EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

DRAFT
EN 50128:2011

prAA

March 2019

ICS 35.240.60; 45.020; 93.100

English Version

Railway applications - Communication, signalling and processing systems - Software for railway control and protection systems

Applications ferroviaires - Systèmes de signalisation, de télécommunication et de traitement - Logiciels pour systèmes de commande et de protection ferroviaire

Bahnanwendungen - Telekommunikationstechnik, Signaltechnik und Datenverarbeitungssysteme - Software für Eisenbahnsteuerungs- und Überwachungssysteme

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

Project: 68080 Ref. No. EN 50128:2011/prAA:2019 E

Content

European foreword ... 3
1 General Changes ... 4
2 Modification to the Introduction ... 4
3 Modification to the Scope ... 4
4 Modification to Clause 2, Normative references ... 4
5 Modifications to 3.1, Terms and definitions ... 4
6 Modifications to Clause 4, Objectives, conformance and software safety integrity levels ... 6
7 Modifications to Clause 5, Software management and organization ... 6
8 Modifications to 6.2, Software verification ... 7
9 Modifications to 6.3, Software validation ... 7
10 Modifications to 6.4, Software assessment ... 7
11 Modifications to 6.5, Software quality assurance ... 7
12 Modifications to 6.7, Support tools and languages ... 8
13 Modifications to Clause 7, Generic software development ... 9
14 Modifications to Clause 8, Development of application data or algorithms: systems configured by application data or algorithms ... 9
15 Modifications to Clause 9, Software deployment and maintenance ... 10
16 Modifications to Annex A, Criteria for the Selection of Techniques and Measures ... 10
17 Modifications to Annex C ... 13

European foreword

This document (EN 50128:2011/prAA:2019) has been prepared by SC 9XA, “Communication, signaling and processing systems”, of Technical Committee CENELEC TC 9X, “Electrical and electronic applications for railways”.

This document is currently submitted to the Enquiry.

The following dates are proposed:

• latest date by which the existence of this document has to be announced at national level	(doa)	dor + 6 months
• latest date by which this document has to be implemented at national level by publication of an identical national standard or by endorsement	(dop)	dor + 12 months
• latest date by which the national standards conflicting with this document have to be withdrawn	(dow)	dor + 36 months (to be confirmed or modified when voting)

The EN 50128:2011 standard was amended to align with EN 50126-1:2017, EN 50126-2:2017 and EN 50129:2018. In addition, some technical mistakes were corrected and some clarifications were added.

This European Standard should be read in conjunction with EN 50126-1:2017 “*Railway applications – The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS) – Part 1: Generic RAMS Process”,* EN 50126-2:2017 *“Railway applications – The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS) – Part 2: Systems Approach to Safety*” and EN 50129:2018 “*Railway applications – Communication, signalling and processing systems – Safety related electronic systems for signalling*”.

1 General Changes

All occurrences of SIL 0 within EN 50128:2011 are replaced by with Basic integrity (EN 50126-1:2017, 3.7).

All occurrences of safety function(s) are replaced by safety-related function(s).

Use of the term “EN 50126-1” is replaced by “EN 50126-1 and EN 50129-2”.

The term “assessment” in the standard is meant as “independent safety assessment” as per definition of EN 50126-1:2017, 3.33.

All statements qualified by the words “software safety integrity level” are applicable also to Basic Integrity.

2 Modification to the Introduction

The following paragraph is added at the end of the Introduction:

This European Standard does not specify the requirements for the development, implementation, maintenance and/or operation of security policies or security services needed to meet security requirements that may be needed by the safety-related system. IT security can affect not only the operation but also the functional safety of a system. For IT security, appropriate IT security standards should be applied.

NOTE IEC/ISO standards that address IT security in depth are ISO 27000 series, ISO/IEC TR 19791 and the IEC 62443 series.

3 Modification to the Scope

The following subclause 1.10 is added:

1.10 For the development of User Programmable Integrated Circuits (e.g. FPGA and CPLD) guidance is provided in EN 50129:2018, Annex F.

