OCORA

Open CCS On-board Reference Architecture

Software Test and Integration Engineering according to EN 50657 or EN 50128

Document ID: OCORA-TWS09-030

Version: 1.00

Release: R1

Date: 18.11.2021

Management Summary

Considering integration & testing activities, this document is a complementary document to the TWS09-010-testing Strategy (Guideline).

This document is a contribution of DB Systemtechnik GmbH to the OCORA project. It is intended to support the elaboration of a process for the development of software for safety-related embedded systems up to SIL4 according to the standard DIN EN 50657:2017-11 (EN50657) resp. DIN EN 50128:2012-03 (EN50128).

This document covers a selection of requirements of the EN50657 resp. EN50128 standard for the testing and integration of software and applies in the OCORA project.

This document is valid in the context of the development of a test strategy in the OCORA project.

Revision history

Version	Change Description	Initial	Date of change
1.00	Official version for OCORA Release R1 *(based on DB document 1.0 27.10.2021)*	JL	18.11.2021

Table of contents

Table of figures

No table of figures entries found.

Table of tables

References

Reader’s note: please be aware that the numbers in square brackets, e.g. [1], as per the list of referenced documents below, is used throughout this document to indicate the references to external documents. Wherever a reference to a TSI-CCS SUBSET is used, the SUBSET is referenced directly (e.g. SUBSET-026). OCORA always reference to the latest available official version of the SUBSET, unless indicated differently.

[1] OCORA-BWS01-010 – Release Notes

[2] OCORA-BWS01-020 – Glossary

[3] OCORA-BWS01-030 – Question and Answers

[4] OCORA-BWS01-040 – Feedback Form

[5] OCORA-BWS03-010 – Introduction to OCORA

[6] OCORA-BWS04-010 – Problem Statements

[7] OCORA-TWS01-030 – System Architecture

[8] OCORA-BWS08-020 – Tooling

[9] OCORA-TWS06-020 – (Cyber-) Security – Guideline

[10] OCORA-TWS07-010 – Modular Safety – Strategy

[11] OCORA-TWS09-010 – Testing Strategy

1 Introduction

1.1 Purpose of the document

This document is a complementary document to the TWS09-010-testing Strategy (Guideline).

It is intended to support the elaboration of a process for the development of software for safety-related embedded systems up to SIL4 according to the standard DIN EN 50657:2017-11 (EN50657) resp. DIN EN 50128:2012-03 (EN50128).

This document covers a selection of requirements of the EN50657 resp. EN50128 standard for the testing and integration of software and applies in the OCORA project. The purpose of this document is to address various aspects of software testing according to EN50657 resp. EN50128. The document is not intended as a guide neither about software development nor about software testing.

This document is addressed to experts in the CCS domain and to any other person, interested in the OCORA concepts for on-board CCS. The reader will gain insights regarding the topics listed in chapter 1.1, and is invited to provide feedback to the OCORA collaboration and can, therefore, engage in shaping OCORA. Feedback to this document and to any other OCORA documentation can be given by using the feedback form [4].

If you are a railway undertaking, you may find useful information to compile tenders for OCORA compliant CCS building blocks, for tendering complete CCS system, or also for CCS replacements for functional upgrades or for life-cycle reasons.

If you are an organization interested in developing CCS building blocks according to the OCORA standard, information provided in this document can be used as input for your development.

Before reading this document, it is recommended to read the Release Notes [1]. If you are interested in the context and the motivation that drives OCORA we recommend to read the Introduction to OCORA [5], and the Problem Statements [6]. The reader should also be aware of the Glossary [2] and the Question and Answers [3].

1.2 Context of the document

This document is published as part of the OCORA release R1, together with the documents listed in the release notes [1]. Before reading this document, it is recommended to read the Release Notes [1]. If you are interested in the context and the motivation that drives OCORA we recommend to read the Introduction to OCORA [5], and the Problem Statements [6]. The reader should also be aware of the Glossary [2] and the Question and Answers [3].

