle. It has been derived from Fig. 1 from [3]. The main differences to that standard consist in:

- A system phase has been added which covers the activities preceding the sub-system development (starting from the sub-system requirements)
- The "SW architecture and design phase" and the "SW component design phase" have been combined into one phase. This is purely organizational. All information of the standard approach still have to be generated.

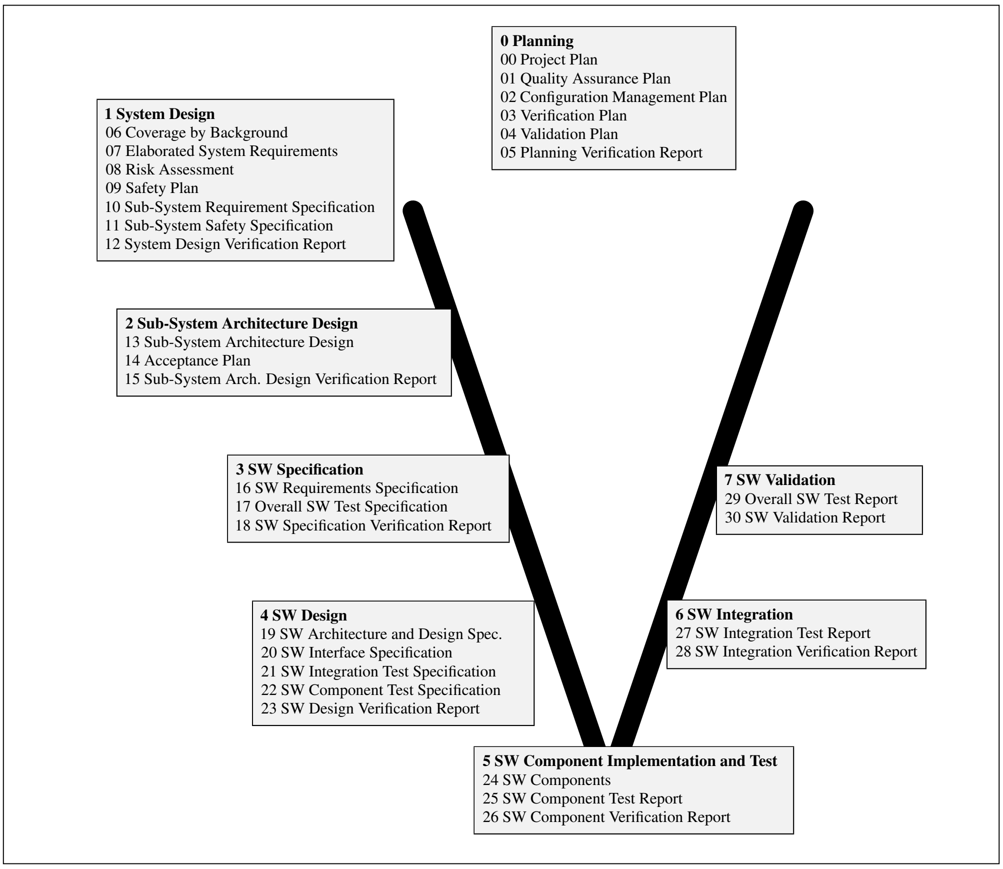

0 Planning
00 Project Plan
01 Quality Assurance Plan
02 Configuration Management Plan
03 Verification Plan
04 Validation Plan
05 Planning Verification Report
1 System Design
06 Coverage by Background
07 Elaborated System Requirements
08 Risk Assessment
09 Safety Plan
10 Sub-System Requirement Specification
11 Sub-System Safety Specification
12 System Design Verification Report
2 Sub-System Architecture Design
13 Sub-System Architecture Design
14 Acceptance Plan
15 Sub-System Arch. Design Verification Report
3 SW Specification
16 SW Requirements Specification
17 Overall SW Test Specification
18 SW Specification Verification Report
7 SW Validation
29 Overall SW Test Report
30 SW Validation Report
4 SW Design
19 SW Architecture and Design Spec.
20 SW Interface Specification
21 SW Integration Test Specification
22 SW Component Test Specification
23 SW Design Verification Report
6 SW Integration
27 SW Integration Test Report
28 SW Integration Verification Report
5 SW Component Implementation and Test
24 SW Components
25 SW Component Test Report
26 SW Component Verification Report

Figure 1. openETCS Development Lifecycle

- The “SW component implementation phase” and “SW component testing phase” have been combined into one. This is motivated by changing the responsibility for performing component tests to the Implementer and letting the Tester only check the results. This decision is in accordance with [3, 6.1.4.1 to 6.1.4.5] (see below). But nevertheless it might be reconsidered in the process implementation.

- According to the standard, SW validation should happen *after* SW/HW integration. Since it would contradict the goals of openETCS to fix the hardware, the SW cannot be deployed to the target HW and thus not be validated on it. To validate the SW within openETCS, the best option seems to be to do validation within simulated hardware.