4 Modification to Clause 2, Normative references

Replace the list of normative references by the following:

EN 50126-1:2017, Railway applications – The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS): Generic RAMS Process

EN 50126-2:2017, Railway applications – The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS): Systems Approach to Safety

EN 50129:2018, Railway applications – Communication, signalling and processing systems – Safety related electronic systems for signalling

EN ISO 9000, Quality management systems – Fundamentals and vocabulary

EN ISO 9001, Quality management systems – Requirements

ISO/IEC 90003, Software engineering – Guidelines for the application of ISO 9001 to computer software

ISO/IEC 9126 series, Software engineering – Product quality

5 Modifications to 3.1, Terms and definitions

Replace 3.1.9 (deleted) with:

3.1.51
error, < in software >
defect, mistake or inaccuracy in the development process which could result in a deviation from the intended performance or behaviour of the software

Note 1 to entry: definition is derived from EN 50126-1, 3.20 and adapted for software

3.1.52
fault
abnormal condition that could lead to an error in a system

Note 1 to entry: A fault for software is systematic

[SOURCE: IEC 60050-821:2017, 821-11-20, modified – The note 1 to entry has been modified.]

Replace 3.1.10 with:

3.1.10
failure, < of an item >
loss of ability to perform as required

Note 1 to entry: “Failure” is an event, as distinguished from “fault”, which is a state.

[SOURCE: IEC 60050-821:2017, 821-11-19, modified – The notes 1 and 2 have been omitted. A new note 1 to entry has been added.]

Replace 3.1.17 with:

3.1.17
pre-existing software
all software developed prior to the application currently in question is classed as pre-existing software

Note 1 to entry: That includes commercial off-the-shelf software, open-source software and software previously developed but not in accordance with this European Standard.

[SOURCE: EN 50126-1:2017, 3.43, modified – The end of the definition has been moved to the note 1 to entry.]

Definition 3.1.26 replaced by:

3.1.26
risk, < for railway RAMS >
combination of expected frequency of loss and the expected degree of severity of that loss

[SOURCE: EN 50126-1:2017, 3.57]

Definition 3.1.27 replaced by:

3.1.27
safety
freedom from unacceptable risk

[SOURCE: IEC 60050-903:2013, 903-01-19]

Definition 3.1.28 replaced by:

3.1.28
safety authority
body responsible for delivering the authorization for the operation of the safety-related system

[SOURCE: IEC 60050-821:2017, 821-12-52]

Remove the term 3.1.29 and its definition (see also General Changes).

Definition 3.1.30 replaced by:

3.1.30
safety-related software
software which performs safety-related functions

Note 1 to entry: software is called safety-related if at least one of its properties is used in the safety argument for the system in which it is applied. These properties can be of functional or non-functional nature.

[SOURCE: IEC 60050-821:2017, 821-12-60, modified – “safety functions” has been replaced with “safety-related functions”. The note 1 to entry has been added.]

Definition 3.1.46 replaced by:

3.1.46
validation
confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled

Note 1 to entry: The term “validated” is used to designate the corresponding status.

Note 2 to entry: The use conditions for validation can be real or simulated.

Note 3 to entry: In design and development, validation concerns the process of examining an item to determine conformity with user needs.

Note 4 to entry: Validation is normally performed during the final stage of development, under defined operating conditions, although it can also be performed in earlier stages.

Note 5 to entry: Multiple validations can be carried out if there are different intended uses.

[SOURCE: IEC 60050-192:2015, 192-01-18]

Definition 3.1.48 replaced by:

3.1.48
verification
confirmation, through the provision of objective evidence, that specified requirements have been fulfilled

Note 1 to entry: The term “verified” is used to designate the corresponding status.

Note 2 to entry: Design verification is the application of tests and appraisals to assess conformity of a design to the specified requirement.