1.3 Declaration

This document is exclusively for use within the framework of the OCORA project and presupposes the standard DIN EN 50657:2017-11 resp. DIN EN 50128:2012-03 also in the sense of copyright or author's right. Reproduction of any kind, including copying of parts, is not permitted.

In the standard DIN EN 50657:2017-11 resp. DIN EN 50128:2012-03, reference is made to the possibility that various aspects of the standard may affect patent rights. This also applies to this document. Therefore, no guarantee is given that the procedures etc. described in this document are free of third-party property rights.

The authors of the standard DIN EN 50657:2017-11 resp. DIN EN 50128:2012-03 or the responsible organisation declare that they are not responsible for the identification of corresponding patent rights. This also applies to this document.

The information contained in this document has been compiled to the best of our knowledge and with care. Nevertheless, errors cannot be completely ruled out. The information contained in this document does not claim to be complete. Therefore, the information contained in this document is not subject to any obligation or guarantee of any kind. The authors of this document accept no responsibility of any kind and assume no liability whatsoever arising in any way from the use of this information or parts thereof.

Should this document use common names, trade names, product designations, etc., this does not entitle the user to assume, without special identification, that such names are to be regarded as free within the meaning of the laws on trademarks and service marks and may therefore be used by anyone.

2 Definitions

References, names, terms and abbreviations as well as notations are to be used consistently in the context of software development according to the standard EN50657 resp. EN50128. These are defined within the scope of the following subchapters for the scope of this document.

Names, terms and abbreviations, unless specified in this document, are taken from EN50657 resp. EN50128 and IEC 60050-351:2013. If contradictions arise, the definitions from these standards shall apply.

2.1 Relevant Standards

Reference	Source
EN50126	EN 50126:2017 – Series; Railway Applications - The Specification and Demonstration of Reliability, Availability, Maintainability and Safety
EN50128	EN 50128:2011; Railway applications. Communication, signalling and processing systems. Software for railway control and protection systems
EN50657	EN 50657:2017; Railways Applications - Rolling stock applications - Software on Board Rolling Stock
IEC61160	IEC 61160:2005; Design review
IEC60050	IEC 60050-351:2013; International electrotechnical vocabulary - Part 351: Control technology
ISO/IEC250nn	ISO/IEC 250nn – Series; Systems and software engineering. Systems and software Quality Requirements and Evaluation
ISO/IEC9126	ISO/IEC 9126 – Series (withdrawn); Software Engineering – Product Quality

The EN50657 resp. EN50128 standards require verification, validation and assessment for the software development process. Checks are a central concept of the standard and must be integrated in various forms throughout the entire development process. Consistent application of the required procedures leads to effective development in the sense of the standard and also efficient development in the sense of project success.

2.2 Traceability

A fundamental prerequisite for checks is the identifiability and traceability of relevant elements and units of all abstraction levels in all phases of the development process. This starts with the classification and identification of relevant elements and units as well as their change and configuration management. This leads to the mapping of classifiable relationships between relevant elements and entities. These exist in the context of application development and variants, in the context of versioning as well as in the context of generic developments. Without the implementation of identifiability and traceability in the development process, the application of checks in the sense of the standard is hardly possible.

2.3 Verification

Verification determines the correctness of individual development steps. This involves assessing whether the task of the higher-level abstraction has been developed correctly and consistently. The evaluation of the results must be carried out phase by phase, up to the design of the software components. After implementation or code generation, the freedom from errors of the developed software must be demonstrated by testing the components, from step-by-step integrated software to the fully integrated system. Depending on the required safety integrity level, the methods to be used are specified by the EN50657 resp. EN50128 standard.

2.4 Testing

On the one hand, tests should confirm the correctness of the tested functions and properties, and on the other hand, possible errors should be detected. For this purpose, test cases are to be defined in such a way that as many errors as possible can be found and tests can be carried out as efficiently as possible. Depending on the required safety integrity level, the methods to be used are specified by the EN50657 resp. EN50128 standard.

