

<b>STN</b>	<b>Dráhové aplikácie Požiadavky na vývoj softvéru</b>	<b>STN EN 50716</b>  34 2680
------------	---	--

Railway Applications - Requirements for software development

Táto norma obsahuje anglickú verziu európskej normy.  
This standard includes the English version of the European Standard.

Táto norma bola oznámená vo Vestníku ÚNMS SR č. 01/24

Obsahuje: EN 50716:2023

Oznámením tejto normy sa od 30.10.2026 ruší  
STN EN 50128 (34 2680) z novembra 2012

STN EN 50657 (34 1518) z mája 2018

**138008**

---

Úrad pre normalizáciu, metrológiu a skúšobníctvo Slovenskej republiky, 2024  
Slovenská technická norma a technická normalizačná informácia je chránená zákonom č. 60/2018 Z. z. o technickej normalizácii  
v znení neskorších predpisov.

EUROPEAN STANDARD

**EN 50716**

NORME EUROPÉENNE

EUROPÄISCHE NORM

November 2023

ICS 35.240.60

Supersedes EN 50128:2011; EN 50128:2011/AC:2014;  
EN 50657:2017; EN 50128:2011/A1:2020; EN  
50128:2011/A2:2020; EN 50657:2017/A1:2023

English Version

## Railway Applications - Requirements for software development

Applications ferroviaires - Exigences pour le développement  
de logiciels

Sektorübergreifende Software-Norm für Eisenbahnen

This European Standard was approved by CENELEC on 2023-10-30. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Türkiye and the United Kingdom.



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**

**EN 50716:2023 (E)**

<b>Contents</b>		<b>Page</b>
European foreword.....		6
Introduction.....		8
1	Scope .....	11
2	Normative references .....	11
3	Terms, definitions and abbreviations .....	12
3.1	Terms and definitions .....	12
3.2	Abbreviations.....	17
4	Software integrity levels conformance .....	18
5	Software management and organization .....	19
5.1	Organization and independence of roles .....	19
5.1.1	Objective .....	19
5.1.2	Requirements .....	19
5.2	Personnel competence and responsibilities.....	22
5.2.1	Objectives.....	22
5.2.2	Requirements.....	22
5.3	Lifecycle issues and documentation .....	23
5.3.1	Objectives.....	23
5.3.2	Requirements .....	23
6	Software assurance .....	24
6.1	Software testing .....	24
6.1.1	Objective .....	24
6.1.2	Input documents.....	24
6.1.3	Output documents.....	24
6.1.4	Requirements .....	24
6.2	Software verification .....	25
6.2.1	Objective .....	25
6.2.2	Input documents.....	25
6.2.3	Output documents .....	25
6.2.4	Requirements .....	26
6.3	Software validation.....	27
6.3.1	Objective .....	27
6.3.2	Input documents.....	27
6.3.3	Output documents .....	27
6.3.4	Requirements .....	27
6.4	Software assessment.....	28
6.4.1	Objective .....	28

**EN 50716:2023 (E)**

6.4.2	Input documents.....	29
6.4.3	Output documents.....	29
6.4.4	Requirements.....	29
6.5	Software quality assurance.....	30
6.5.1	Objectives.....	30
6.5.2	Input documents.....	30
6.5.3	Output documents.....	30
6.5.4	Requirements.....	30
6.6	Modification and change control.....	33
6.6.1	Objectives.....	33
6.6.2	Input documents.....	33
6.6.3	Output documents.....	33
6.6.4	Requirements.....	33
6.7	Support tools and languages.....	34
6.7.1	Objectives.....	34
6.7.2	Input documents.....	34
6.7.3	Output documents.....	34
6.7.4	Requirements.....	34
7	Software development.....	37
7.1	Lifecycle and documentation for software.....	37
7.1.1	Objectives.....	37
7.1.2	Requirements.....	37
7.2	Software requirements.....	37
7.2.1	Objectives.....	37
7.2.2	Input documents.....	37
7.2.3	Output documents.....	37
7.2.4	Requirements.....	37
7.3	Architecture and design.....	39
7.3.1	Objectives.....	39
7.3.2	Input documents.....	40
7.3.3	Output documents.....	40
7.3.4	Requirements.....	40
7.4	Component design.....	47
7.4.1	Objectives.....	47
7.4.2	Input documents.....	47
7.4.3	Output documents.....	47
7.4.4	Requirements.....	47
7.5	Component implementation and testing.....	49
7.5.1	Objectives.....	49
7.5.2	Input documents.....	49
7.5.3	Output documents.....	49