- Phases after SW validation are outside the current scope of openETCS, though a process for SW maintenance should be installed in the long run.

The phases are detailed below. For each phase, its objectives and the activities to reach these objectives are given. Inputs and outputs are listed, the latter usually with some more detail than in the table above. Finally, all requirements of the relevant standard on the respective phase are summarized together with a reference to the full text. If the phase selects a particular alternative, a rationale is given.

3.0 Planning

3.0.1 Objectives

Generate first versions of the planning documents.

3.0.2 Activities

Review the ideas and concepts of the openETCS approach in view of the standard requirements (EN 50126 and EN 50128). Design a lifecycle for the realization of the EVC software and define the activities (design, verification, validation). Make preliminary choices for methods and tools. The resulting documents are verified (VER), the verification itself is checked (VAL).

3.0.3 Input

The CENELEC standards, UNISIG specifications and the ITEA 2 project application documents [9].

3.0.4 Output

First versions of the following documents.

1. **Project Plan (PM)** A management structure for the development, including a documentation of the assignment of roles and personnel to activities. For details, it refers to other planning documents. Deviating from the standard requirements, in the openETCS project there is no fixed assignment of personnel to roles. The project outcome is not intended to be taken directly to SIL 4, so this is not strictly necessary. It will have to be considered in ensuing activities.

2. **Quality Assurance Plan (PM)** Definition of all activities to ensure the quality of the openETCS development (RAMS, functionality etc.). This plan has supplements providing details on the lifecycle (this document), the tools and methods for design, modeling and implementation, as well as the necessary tool qualification.

3. **openETCS Process (PM)** (this document) This is a supplement to the “Quality Assurance Plan” and provides a consistent, high-level view on a standard-compliant development process, taylored to the goals and approach of the openETCS activity.

4. **Definition of the tools and methods for software development (PM)**: This is a supplement to the “Quality Assurance Plan”. It defines the openETCS “Primary Tool Chain”: Methods, languages and tools for SW design and implementation, and their qualifications.

5. **Configuration Management Plan (CM)** A document which details how to handle versions (documents, software, tools) in the development.

6. **Verification Plan (VER)** Definition of all verification activities (checks to establish the adequacy, completeness and correctness of each phase) and how they are to be documented. It also addresses methods and tools.

7. **Validation Plan (VAL)** Definition of how to demonstrate that the end result (EVC software) serves its purpose within the context of the ETCS system. The plan includes a description the means to perform the validation and the techniques employed.

8. **Planning Verification Report (VER,VAL)** Documentation of the verification of the planning documents. This is done by the Verifier except for the “Verification Report”, which is checked by the validator.

All of these documents are to be updated in later phases of the development to include additions and concretions and to reflect any changes which are made in the plans. All changes shall be made in the plans before the respective activities are performed. Concretions will concern in particular, but not exclusively, tools and methods for development, verification and validation.

3.0.5 Requirements

The requirements to be met are defined mainly in [3, Sec. 5, 7.1], with additional detail on SW quality assurance in Sec. 6.

Req 0.1, [3, Sec. 5.1]: Organisation, roles and responsibilities are described in the “Project Plan” and “Quality Assurance Plan”.

Req 0.2, [3, Sec. 5.2]: Personnel competence is to be addressed in development activities taking up the results of the openETCS project.

Req 0.3, [3, Sec. 5.3]: The lifecycle is defined in this document.

Req 0.4, [3, Sec. 7.1, 5.3.2.5]: The phases are implemented in an iterative, overlapping fashion (see Sec. 4. All activities which are performed in the development are planned beforehand.

Req 0.5, [3, Sec. 6]: The quality assurance requirements shall be regarded in writing the plans for quality assurance, verification and validation.

3.1 System Design

3.1.1 Objectives

The main goal of this phase is to generate everything that is necessary to start the development of the sub-system, i.e. the EVC, from the available input (Subset 026 and further material). This phase shall cover the first five phases of the RAMS lifecycle of the ETCS system as defined according to the EN 50126:

1. Concept
2. System definition and application conditions
3. Risk analysis
4. System requirements
5. Apportionment of system requirements

The combination of five phases of the EN 50126 life cycle into only one is motivated by the fact that not a full development needs to be performed for openETCS. First, much is already available in background documents. Further, only what is relevant to the EVC software has to be provided for the subsequent development. Thus, it suffices to complete the items which are inherited (at least partly) by the requirements on the EVC SW, and those items which are related to the integration of the SW into the ETCS system (e.g., interfaces and assumption that are made about the rest of the system).