2.5 Integration

The application-specific software is to be specified or designed according to the top-down principle and integrated according to the bottom-up principle. During integration, tested software components are to be combined step by step and systematically into larger units or with their target hardware and tested as a composite in each case. Finally, the embedded system is to be integrated into its target environment at common interfaces.

2.6 Validation

During validation, analysis and testing must be used to check whether the overall software has been correctly realised in accordance with the requirements specification. First, the results of the verifications as well as the component tests and integration must be checked. The conformity of the development results with the requirements specification must be traced and evaluated. The focus is on functions and properties according to the assigned safety integrity level. Depending on the required safety integrity level, the methods to be used are specified by the EN50657 resp. EN50128 standard. This includes additional tests that include complex scenarios for stressing the system and reflect the actual requirements of the application in the target environment.

3 Generic based test automatization

If a specific application is based on generic products resp. a generic application (GP/GA) according to EN50126, a realisation of the specific application according to chapter 8 of the standard EN50657 or EN50128 is possible. The hardware and software units of the GP/GA remain unchanged but are to be configured and parameterised according to specific requirements.

It is assumed that the proof of compatibility of units of the GP/GA has already been provided or can be provided based on conditions of use.

The specification and design are determined according to the top-down approach. However, based on the GP/GA, this should follow an architecture-based approach and be focused on the definition of necessary functions, units and parameters.

The definition of a configuration of software and hardware units is based on the requirements of the specific application as well as safety-relevant design principles.

The definition of the parameters depends on the requirements of the specific application as well as on the configuration.

The integration of the specific application takes place according to the bottom-up approach at the interfaces of the GP/GA as well as at the common interfaces with other subsystems of the superordinated system.

The integration of the specific application and the integration into the superordinated system depend on the requirements of the specific application and the resulting configuration and parameterisation.

When planning integration and testing, task assignments based on the OSI model and relevant (safety related) application conditions of the GP/GA must be considered.

When safeguarding parameters, both values and their relationships to each other must be considered. In addition, consistency and integrity of the corresponding data must be ensured.

If not already done as part of the GP/GA, performance and functional tests must be carried out.

Normative Requirements: In the context of a generation of specific application data and algorithms, the coherence and completeness of the data and algorithms with respect to the application principles as well as the specific architecture must be ensured (see EN50657 resp. EN50128, Chapter 8). In the context of the integration of a specific application and the acceptance tests, the concrete integration of the data or algorithms of the generic hardware and software (if required) as well as the complete plant must be ensured.

According to EN50657 resp. EN50128 Chapter 8 Paragraph 8.4.1.4 resp. Table A.11 **Functional Testing** (A.14) are to be applied. Requirements for application tests are to be considered according to Chapter 8.4.4 within the scope of the generation of application data and algorithms as well as Chapter 8.4.5 within the scope of the integration of an application and acceptance tests.

If combinations of data or algorithms have not been adequately checked within the GP/GA, these must be considered during verification, testing and validation of the specific application. This leads to the application of appropriate methods according to Chapter 7 of the EN50657 resp. EN50128 standard.

Test of the overall software resp. software/hardware integration shall be specified according to EN50657 resp. EN50128 Chapter 7 Paragraph 7.2.4.16 to .19 resp. 7.3.4.33 to .39 respectively. Please see also in [Chapter 4] of this document about the selection of methods according to EN50657 resp. EN50128. For the specification of the overall software tests, EN50657 resp. EN50128 Chapter 7.2.4.18 requires the use of methods from Table A.7. For the specification of the software/hardware integration, EN50657 resp. EN50128 Chapter 7.3.4.32 requires the application of methods from Table A.5. In each case, the approved combinations of techniques for safety integrity level 4 are specified there. Within the scope of the overall software test, the validator must define additional tests according to Chapter 7.7.4.3. These must stress the system with **complex scenarios** and reflect the actual requirements of the application.