**EN 50716:2023 (E)**

7.5.4	Requirements .....	49
7.6	Integration .....	50
7.6.1	Objectives .....	50
7.6.2	Input documents .....	50
7.6.3	Output documents .....	50
7.6.4	Requirements .....	50
7.7	Overall software testing / Final validation .....	52
7.7.1	Objectives .....	52
7.7.2	Input documents .....	52
7.7.3	Output documents .....	52
7.7.4	Requirements .....	52
8	Development of application data: systems configured by application data .....	54
8.1	Objectives .....	54
8.2	Input documents .....	54
8.3	Output documents .....	55
8.4	Requirements .....	55
8.4.1	Application development process .....	55
8.4.2	Application requirements specification .....	56
8.4.3	Architecture and Design .....	57
8.4.4	Application data production .....	57
8.4.5	Application integration and testing .....	58
8.4.6	Application validation and assessment .....	58
8.4.7	Application preparation procedures and tools .....	58
9	Software deployment and maintenance .....	59
9.1	Software deployment .....	59
9.1.1	Objective .....	59
9.1.2	Input documents .....	59
9.1.3	Output documents .....	59
9.1.4	Requirements .....	59
9.2	Software maintenance .....	61
9.2.1	Objective .....	61
9.2.2	Input documents .....	61
9.2.3	Output documents .....	61
9.2.4	Requirements .....	61
Annex A (normative)	Criteria for the selection of techniques and measures .....	64
Annex B (normative)	Key software roles and responsibilities .....	75
Annex C (informative)	Guidance on software development .....	81
Annex D (informative)	Bibliography of techniques .....	94
Annex ZZ (informative)	Relationship between this European Standard and the Essential Requirements of EU Directive (EU) 2016/797 aimed to be covered .....	122
Bibliography	.....	124

**List of figures**

Figure 1 — Illustrative Software Route Map .....	10
Figure 2 — Illustration of the organizational structure .....	21
Figure C.1 — Linear lifecycle model example 1 (Waterfall).....	82
Figure C.2 — Linear lifecycle model example 2 (V model).....	83
Figure C.3 — Iterative lifecycle model example 1 .....	84
Figure C.4 — Iterative lifecycle model example 2 .....	85
Figure C.5 — Example for iterative development of a work product .....	86
Figure C.6 — Iterative lifecycle model example 3 .....	86

**List of tables**

Table 1 — Relation between tool class and applicable subclauses .....	36
Table A.1 — Lifecycle issues and documentation (5.3).....	65
Table A.2 — Software Requirements Specification (7.2) .....	67
Table A.3 — Software architecture (7.3).....	67
Table A.4 — Software design and implementation (7.3, 7.4 and 7.5) .....	68
Table A.5 — Software component analysis and testing (6.2 and 7.4).....	69
Table A.6 — Software integration analysis and testing (7.3 and 7.6).....	69
Table A.7 — Overall software analysis and testing (6.2 and 7.2).....	69
Table A.8 — Intentionally left blank .....	69
Table A.9 — Software quality assurance (6.5) .....	70
Table A.10 — Software maintenance (9.2).....	70
Table A.11 — Data preparation techniques (8.4) .....	70
Table A.12 — Coding standards .....	71
Table A.13 — Dynamic analysis and testing .....	71
Table A.14 — Intentionally left blank .....	71
Table A.15 — Suitable Programming Languages .....	72
Table A.16 — Intentionally left blank .....	72
Table A.17 — Modelling .....	73
Table A.18 — Performance testing .....	73
Table A.19 — Static analysis .....	73
Table A.20 — Components .....	74
Table A.21 — Test coverage for code .....	74
Table B.1 — Requirements Manager role specification .....	75
Table B.2 — Designer role specification.....	76
Table B.3 — Implementer role specification .....	76
Table B.4 — Tester role specification .....	77
Table B.5 — Verifier role specification .....	77
Table B.6 — Validator role specification .....	78
Table B.7 — Assessor role specification .....	79
Table B.8 — Project Manager role specification.....	80
Table B.9 — Configuration Manager role specification .....	80
Table C.1 — Architecture and design typical adaptation for modelling .....	90
Table C.2 — Component implementation and testing typical adaptation for modelling .....	91
Table C.3 — Coding standards techniques / measures typical adaptation for modelling .....	91
Table ZZ.1 — Correspondence between this European Standard, Commission Regulation 2016/919 concerning the technical specification for interoperability relating to the 'control-command and signalling' subsystems of the rail system in the European Union* and Directive (EU) 2016/797.....	122
Table ZZ.2 — Correspondence between this European Standard, Commission Regulation (EU) N° 1302/2014 concerning the technical specification for interoperability relating to the 'rolling stock — locomotives and passenger rolling stock' subsystem of the rail system in the European Union* and Directive (EU) 2016/797 .....	123

**EN 50716:2023 (E)****European foreword**

This document (EN 50716:2023) has been prepared by CLC/TC 9X “Electrical and electronic applications for railways”.

The following dates are fixed:

- latest date by which this document has to be (dop) 2024-10-30 implemented at national level by publication of an identical national standard or by endorsement
- latest date by which the national standards (dow) 2026-10-30 conflicting with this document have to be withdrawn

This document supersedes EN 50128:2011 and EN 50657:2017 and all of their amendments and corrigenda (if any).