3.1.2 Activities

Identification of Coverage by Background Material

The available background material to cover requirements of the EN 50126 is identified. “Documentation of Coverage by Background Material” provides a detailed description of how this material covers requirements. Sec. 3.1.5 below lists what is expected to be covered. For each instance where this expectations turns out to be wrong, the respective requirement has to be met by some dedicated activity. The outcome shall be referenced in the “Documentation of Coverage by Background Material”.

System Requirement Elaboration

The available requirement specification of the ETCS system (mainly given by Subset 026) relevant to the EVC is identified, analyzed and extended by corrections and formalizations. This activity shall produce the “Elaborated System Requirements”. The analysis may use semi-formal or formal modeling and analyses of the models.

Sub-System Requirement Specification

A definition of the sub-system boundaries and interfaces is derived, and those of the system requirements that are to be allocated at least partially to the sub-system are identified. From that, a sub-system requirement specification is developed. This specification may be text-based, or involve formalizations (e.g., models) as available. A Sub-System Safety Specification shall be a separate document or a separate section. The safety specification is based on risk analysis activities. Output of this activity are “Sub-System Requirement Specification” and “Sub-System Safety Specification”.

System Risk Analysis and Sub-System Risk Identification

Hazards and risks of the ETCS system are identified. This must cover at least those which involve the EVC, and the way these are related to hazards and risks in the full ETCS system. Existing sources for such analyses may be used. The results shall be documented as part of the Risk Analysis and Hazard Log.

Sub-System Safety, Acceptance and Quality Assurance Planning

This planning shall take into account that openETCS is concerned with the software of the sub-system. It is therefore sufficient to delineate the plans on the sub-system level. These delineations shall be parts of the respective plans of openETCS, i.e., the Safety Plan, Acceptance Plan and Quality Assurance Plan.

Verification

The results of this phase shall be verified w.r.t.:

- selection of the right input information
- adequacy of the methods and tools used in the derivation
- correctness
- adequacy and completeness for the respective purpose

3.1.3 Input

The relevant input includes the UNISIG specification documents [1, 10, 11, 12, 13], which are based on the Commission Decision 2012/88/EU on the technical specification for interoperability relating to the control-command and signalling subsystems of the trans-European rail system [14]. Additional sources may come from national instantiations of ETCS track side equipment (e.g., Deutsche Bahn, SNCF, ProRail) and operating rules.

3.1.4 Output

1. **Documentation of Coverage by Background Material (RQM)** A listing of relevant available background material and a documentation of what is used for which purpose in openETCS. These purposes include:
   (a) System scope, context, purpose and environment
   (b) System RAMS targets, policy
   (c) System hazards and risks
2. **Elaborated System Requirement Specification (RQM)** Corrections and additions to background requirement definitions, including
   (a) System and sub-system functionality
   (b) System RAMS policy
3. **Sub-System Requirement Specification (RQM)** The requirements on the EVC. This artifact defines the interface and the data flow over the interface, functionality of the EVC, and extra-functional requirements. It may refer to external documents for parts of this definition.
4. **Risk Assessment** (PM) Documentation of the results of the analysis and evaluation of risks, including a hazard log for safety management. The project manager has to organize the risk analysis and evaluation leading to the creation of the “Risk Assessment” document, engaging all necessary parties. The “Risk Assessment” shall include:
   (a) A documentation of the system risk analysis and evaluation performed
   (b) An identification of risks relevant for the sub-system
   (c) A hazard log for the sub-system
5. **Sub-System Safety Specification (RQM)** The safety requirements on the EVC.
6. Sub-System Safety Plan (part of the “Safety Plan)”
7. Sub-System Acceptance Plan (part of the “Acceptance Plan”)
8. Sub-System Quality Assurance Plan (part of the “Quality Assurance Plan”)
9. Sub-System Hazard Log (part of the “Sub-System Safety Specification”)
10. **System Design Verification Report (VER)** Documentation of the verification of all activities and results of the first openETCS process phase.

3.1.5 Requirements

The following text lists the requirements as defined in the EN 50126 for those 50126 phases included in this openETCS phase. For each requirement it is said whether it is (expected to be) covered by background material, or the designated outputs and the activities described above. Requirements of the EN 50126 are referenced by their identifying numbers. "**Req 6.1.3.4**" refers to the item **6.1.3.4** in the EN 50126. This is "Requirement **4**" of the phase "Concept", described in Sec. **6.1** of the EN 50126, whose requirements are listed in Sec. **6.1.3**.

The coverage by background material shall be detailed in "Documentation of Coverage by Background Material", if necessary with adequate justifications. A general requirement is that full bidirectional traceability between the requirements in the UNISIG specifications (those which refer to the sub-system) and the "Sub-System Requirements Specification" and "Sub-System Safety Specification" shall be realized.

Phase "50126 6-1 Concept"

As the ETCS system has already been implemented in numerous instances, most items of the Concept Phase are expected to be covered by background material.