Functional and Black-box Testing (D.14) or **Equivalence Classes and Input Partition Testing** (D.18) and **Boundary Value Analysis** (D.4) are required to specify the **Overall Software Tests** and **Software/Hardware Integration**.

For the specification of the **Overall Software Tests**, **Performance Testing** (A.18) resp. tests based on **Response Timing and Memory Constraints** (D.45), based on **Performance Requirements** (D.40), and **Avalanche-/Stress Testing** (D.3) are additionally required.

For the specification of the **Software/Hardware Integration**, additional **Structure-based Testing** (D.50) and tests based on **Performance Modelling** (D.39) are required in the context of **Dynamic Analysis and Testing** (A.13).

On the one hand, the validation tasks are based on Chapter 8.4.6 of the EN50657 resp. EN50128 standard. Accordingly, audits are to be performed in accordance with the validation plan. This includes the fulfilment of the requirements from Chapter 8.4.8 of the standard.

On the other hand, the tasks of validation are also based in particular on Chapter 8.4.8.6 of the standard. If data and algorithms have not been safeguarded within the scope of GP/GA, these must be considered accordingly during verification, testing and validation.

In the context of the specific application, the focus of the validation therefore lies in the area of the application tests or rather in the area of the overall software tests or the final validation according to Chapter 7.7. of EN50657 resp. EN50128.

Architecture-based approach: Based on an architectural model, it is possible to use assignments of interfaces, functions and parameters for the planning of Functional **and Black-box Testing** (D.14).

The detailing can be carried out, for example, by **Equivalence Classes and Input Partition Testing** (D.18) or **Boundary Value Analysis** (D.4) up to an automatic generation and documentation of the tests. Thus, vectors of input and expected values as well as test criteria and so on can be generated. These vectors, in turn, can be used to model sequences.

This concept can also be applied to **Structure-based Testing** (D.50), tests based on **Performance Requirements** (D.40) or **Performance Modelling** (D.39).

If corresponding structures and meta-/data are embedded in the architectural models of the generic products or applications and made available for the realization of specific applications, a standardization of some parts of test and integration can be achieved.

4 Methods

To process the required documents, this chapter selects relevant techniques and measures (methods) according to the EN50657 resp. EN50128 standard. First of all, the requirements from the standard Chapters 4.8 and 4.9 shall be fulfilled. In this context, reference is also made to Chapters 4.6 to 4.10 and the introductory section in Annex A of the standard about criteria for the selection of techniques and measures. The selection from the tables in Annex A of the standard can be made according to the following premises:

1) A method or a combination is requested in a table.

2) The definition of a method is mandatory or highly recommended.

Following the premises and the requirements of EN50657 resp. EN50128, the methods for SIL2 and SIL4 were defined for application based on the tables mentioned and the tables referenced in Annex A of the standard. Accordingly, the relevant chapters of the documents and the relevant tables or identifiers of the standard are assigned to the selected methods.

4.1 Overall Software Test

For the specification of tests of the overall software, Chapter 7.2.4.18 of EN50657 resp. EN50128 requires the use of methods from Table A.7. There, an approved combination of techniques is specified for SIL2, the application of which is in turn mandatory for SIL4.

Table 1 – Techniques & measures for software test

Document		**Compliance with the standard**	**Application**	
	Technique/Measure		**SIL2**	**SIL4**
Overall Software Test		**Chapter 7.2.4.18**		
	Functional and Black-box Testing (A.14)	Table A.7	yes	yes
	Equivalence Classes and Input Partition Testing (D.18)	Table A.14	yes	yes
	Boundary Value Analysis (D.4)	Table A.14	yes	yes
	Performance Testing (A.18)	Table A.7	yes	yes
	Response Timing and Memory Constraints (D.45)	Table A.18	yes	yes
	Avalanche-/Stress Testing (D.3)	Table A.18	no	yes
	Performance Requirements (D.40)	Table A.18	yes	yes
	Traceability (D.58)	Table A.9	yes	yes

4.2 Software/Hardware Integration

For the specification of the software/hardware integration, Chapter 7.3.4.39 of EN50657 resp. EN50128 requires the use of methods from Table A.5. There, approved combinations of techniques are specified for SIL2 and SIL4 respectively. A distinction is made as to which methods are to be assigned to verification or testing.