Note 3 to entry: Verification is conducted at various life cycle phases of development, examining the system and its constituents to determine conformity to the requirements specified at the beginning of that life cycle phase.

[SOURCE: IEC 60050-192:2015, 192-01-17, modified – The note 3 to entry has been modified.]

Add the following 3.1.50 (in line with EN50126-1):

3.1.50
safety-related
carries responsibility for safety

[SOURCE: IEC 60050-821:2017, 821-01-73]

6 Modifications to Clause 4, Objectives, conformance and software safety integrity levels

4.4 is replaced by:

4.4 At least the basic integrity requirements of this European Standard shall be fulfilled for the software part of functions that have a safety impact below SIL 1.

NOTE Basic integrity requirements can also be used for development of non safety-related software.

7 Modifications to Clause 5, Software management and organization

In 5.1.2.10 bullet n) replace as follows:

n) A person who is Validator may also perform the role of Verifier, but still maintaining independence from the Project Manager. In this case, as for all other development activities, the Validator/Verifier outputs shall be reviewed by another competent person.

In 5.1.2.11 bullet m) replace as follows:

m) A person who is Validator may also perform the role of Verifier, Integrator and Tester. In this case, as for all other development activities, the Validator/Verifier outputs shall be reviewed by another competent person.

8 Modifications to 6.2, Software verification

In 6.2.3, Output documents, Bullet 3) is replaced by:

3) Software Planning Verification Report

In 6.2.4, Requirements, 6.2.4.10 is replaced by:

6.2.4.10 A Software Planning Verification Report shall be written, under the responsibility of the Verifier, on the basis of the input documents from 6.2.2. The Software Planning Verification Report shall be reviewed by the Validator.

The requirement in 6.2.4.11 refers to the Software Planning Verification Report.

6.2.4.11 is replaced by:

6.2.4.11 Once the software plans have been established (Software Quality Assurance Plan, Software Configuration Management Plan, Software Verification Plan, Software Validation Plan, and Software Maintenance Plan) verification shall address

a) that the software plans meet the general requirements for readability and traceability in 5.3.2.7 to 5.3.2.10 and in 6.5.4.14 to 6.5.4.17 as well as the specific requirements in 6.2.4.3 to 6.2.4.9,

b) the internal consistency of the software plans,

c) the coherency of the software plans.

The results shall be recorded in a Software Planning Verification Report.

9 Modifications to 6.3, Software validation

In 6.3.3, Output documents:

Remove 3) Software Validation Verification Report

Remove 6.3.4.12, 6.3.4.13 and 6.3.4.14

10 Modifications to 6.4, Software assessment

In 6.4.3, Output documents:

Remove 3) Software Assessment Verification Report

Remove 6.4.4.6 and 6.4.4.7

11 Modifications to 6.5, Software quality assurance

In 6.5.3, Output documents:

Remove 3) Software Quality Assurance Verification Report

In 6.5.4, Requirements:

6.5.4.3 is replaced by:

6.5.4.3 A Software Quality Assurance Plan shall be written on the basis of the input documents from 6.5.2.

Remove 6.5.4.7 and 6.5.4.8.

12 Modifications to 6.7, Support tools and languages

6.7.4.4 is replaced (based on EN 50129:2018, 6.3) by:

6.7.4.4 For each tool in class t3, evidence shall be available that the output of the tool conforms to the specification of the output or failures in the output are detected. Evidence may be based on the same steps necessary for a manual process as a replacement for the tool and an argument presented if these steps are replaced by alternatives (e.g. validation of the tool). Evidence shall be based on one or more of the following techniques.

a) a suitable combination of history of successful use, for which the following conditions apply:

- It has been confirmed and periodically checked by analysis, examination and provision of objective evidence that the tool is suitable for the purpose for which it is to be used.
- There is sufficient qualitative evidence from previous use to show that the output of the tool can be trusted, including a suitable combination of documented history of successful use (project and applications realized, list of anomalies, identification of successive versions).