Req 1.1, [4, 6.1.3.1] System scope, context, purpose and environment: Background

Req 1.2, [4, 6.1.3.2] RAMS implications of financial aspects and feasibility studies: Background

Req 1.3, [4, 6.1.3.3] Hazard sources: Background

Req 1.4, [4, 6.1.3.4] Information about RAMS performance, regulations and safety targets: Background

Req 1.5, [4, 6.1.3.5] Definition of the management requirements: "Quality Assurance Plan"

Phase "50126 6.2 System definition and application conditions"

Req 1.6, [4, 6.2.3.1] Mission profile, boundary, application conditions and hazards of the system: Background

Req 1.7, [4, 6.2.3.2] Preliminary RAM analysis and hazard identification: Background

Req 1.8, [4, 6.2.3.3] RAMS policy for the system: Background

Req 1.9, [4, 6.2.3.4] System Safety Plan: To be included in the "Safety Plan" (to the extent which is necessary)

Phase "50126 6.3 Risk analysis"

A risk analysis is performed for all items which concern the EVC SW and how they relate to risks and hazards of the full system. A hazard log with all relevant items enters the Sub-System Safety Specification. The risk analysis may be based on background material.

Phase “50126 6.4 System requirements”

Req 1.10, [4, 6.4.3.1] Specification of RAMS requirements: Covered by “Elaborated System Requirements”

Req 1.11, [4, 6.4.3.2] Specification how to achieve compliance with the RAMS requirements: “Safety Plan”, “Validation Plan” and “SW Acceptance Plan”

Req 1.12, [4, 6.4.3.3] Detailed RAM Programme: “Verification Plan”, “Quality Assurance Plan”

Req 1.13, [4, 6.4.3.4] Updated Safety Plan: “Safety Plan”

Phase “50126 6.5 Apportionment of system requirements”

Req 1.14 [4, 6.5.3.1] Allocate system requirements to sub-systems: The “Sub-System Requirement Specification” and the “Sub-System Safety Specification” address the parts relevant to openETCS, the rest should be covered by background material

Req 1.15 [4, 6.5.3.2] Acceptance criteria for all sub-systems: the openETCS-relevant part is to be spelled out in the “Acceptance Plan”

Req 1.16 [4, 6.5.3.3] Updated Safety and Validation Plan: “Safety Plan” and “Validation Plan”

3.2 Sub-System Architecture Design

3.2.1 Objectives

This phase shall complete the preparations for the main focus of openETCS, the development of the EVC software. It shall define the SW/HW architecture, the SW/HW interface and interaction, and allocate the system and system safety requirements to the architecture components.

The last sub-phase of “System Design”, the “Apportionment of system requirements”, allocated requirements to the sub-system. Here, these are further refined to the SW/HW architecture of the sub-system. The subsequent phase “SW Specification” completes the derivation of SW requirements (of the EVC) from the system requirements (of ETCS).

3.2.2 Activities

The hardware architecture of the system is described, and the SW/HW interface is defined. The hardware is described at least generically, so that the it can be shown that the EVC functionality can be realized on that architecture. The requirements and safety requirements which concern the software are identified. Risk analysis and planning documents are revised as necessary.

3.2.3 Input

The main phase-specific inputs are:

1. “Sub-System Requirement Specification”
2. “Sub-System Safety Specification”
3. UNISIG specifications of the architecture of the EVC

3.2.4 Output

1. **Sub-System Architecture Design Specification (DES)**: The SW/HW architecture of the EVC is described. The requirements and safety requirements are attributed to the components of the architecture. The HW architecture is described on a high level. Its interface to the SW is given by a detailed HW API, describing in detail the cooperation and data flow between SW and HW.
2. **SW Acceptance Plan (VAL)** Definition of acceptance criteria and acceptance procedure for the EVC SW.
3. **Sub-System Architecture Design Verification Report (VER)**: The verification shall check that the specification describes a realizable design, that the requirement allocation is complete and consistent with the architecture details, and that all aspects, in particular safety, have been adequately taken care of.

3.2.5 Requirements

The requirements to be covered come from the EN 50126, Phase 5, "Apportionment of system requirements".

Req 2.1, [4, 6.5.3.1] Allocation of requirements (also safety) to sub-systems and specification of the sub-systems: The allocation is done in the "Sub-System Architecture Design Specification". The HW is described in its structure, and its interface to the SW is specified in detail. This enables a specification of the SW (including the safety aspect) in the subsequent phase. As the hardware is not in the focus of openETCS, the interface description is sufficient for the further openETCS development.

Req 2.2, [4, 6.5.3.2] Specification of acceptance criteria for the architecture components. For the SW, this is done in the subsequent phase. The HW must provide the specified interface (API) to the SW. Besides that, detailed compliance criteria for the HW are to be provided by entities developing it. This is sufficient for the purposes of openETCS, as the rest of the system is outside the scope of openETCS.