Table 2 – Techniques & measures for software integration

Document		**Compliance with the standard**	**Application**	
	Technique/Measure		**SIL2**	**SIL4**

Software/Hardware Integration		**Chapter 7.3.4.39**		
	Dynamic Analysis and Testing (A.13)	Table A.5	yes[*]	yes
	Equivalence Classes and Input Partition Testing (D.18)	Table A.13	yes[*]	yes
	Performance Modelling (D.39)	Table A.13	no	yes
	Structure-based Testing (D.50)	Table A.13	no	yes
	Test Case Execution from Boundary Value Analysis (D.4)	Table A.13	yes[*]	yes
	Functional and Black-box Testing (A.14)	Table A.5	yes[*]	yes
	Equivalence Classes and Input Partition Testing (D.18)	Table A.14	yes[*]	yes
	Boundary Value Analysis (D.4)	Table A.14	yes[*]	yes
	Traceability (D.58)	Table A.9	yes	yes
[*] The standard EN50657 or EN50128 leaves it up to the user whether Dynamic Analysis and Test or Functional and Black Box Tests are used. However, the techniques/measures are to be applied according to the specifications made here.				

4.3 Software Integration and Software Component Test

For the specification of software integration and component tests, Chapter 7.3.4.32 and Chapter 7.4.4.10 of EN50657 resp. EN50128 require the use of methods from Table A.5. There, approved combinations of techniques are specified for SIL2 and SIL4 respectively. A distinction is made as to which methods are to be assigned to verification or testing. In connection with structure-based tests, a quantified measure of test coverage is required.

Document		**Compliance with the standard**	**Application**	
	Technique/Measure		**SIL2**	**SIL4**
Software Integration and Software Component Test		**Chapter 7.3.4.32**		
	Dynamic Analysis and Testing (A.13)	Table A.5	yes[1)]	yes
	Equivalence Classes and Input Partition Testing (D.18)	Table A.13	yes[1)]	yes
	Performance Modelling (D.39)	Table A.13	no	yes
	Structure-based Testing (D.50)	Table A.13	yes[2)]	yes[2)]
	Test Coverage: Statement (D.50)	Table A.21	yes[2)]	no
	Test Coverage: Path (D.50)	Table A.21	no	yes[2)]
	Test Case Execution from Boundary Value Analysis (D.4)	Table A.13	yes[1)]	yes
	Functional and Black-box Testing (A.14)	Table A.5	yes[1)]	yes
	Equivalence Classes and Input Partition Testing (D.18)	Table A.14	yes[1)]	yes
	Boundary Value Analysis (D.4)	Table A.14	yes[1)]	yes
	Traceability (D.58)	Table A.9	yes	yes

Document		Compliance with the standard	Application	
	Technique/Measure		SIL2	SIL4
[1] The standard EN50657 or EN50128 leaves it up to the user whether Dynamic Analysis and Test or Functional and Black Box Tests are used. However, the techniques/measures are to be applied according to the specifications made here.				
[2] For code that cannot be tested appropriately, the EN50657 or EN50128 standard requires proof of correctness, for example by static analysis.				

4.4 Metrics

According to the standards ISO/IEC 9126 and ISO/IEC 250nn, metrics for the evaluation of software structures and software code can be distinguished. Within the scope of this document, the application of appropriate metrics shall be focused on safety-related functions. Metrics are mainly to be applied for the evaluation of design specifications as well as in the development of software components.

Based on the traceability of requirements, links with components, integration and testing of the entire software or the final validation are to be used to collect metrics.