b) tool validation as specified in 6.7.4.5,

c) tool diversity, which is typically obtained by using one of the possibilities below:

- Use of two diverse tools to perform the same function, with comparison of their outputs.
- Use one tool to perform the function and feed its output into a diverse tool which regenerates the input data provided to the first tool, with comparison of the original and regenerated input data.
- Use one tool to perform the function and subject its output to verification of results by an independent tool, which is diverse from the first tool because it works in accordance with different principles, for example a rule-checking tool to confirm that the output conforms to all relevant rules (e.g. rules for the positioning of balises).

d) compliance with the safety integrity levels derived from the risk analysis of the process and procedures including the tools,

e) other appropriate methods for avoiding or handling failures introduced by tools (e.g. verification of tool output).

NOTE 1 As an example the use of a non-trusted compiler can be justified as follows.

The object code produced by the compiler has been subjected to a combination of tests, checks and analyses which are capable of ensuring the correctness of the code to the extent that it is consistent with the target Safety Integrity Level. In particular, the following applies to all tests, checks and analyses.

- Testing has been shown to have a sufficiently high coverage of the implemented code. If there is any code unreachable by testing, it has been shown by checks or analyses that the function concerned is executed correctly when the code is reached on the target.
- Checks and analyses have been applied to the object code and shown to be capable of detecting the types of errors which might result from a defect in the compiler.
- No more translation with the compiler has taken place after testing, checking and analysis.
- If further compilation or translation is carried out, all tests, checks and analyses will be repeated.

NOTE 2 The evidence listed for T3 can also be used for T2 tools in judging the correctness of their results.

Remove 6.7.4.6 (now redundant with replaced 6.7.4.4).

13 Modifications to Clause 7, Generic software development

7.3.4.32 is replaced by:

7.3.4.32 The software integration test specification shall choose techniques and measures from Table A.6. the selected combination shall be justified as a set satisfying 4.8 and 4.9.

7.3.4.39 is replaced by:

7.3.4.39 The software/hardware integration test specification shall choose techniques and measures from Table A.6. The selected combination shall be justified as a set satisfying 4.8 and 4.9.

14 Modifications to Clause 8, Development of application data or algorithms: systems configured by application data or algorithms

Add the following NOTE in 8.1.1.

NOTE Complying with Clauses 1–7 and 9 ensures that development of application algorithms also complies with 8.4.1 to 8.4.7.

8.4.1.2 is replaced by:

8.4.1.2 An Application Preparation Plan shall be produced in order to define and detail the application development process, including all the activities, deliverables and roles in charge of them.

NOTE The Application Preparation Plan contains the definition of an application development process that can be re-used for a class of specific applications (i.e. for a generic application).

8.4.1.12 is replaced by:

8.4.1.12 An application preparation verification report shall be written, under the responsibility of the verifier, on the basis of the input documents from 8.2.

The requirement in 8.4.1.13 refers to the Application Preparation Verification Report.

8.4.1.13 is replaced by:

8.4.1.13 Once the application preparation plan has been established, verification shall address

a) that the Application Preparation Plan meets the general requirements for readability and traceability in 5.3.2.7 to 5.3.2.10 and in 6.5.4.14 to 6.5.4.17 as well as the specific requirements in 8.4.1.2 to 8.4.1.11,

b) the internal consistency of the Application Preparation Plan.

The results shall be recorded in the Application Preparation Verification Report.

Replace contents of 8.4.3 with:

8.4.3.1 The quantity and type of the generic hardware and software components to be used in the specific application shall be specified. The location of components, application data and algorithms in the specific application architecture shall be defined. The application data and algorithms processed by the generic software shall be designed at this stage.

The results of these activities shall be recorded in the output document Application Architecture and Design.

Replace contents of 8.4.4.4 with:

8.4.4.4 The application data/algorithms verification report shall

a) document every activity performed to ensure correctness and completeness of data/algorithm and their coherency with application principles and specific application architecture,

b) evaluate compatibility of data/algorithms with conditions from the generic application.