Req 2.3, [4, 6.5.3.3] Reviews and updates to the "Safety Plan" and "Validation Plan". These are done to the extent necessary (see "Activities" above).

3.3 SW Specification

As mentioned above in Sec. 3.2.1, this phase completes what is to be done in Phase 5 of the EN 50126. In the EN 50128, the SW requirement are defined in a separate phase. We adopt this structuring, here. It goes well with the general idea of the openETCS project, which focuses the SW development.

3.3.1 Objectives

A complete and consistent set of requirements, with specifically designated safety requirements, for the SW shall be defined. This is to be complemented by a definition of acceptance criteria for the SW.

3.3.2 Activities

An analysis of the requirements which have been allocated completely or partially to the SW in the "Sub-System Architecture Design Specification" is done. The goal of the analysis is to extract the SW requirements. In addition to the requirement documents produced in previous process steps, the background material referenced in "Documentation of Coverage by Background Material" is a source for requirements on the SW.

Test cases are collected which cover functionality and performance of each function, and in operational scenarios the interplay of the functions.

3.3.3 Input

1. "Documentation of Coverage by Background Material"
2. Background material: UNISIG specifications [1, 10, 11, 12, 13] and further material as listed in the documentation of coverage by background material.
3. "Elaborated System Requirement Specification"
4. "Sub-System Requirement Specification"
5. "Sub-System Safety Specification"
6. "Sub-System Architecture Design Specification"

3.3.4 Output

1. **SW Requirements Specification (DES)** A complete and consistent specification of the SW requirements, including the safety requirements in a specific section.
2. **Overall SW Test Specification (TST)** Specification of tests checking the conformity of the sub-system software to its specification.
3. **SW Specification Verification Report (VER)** Report on the verification of the results of SW Requirements Phase.

3.3.5 Requirements

The requirements are derived from [3, Sec. 7.2.4] and adapted to the EVC SW.

Req 3.1 The "SW Requirements Specification" shall

a address functionality, robustness, maintainability, safety, efficiency, portability, self checking, testing in operation,

b be complete, clear, precise, unequivocal, verifiable, testable, maintainable, feasible, back traceable, understandable

c list all interfaces, all HW-related issues and modes of operation

d distinguish safety-related from other functions

e use formalizations for complex logical and numerical functionalities and semi-formal modeling for structural representations

Req 3.2 The “Overall SW Test Specification” shall

a include functional black-box tests based on a boundary value analysis and apply input partitioning according to equivalence classes of inputs according to the specification

b include performance tests for response times

c be specified in more detail in the “Verification Plan” with rationales for the adequacy of the chosen measures

Req 3.3 The “SW Specification Verification Report” shall be specified in the “Verification Plan”.

3.4 SW Design

3.4.1 Objectives

A structure of the SW is defined, with components whose combination will realize the SW requirements. The interplay of the components is specified, in a form that it can be checked on the implementation. Safety requirements are treated with adequate care so that the contribution of each component to the safety function of the SW is unambiguously clear and safety is verifiable. The description must enable implementors to independently develop components and provide them with references to all means to do so.

3.4.2 Activities

The SW requirements are analyzed to generate a suitable decomposition into components realizing subfunctions. An execution paradigm for the whole SW and the interplay of components (timing, execution order, communication paradigm) is defined. The SW interfaces are specified precisely. Languages, coding rules and tools for the implementation are chosen. A test strategy for integration testing is chosen and the integration and component tests are specified. The adequacy and correctness of the results is verified.

3.4.3 Input

The main phase-specific inputs are:

1. **SW Requirements Specification**
2. **Sub-System Architecture Design** provides the external interface of the SW via the HW API definition.

3.4.4 Output

1. **SW Architecture and Design Specification (DES)** This artifact covers SW architecture, overall SW design and detailed component design in the form of detailed specifications. All functionality is described in its realization by independent functions. The coding principles and rules for implementing the components are defined. In particular, this shall address the use of the SCADE Suite. The artifact may be split over several documents.

2. **SW Interface Specification (DES)** This complements the "SW Architecture and Design Specification" by providing detailed interface specifications of the overall SW and of the SW components.

3. **SW Integration Test Specification (INT)** Specification of the procedure for integrating the SW, testing that the integrated SW components interact correctly, and that the SW can run on the specified HW. The SW integration test may reuse tests from the component tests. The integration with HW is tested via a HW simulation.

4. **SW Component Test Specification (TST)** On the level of modules, the specification defines a general procedure for white-box testing which is to be instantiated with adequate tests for each module. The procedure includes a coverage criterion for SCADE models which is to be achieved by the tests. On the level of components, black-box tests checking the correct implementation of the functionality of the components are specified.

5. **SW Design Verification Report (VER, VAL)** Report on the verification of the results of this phase.