5 Test and integration according to ISO 26262

Software for safety-related embedded systems to be installed in series production passenger cars with a maximum gross vehicle mass up to 3500 kg shall be developed according to the standard ISO 26262-6. Regarding generic software developed during the past, a part of the methodology of the ISO 26262-6:2011 (ISO26^3) will be discovered within this chapter and compared with the EN50657 focused on test and integration (see ISO26^3 Chapter 9 – Software unit tests and Chapter 10 – Software integration and testing). In addition, it is highly recommended to explore the latest version of the ISO 26262-6 too.

But first, it shall be recognised that the categories to support the selection of methods are different (see ISO26^3 Chapter 4.2 Interpretations of tables). And furthermore, there are no approved combinations of techniques recommended. So, it seems to be necessary, to compare the concrete application of methods used within the development of a generic product designed according to the ISO26^3 standard.

In case of **Software unit tests** as well as **Software** integration both, tests based on **Generation and analysis of equivalence classes** as well as **Analysis of boundary values** are a recommended option in case of ASIL B up to ASIL D.

In case of **Software unit tests** the **Statement, coverage** is a mandatory in case of ASIL A and a valid option in case of ASIL B. In case of ASIL B and ASIL D **Branch coverage** is a recommended option. In case of ASIL C **Branch coverage** is mandatory. In case of ASIL D **Modified Condition/Decision Coverage** is a valid option too.

Resource usage tests in case of ASIL D are stated and recommended as an option. Regarding to EN50657 SIL4 a kind of methods to found corresponding tests seems to be missing: Performance Analysis resp. Performance Modelling. Neither in ISO26^3-6 nor in ISO26^3-8 Chapter 12, about Qualification of software components, methods about Performance Analysis resp. Performance Modelling are stated. Regarding Software architectural design (see ISO26^3 Chapter 7) static as well as dynamic aspects are addressed including the description of timing behaviour. But not any specific method seems to be stated.

Based on the previous information, the compliance of ASIL B will be discussed. For this purpose, methods according to EN50657 and ISO26^3 are stated within [Table 3] to be compared accordingly. It came to a conclusion, that if software units were tested and integrated according to ISO26^3 ASIL B might also be recognised to comply with EN50657 SIL2 under the precondition, that another valid method was used then the Analysis of requirements.

Table 3 – IOS 26262 techniques & measures

Document		**Recommendation**	**Application**[*)]	
	Technique/Measure	**ASIL B**	**SIL2**	**SIL4**
Software Integration and Software Component Test				
	Dynamic Analysis and Testing (A.13)	---	yes[1)]	yes
	Equivalence Classes and Input Partition Testing (D.18)	++[3)]	yes[1)]	yes
	Performance Modelling (D.39)	---	no	yes
	Structure-based Testing (D.50)	---	yes[2)]	yes[2)]
	Test Coverage: Statement (D.50)	++[4)]	yes[2)]	no
	Test Coverage: Path (D.50)	++[4)]	no	yes[2)]
	Test Case Execution from Boundary Value Analysis (D.4)	++[3)]	yes[1)]	yes
	Functional and Black-box Testing (A.14)		yes[1)]	yes
	Equivalence Classes and Input Partition Testing (D.18)	++[3)]	yes[1)]	yes

Document		Recommendation	Application[*]	
	Technique/Measure	ASIL B	SIL2	SIL4
	Boundary Value Analysis (D.4)	++[3]	yes[1]	yes
	Analysis of requirements	++[3]	---	---
	Traceability (D.58)	---	yes	yes
[1] The standard EN50657 or EN50128 leaves it up to the user whether Dynamic Analysis and Test or Functional and Black Box Tests are used. However, the techniques/measures are to be applied according to the specifications made here.				
[2] For code that cannot be tested appropriately, the EN50657 or EN50128 standard requires proof of correctness, for example by static analysis.				
[3] The standard ISO26^3 leaves it up to the user whether Analysis of requirements, Equivalence Classes and Input Partition Testing or Test Case Execution from Boundary Value Analysis are used.				
[4] The standard ISO26^3 leaves it up to the user whether Statement or Path Coverage are used.				
[*] According to [Chapter 4.3]				