Replace contents of 8.4.5.1 with:

8.4.5.1 For some systems the application data/algorithms can be integrated with the generic hardware and software for a factory test before installation on the target system. This may not be necessary where a sufficient degree of confidence can be obtained by other means. The application shall then be installed on the target system, and integration tests within the complete installation shall be carried out. Finally the target system shall be commissioned as a fully operational system, and a final acceptance process of the target system in the complete installation shall be carried out. the application test report shall document the correct and complete execution of tests defined in the application test specification. The application data/algorithms verification report shall check the completeness and correctness of tests performed on the complete installation.

Replace contents of 8.4.7.6 with:

8.4.7.6 The application preparation verification report shall demonstrate the coverage and enforcement of the application conditions of the generic software and application tools.

15 Modifications to Clause 9, Software deployment and maintenance

9.2.1.1 is replaced by:

9.2.1.1 To ensure that the software performs as required, preserving the required software safety integrity level and dependability when making corrections, enhancements or adaptations to the software itself. See also 7.12 “phase 11: operation, maintenance and performance monitoring” in EN 50126-1 and EN 50129-2.

9.2.4.8 is replaced by:

9.2.4.8 A software maintenance record shall be established for each software item before its first release, and it shall be maintained. In addition to the recommendations of ISO/IEC 90003:2014 for “maintenance”, this record shall also include

a) references to all the Software Change Records for that software item,

b) change impact assessment,

c) test cases for components, including revalidation and regression testing data, and

d) software configuration history.

16 Modifications to Annex A, Criteria for the Selection of Techniques and Measures

Add the following NOTE [extracted from EN 50657]:

NOTE 1: The following examples provide further guidance on the application of the clause tables (see A.1) and the detailed tables (see A.2).

- Table A.7 defines HR for “Functional and Black-box Testing” for Basic Integrity and refers to Table A.14. But A.14 only contains “R” methods. This means it is highly recommended to apply functional and black-box testing but the user of the standard is completely free to apply the techniques from A.14 or any other technique.

- Table A.5 has no requirement for “Static Analysis” for Basic Integrity and refers to Table A.19. But A.19 contains a “HR” for “Walkthroughs/Design Reviews, This means it is not required to do static analysis but if the user of the standard nevertheless decides to do so, then the HR in Table A.19 applies.

- Table A.7 defines “M” for Performance Testing for SIL3/4 and refers to Table A.18. But A.18 contains only “HR” methods. This means it is mandatory to do performance testing but it is allowed to apply a technique that is not mentioned in A.18, if the rationale for using alternative techniques can be given.

Replace Table A.1 by the following (re-used from EN 50657 and adapted as per SGA18 Report (see SC9XA/Sec1066/DS)).

Table A.1 – Lifecycle Issues and Documentation (5.3)

DOCUMENTATION	Basic Integrity	SIL 1	SIL 2	SIL 3	SIL 4
Planning					
1. Software Quality Assurance Plan	HR	HR	HR	HR	HR
2. Software Planning Verification Report	R	HR	HR	HR	HR
3. Software Configuration Management Plan	HR	HR	HR	HR	HR
4. Software Verification Plan	HR	HR	HR	HR	HR
5. Software Validation Plan	HR	HR	HR	HR	HR
Software requirements					
6. Software Requirements Specification	HR	HR	HR	HR	HR
7. Overall Software Test Specification	HR	HR	HR	HR	HR
8. Software Requirements Verification Report	R	HR	HR	HR	HR
Architecture and design					
9. Software Architecture Specification	R	HR	HR	HR	HR
10. Software Design Specification	R	HR	HR	HR	HR
11. Software Interface Specifications	HR	HR	HR	HR	HR
12. Software Integration Test Specification	R	HR	HR	HR	HR
13. Software/Hardware Integration Test Specification	R	HR	HR	HR	HR
14. Software Architecture and Design Verification Report	R	HR	HR	HR	HR
Component Design					
15. Software Component Design Specification	-	HR	HR	HR	HR
16. Software Component Test Specification	-	HR	HR	HR	HR
17. Software Component Design Verification Report	-	HR	HR	HR	HR
Component Implementation and Testing					
18. Software Source Code and supporting documentation	HR	HR	HR	HR	HR
19. Software Component Test Report	-	HR	HR	HR	HR
20. Software Source Code Verification Report	-	HR	HR	HR	HR
Integration					
21. Software Integration Test Report	R	HR	HR	HR	HR
22. Software/Hardware Integration Test Report	R	HR	HR	HR	HR
23. Software Integration Verification Report	R	HR	HR	HR	HR
Overall Software Testing / Final Validation					
24. Overall Software Test Report	HR	HR	HR	HR	HR
25. Software Validation Report	HR	HR	HR	HR	HR
26. Tools Validation Report	-	HR	HR	HR	HR
27. Release Note	HR	HR	HR	HR	HR
Systems configured by application data					
28. Application Requirements Specification	HR	HR	HR	HR	HR
29. Application Preparation Plan (see NOTE 4)	R	HR	HR	HR	HR