3.4.5 Requirements

Req 4.1 [3, 7.3.4.2 to 7.4.3.17] The "SW Architecture and Design Specification" defines the structure of the SW addressing feasibility, HW/SW interaction, components, scheduling, the paradigm for component communication and interaction, safety, fault handling, the use of prototype models, development strategy, techniques and measures. No pre-existing SW will be used. The architecture measures to be taken according to [3, A.3] are Defensive Programming, Diverse Programming, Fully Defined Interfaces, Modeling and Structured Methodology. *Rationale: This is one of the approved combination ("1) a)" with "Modeling") of techniques. To use "Error Detecting Codes" as in combination "1) b)" would not make sense, so "1) a)" is chosen. "Modeling" is a technique which is applied in the openETCS approach, so it is taken as the complementary technique for "1) a)".*

Req 4.2 [3, 7.3.4.18 to 7.4.3.20]: The "SW Interface Specification" addresses input and output invariants (including bounds) and measures in case of invariant violation. The communication mechanism are detailed in the SW architecture and design.

Req 4.3 [3, 7.3.4.21 to 7.4.3.24, 7.4.4.1 to 7.4.4.6]: The overall SW and component design uses, from [3, A.4], Modeling, a modular approach, design and coding standards and a strongly typed language. *Rationale: This is one of the approved combination of techniques, with "Modeling" chosen instead of the alternative "Formal Methods".* Components shall have fully defined interfaces with a general strategy of explicitly specified ways of data access. *Rationale: This is in accordance with the requirements of [3, A.20].* The SCADE Suite Advanced Modeler is used for implementation. Its language is strongly-typed. The style of modeling for the software and component design is to be defined in accordance with [3, A.17]. The component design shall relate SW requirements to those components which contribute to the requirement's implementation (with precise detailing of the safety aspect) so that both the component requirements are defined clearly and requirements can be traced, verified and tested. The way the components are implemented shall be sketched (main internal data representations and, where applicable, algorithms). Each component specification shall be detailed so that it can be independently implemented, i.e., not requiring knowledge of the implementation of other components. This entails a definition of the overall and specific cooperation principles between components and their interfaces.

Req 4.4 [3, 7.3.4.25 to 7.4.3.28]: A modeling guideline for SCADE models shall be defined which transfers accepted criteria for SIL-4 coding [3, A.12] to the level of SCADE models with their LUSTRE semantics. A style for documentation shall be defined.

Req 4.5 [3, 7.3.4.29 to 7.4.3.32]: The “SW Integration Test Specification” shall employ Dynamic Analysis and Testing, Functional/Black-box Testing and Performance Testing [3, A.5,A.6]. *Rationale: This combination is approved for SIL 4. Performance testing, which is highly recommended, is included as it seems necessary.* A precise description of form and content of the test specification shall be given in the “Verification Plan”.

Req 4.6 [3, 7.3.4.33 to 7.3.4.39]: These requirements concern the SW/HW integration. Since no hardware is being developed, the SW/HW integration will be replaced by simulating the SW in an adequate environment, e.g., with simulated HW. This shall be done in the SW Validation Phase and addressed in the “Validation Plan”.

Req 4.7 [3, 7.4.4.7 to 7.4.4.10]: For each component, the “SW Component Test Specifications” shall employ Dynamic Analysis and Testing, Test Coverage for Code, Functional/Black-box Testing and Performance Testing [3, A.5]. A precise description of form and content of the test specifications shall be given in the “Verification Plan”.

Req 4.8 [3, 7.3.4.40 to 7.3.4.43, Sec. 6.1.4]: The verification of the architecture and design shall check that all SW requirements are allocated, all SW/HW interaction aspects are considered, that the architecture and design describe an adequate decomposition into subfunctions. It shall be verified that the aptness of the SCADE Suite in its use according to the coding principles and rules is sufficiently justified, and that the measures as a whole are in accordance with the EN 50128. It shall employ Static Analysis and Tracing to verify the suitability and completeness of the test specifications.

3.5 SW Component Implementation and Test

3.5.1 Objectives

This phase shall provide verified implementations of the components via modeling and code generation in the SCADE tool suite.

There may be additional code components developed by other means. The following text (Sec. 3.5.2 to 3.5.5) does not explicitly address these, but shall apply analogously to them.

3.5.2 Activities

The components are modeled in the SCADE Suite. Code is generated from the models with the SCADE Suite code generator. The code is verified as described in the component test specification. This includes the generation and execution of white-box tests according to the procedure defined in “SW Component Test Specification” to reach the required coverage. Black-box test cases are to be derived from “SW Component Test Specification” (addressing functionality and performance) and executed. The component design, implementation and test shall be independently verified.

3.5.3 Input

The main phase-specific inputs are:

1. “SW Architecture and Design Specification”

2. “SW Interface Specification”
3. “SW Component Test Specification”

3.5.4 Output

1. **SW Components (IMP)** The software components in the form of SCADE models and C-code generated from them with the SCADE KCG. The modules are tested for basic functionality and compatibility with interface requirements
2. **SW Component Test Report (TST)** Documentation of white-box and black-box tests according to the “SW Component Test Specification”, as performed by the Tester. Test environment and test cases are to be provided so that the tests can be rerun.
3. **SW Component Verification Report (VER)** Report on the verification of the results of this phase.

3.5.5 Requirements

Req 5.1 [3, 7.5.4.1 to 7.5.4.4]: The SCADE models shall have balanced size and complexity and be readable, understandable and testable. Details on how to achieve this are to be given in specific modeling guidelines. By testing, the Implementer shall assert that a component produced is mature enough for SW integration. *Rationale: It is more efficient to let the programmer perform basic tests than to involve a third party. This also offers the possibility to improve code quality by testing partially integrated SW, prior to rigorous verification.* the code to a maturity level

Req 5.2 [3, 6.1.4.5, 7.5.4.7]: The test report shall document all essential information about the test object, tester, results, coverage and evaluation. This report addresses the verification activities performed by the Tester. Details shall be defined in the verification plan.

Req 5.3 [3, 7.5.4.8 to 7.5.4.10]: The verification of the results (documented in the “SW Component Verification Report”) shall address the usage of the coding rules in the actual models, check structure, shape and documentation of the models and control that all required tests are documented.

3.6 SW Integration

3.6.1 Objectives

The integration shall result in a functional, integrated SW with correctly interacting components.

3.6.2 Activities

The SW components are integrated according to the integration strategy. Concrete test cases for integration tests are constructed as specified in the “SW Integration Test Specification” and applied to check the correct interplay of the components.

3.6.3 Input

1. “SW Integration Test Specification”
2. “SW Components”: the objects to be integrated.
3. “SW Component Test Specification”, “SW Component Test Report”: For use in constructing test cases for the integration tests and evaluating the results (reuse of unit tests).

3.6.4 Output

1. **SW Integration Test Report (INT)** Documentation of the integration test.
2. **SW Integration Verification Report (VER)** Report on the verification of the SW integration and test.

3.6.5 Requirements

Req 6.1 [3, 7.6.4.1, 7.4.6.6]: Necessary changes identified in the integration process have to be analyzed for their impact and trigger also re-verification. The tests shall be fully documented and be repeatable, and it shall be justified that an appropriate set of techniques and measures have been applied.[3]

Req 6.2 [3, 7.6.4.7 to 7.4.6.10]: These address the SW/HW integration which is not done in openETCS. Substitute activities via HW simulation are done in the subsequent phase.

Req 6.3 [3, 7.6.4.11 to 7.4.6.13]: The verification shall check that all test according to the “SW Integration Test Specification” have been documented in the test report, and that the report meets the general requirements on a test report and the specific ones ([3, 7.6.4.1, 7.4.6.6]).

3.7 SW Validation

3.7.1 Objectives

Validation of the functionality, safety and performance of the SW and its fitness for integration on a suitable HW.

3.7.2 Activities

The SW is tested against its requirements according to the “Overall SW Test Specification” and additional validation scenarios provided by the Validator. HW compatibility is checked via simulation, for which an environment is set up. The validation results are documented.

[3] Inconsistently, the EN 50128:2011 requires in 7.3.4.32 the “SW Integration Test Specification” to adhere to the requirements of Table A.5, and in 7.6.4.6 to check that the techniques and measures from Table A.6 have been correctly used. The procedure here follows Table A.5 which subsumes Table A.6.

3.7.3 Input

1. “Overall SW Test Specification”
2. “SW Components” (integrated)
3. “Validation Plan”
4. “SW Acceptance Plan”

3.7.4 Output

1. **Overall SW Test Report (TST)**: Report on the conformity test of the SW.
2. **SW Validation Report (VAL, VER)**: Report on the validation of the SW, including the validation of the tools used in the process. Verified by the Verifier.

3.7.5 Requirements

Req 7.1 [3, 7.7.4.1 to 7.7.4.4]: The “Overall SW Test Report” shall document

- the tests performed according to the “Overall SW Test Specification”,
- additional tests defined by the Validator, which may in particular address operational aspects or potentially occurring stress situations, and
- how the HW compatibility has been checked via simulation.

Req 7.2 [3, 7.7.4.6 to 7.7.4.11]: The “SW Validation Report” shall state that the development has been done in accordance with the “Validation Plan”, and that all verification activities have followed the “Verification Plan” and meet the requirements of the EN 50126-1:1999, resp., the EN 50128:2011. It shall also contain a substantiated statement that the overall combination of techniques and measures is adequate and in accordance with the standards, and that all tools are qualified to their purpose. Any discrepancies and deviations are to be documented and why they are acceptable. It shall contain a conclusion of the fitness of the SW for its purpose under a specified set of assumptions on the HW.