DOCUMENTATION	Basic Integrity	SIL 1	SIL 2	SIL 3	SIL 4
30. Application Test Specification (see NOTE 4)	HR	HR	HR	HR	HR
31. Application Architecture and Design (see NOTE 4)	R	HR	HR	HR	HR
32. Application Preparation Verification Report	-	HR	HR	HR	HR
33. Application Test Report	HR	HR	HR	HR	HR
34. Source Code of Application Data/Algorithms	HR	HR	HR	HR	HR
35. Application Data/Algorithms Verification Report	HR	HR	HR	HR	HR
Software deployment					
36. Software Release and Deployment Plan	R	HR	HR	HR	HR
37. Software Deployment Manual	R	HR	HR	HR	HR
38. Release Notes	HR	HR	HR	HR	HR
39. Deployment Records	R	HR	HR	HR	HR
40. Deployment Verification Report	R	HR	HR	HR	HR
Software maintenance					
41. Software Maintenance Plan	R	HR	HR	HR	HR
42. Software Change Records	HR	HR	HR	HR	HR
43. Software Maintenance Records	R	HR	HR	HR	HR
44. Software Maintenance Verification Report	R	HR	HR	HR	HR
Software assessment					
45. Software Assessment Plan	-	HR	HR	HR	HR
46. Software Assessment Report	-	HR	HR	HR	HR

NOTE 1 For basic integrity, the content of the software verification plan can be limited to 6.2.4.9 a), b), c), d), f).

NOTE 2 For basic integrity the software interface specification is only highly recommended for the boundary of the overall software (see also 7.3.4.18).

NOTE 3 According to 5.3.2.12 and 5.3.2.13, documents can be combined differently.

NOTE 4 Documents 29, 30 and 31 being HR or R depends on the importance defined in the process and where the verification takes place. E.g. data can only be needed to be verified but tested in the system domain while more functional properties need both test and verification. In this case HR has been marked but can be optional R.

NOTE 5 Software Architecture Specification and Software Design Specification for Basic Integrity is important to establish especially for long-term maintenance, selection can consider this (e.g. adding these information within Software Requirements specification)

Within the tables from A.2 to A.22 replace SIL 0 with basic integrity.

Header of Table A.4 replaced by:

Table A.4– Software Design and Implementation (7.3, 7.4, and 7.5)

Header of Table A.5 replaced by:

Table A.5 – Verification and Testing (6.2 and 7.4)

Header of Table A.6 replaced by:

Table A.6 – Integration (7.3 and 7.6)

Header of Table A.7 replaced by:

Table A.7 – Overall Software Testing (6.2 and 7.2)

Remove technique "Limited number of subroutine parameters" in Table A.12

17 Modifications to Annex C

Replace Table C.1 / Documentation / entry 2 *by*: "Software Planning Verification Report".