Req 7.3 [3, 6.3.4.13 to 6.3.4.14]: The “SW Validation Report” shall contain a report of the verification, by the Verifier, of the validation plan and the reported validation. The verification shall address general and specific requirements and in particular internal consistency. Verification of the validation plan shall precede the further validation activities.

4 Process Implementation

The process will not be implemented in a top-down, sequential style, where, e.g., specifications are completed and verified before the software is designed. Instead, the phases may overlap in time. This means, even software components may be designed before the SW specification is finalized. However, the final documentation of the development shall present a consistent, top-down view of the result. To ensure that this goal is achieved, specific considerations have to be given to the iterative aspect in the “Project Plan”, “QA-Plan” and in particular the “Configuration Management Plan” .

Ideally, a plan should be made which sets the goals to be achieved in the openETCS project. This consists in defining

- the functionality to be realized, and
- the completeness and maturity levels of the artifacts.

The maturity could, for instance, be defined in classes like “Draft”, “Revised”, “Final”. Completeness refers to functionality and other, artifact specific issues.

The plan should define a timeline which states which grade shall be achieved at which stage, and add corresponding activities.

Besides this general process flow, also the means, i.e., languages and tools, and their impact on the process should be addressed in a detailed description of the process implementation. In particular, the ramifications of the use of the SCADE tool suite deserve specific attention.

5 Glossary

API Application Programming Interface

EVC European Vital Computer

FME(C)A Failure Mode Effect (and Criticity) Analysis

FIS Functional Interface Specification

HW Hardware

I/O Input/Output

OBU On-Board Unit

openETCS activities the openETCS project and ensuing activities

openETCS project the current project (ITEA 2)

openETCS refers to the project and initiative, without a particular focus on either

PHA Preliminary Hazard Analysis

QA Quality Analysis

RBC Radio Block Center

RTM RunTime Model

Semi-formal language Language with a formal syntax (and a semantics which need not be strictly formal) necessarily

SIL Safety Integrity Level

SRS System Requirement Specification

SSHA Sub-System Hazard Analysis

SSRS Sub-System Requirement Specification

strictly-formal language Language with a formal syntax and a formal, mathematical semantics

Sub System This term denotes the EVC, whose software is the main focus of openETCS

SW Software

System This term denotes the ETCS system as described in Subset 026 [1]

Test Case Precise description of input/output to/from a test object, including a success criterion. Ready for execution but not necessarily already in machine readable form.

Test Specification A description of a set of test cases. Not all parameters need to be specified precisely.

THR Tolerable Hazard Rate

V&V Verification & Validation

Vital attribute of safety related items, e.g. artifacts

References

[1] UNISIG. SUBSET-026 - System Requirements Specification. Technical Report 3.3.0, ERA, March 2012.

[2] Marielle Petit-Doche and Matthias Güdemann. openETCS process. Technical Report D2.3, OpenETCS, June 2013.

[3] Railway applications – Communication, signalling and processing systems – software for railway control and protection systems. Norm EN 50128:2011, CENELEC, Brussels, Belgium, 2011.

[4] Railway applications – the specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS). Norm EN 50126-1:1999, CENELEC, Brussels, Belgium, 1999.

[5] Railway applications – communication, signalling and processing systems – safety related electronic systems for signalling. Norm EN 50129:2010, CENELEC, Brussels, Belgium, 2011.

[6] Sylvain Baro and Jan Welte. Requirements for openETCS. Technical Report D2.6, OpenETCS, June 2013.

[7] Michael Jastram and Marielle Petit-Doche. Report on the final choice of the primary toolchain. Technical Report 02, openETCS, November 2014.

[8] Hardi Hungar. openETCS valdiation & verification plan. Technical Report D4.1.1.02, openETCS, July 2014.

[9] Klaus-Rüdiger Hase and Peter Mahlmann. Project outline full project proposal annex openETCS. Technical Report v4.0, openETCS, 2014.

[10] UNISIG. SUBSET-034 3.0.0 - Train interface FIS. Technical Report 3.0.0, ERA.

[11] UNISIG. SUBSET-076 - Test related ERTMS documentation (this version is related to version 2.3.y of SUBSET-026). Technical Report 2.3.y, ERA.

[12] UNISIG. SUBSET-088 2.3.0 - ETCS Application Levels 1 & 2 - Safety Analysis. Technical Report 2.3.0, ERA.

[13] UNISIG. SUBSET-091 3.2.0 - Safety Requirements for the Technical Interoperability of ETCS in Levels 1 & 2. Technical Report 3.2.0, ERA.

[14] Comission Decision. CCS TSI for HS and CR transeuropean rail. Technical Report 2012/88/EU, EU, January 2012.