EN 50716:2023 includes the following significant technical changes with respect to EN 50128:2011 and EN 50657:2017:

- Better alignment with EN 50126-1:2017 and EN 50126-2:2017, including definitions, has been made;
- Clause 5: requirements have been re-written to simplify readability (while keeping existing options for organization broadly unchanged);
- Annex A has been updated to have a better alignment with lifecycle phases;
- In informative Annex C, new clause C.1 has been added with additional guidance on lifecycle models;
- In informative Annex C, new clause C.2 has been added with guidance on modelling for software development;
- Additional guidance provided for software components of different software integrity levels;
- Requirements on programming languages have been generalized.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CENELEC shall not be held responsible for identifying any or all such patent rights;

This document is read in conjunction with EN 50126-1 “*Railway applications – The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS): Basic requirements and generic process*” [1] and EN 50126-2 “*Railway applications – The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS): Systems Approach to Safety*” [2].

For railway related fixed installations (electric traction power control and supply) EN 50562 “*Railway applications - Fixed installations - Process, protective measures and demonstration of safety for electric traction systems*” [20] is applicable.

This document has been prepared under a standardization request addressed to CENELEC by the European Commission. The Standing Committee of the EFTA States subsequently approves these requests for its Member States.

**EN 50716:2023 (E)**

For the relationship with EU Legislation, see informative Annex ZZ, which is an integral part of this document.

Any feedback and questions on this document should be directed to the users' national committee. A complete listing of these bodies can be found on the CENELEC website.



## EN 50716:2023 (E)

### Introduction

This document concentrates on the methods which need to be used in order to provide software which meets the demands for software integrity.

This document provides a set of requirements for the development, deployment and maintenance of any software intended for railway applications. It defines requirements concerning organizational structure, the relationship between organizations and division of responsibility involved in the development, deployment and maintenance activities. Criteria for the qualification and expertise of personnel are also provided in this document.

The key concept of this document is that of levels of software integrity. This document addresses five software integrity levels where Basic Integrity is the lowest and 4 the highest one. The higher the risk resulting from software failure, the higher the software integrity level will be.

NOTE 1 The concept of Basic Integrity used in this document was first introduced in the EN 50126 series ([1] [2]).

This document identifies techniques and measures for the five levels of software integrity. The required techniques and measures for Basic Integrity and for the safety integrity levels 1 to 4 are shown in the normative tables of Annex A. The required techniques for level 1 are the same as for level 2, and the required techniques for level 3 are the same as for level 4. This document does not give guidance on which level of software integrity is appropriate for a given risk. This decision will depend upon many factors including the nature of the application, the extent to which other systems carry out safety-related functions and social and economic factors.

It is within the scope of EN 50126-1 and EN 50126-2 to define the process of specifying the safety-related functions allocated to software.

This document specifies those measures necessary to achieve these requirements.

The EN 50126 series ([1] [2]) requires that a systematic approach is taken to:

- a) identify hazards, assessing risks and arriving at decisions based on risk criteria,
- b) identify the necessary risk reduction to meet the risk acceptance criteria,
- c) define the overall system safety requirements for the safeguards necessary to achieve the required risk reduction,
- d) select a suitable system architecture,
- e) plan, monitor and control the technical and managerial activities necessary to translate the system safety requirements into a safety-related system of a validated safety integrity level.

As decomposition of the specification into a design comprising safety-related systems and components takes place, further allocation of safety integrity levels is performed. Ultimately this leads to the required software integrity levels.

The current state-of-the-art is such that neither the application of quality assurance methods (so-called fault avoiding measures and fault detecting measures) nor the application of software fault tolerant approaches can guarantee the absolute safety of the software. There is no known way to prove the absence of faults in reasonably complex safety-related software, especially the absence of specification and design faults.

The principles applied in developing high integrity software include, but are not restricted to

- top-down design methods,
- modularity,
- verification of each phase of the development lifecycle,

- verified components and component libraries,
- clear documentation and traceability,
- auditable documents,
- validation,
- assessment,
- configuration management and change control,
- appropriate consideration of organization and personnel competency issues.

At the system level, the allocation of system requirements to software functions takes place. This includes the definition of the required software integrity level for the functions. The successive functional steps in the application of this document are shown in Figure 1 and are as follows:

- f) define the Software Requirements Specification and in parallel consider the software architecture. The software architecture is where the strategy is developed for the software and the software integrity level (7.2 and 7.3);
- g) design, implement and test the software according to the Software Quality Assurance Plan, software integrity level and the software lifecycle (7.4 and 7.5);
- h) integrate the software on the target hardware and verify functionality (7.6);
- i) validate and deploy the software (7.7 and 9.1);
- j) software maintenance, if required during operational life (9.2).

A number of assurance activities run across the software development. These include testing (6.1), verification (6.2), validation (6.3), assessment (6.4), quality assurance (6.5) and modification and change control (6.6).

Requirements are given for support tools (6.7) and for systems which are configured by application data (Clause 8).

Requirements are also given for the independence of roles and the competence of staff involved in software development (5.1, 5.2 and Annex B).

This document does not mandate the use of a particular software development lifecycle. However, illustrative lifecycle and documentation sets are given in 5.3, 7.1, and in C.1.

Tables have been formulated ranking various techniques/measures against the software integrity levels 1 to 4 and for Basic Integrity. The tables are in Annex A. Cross-referenced from the tables is a bibliography giving a brief description of each technique/measure with references to further sources of information. The bibliography of techniques is in Annex D.

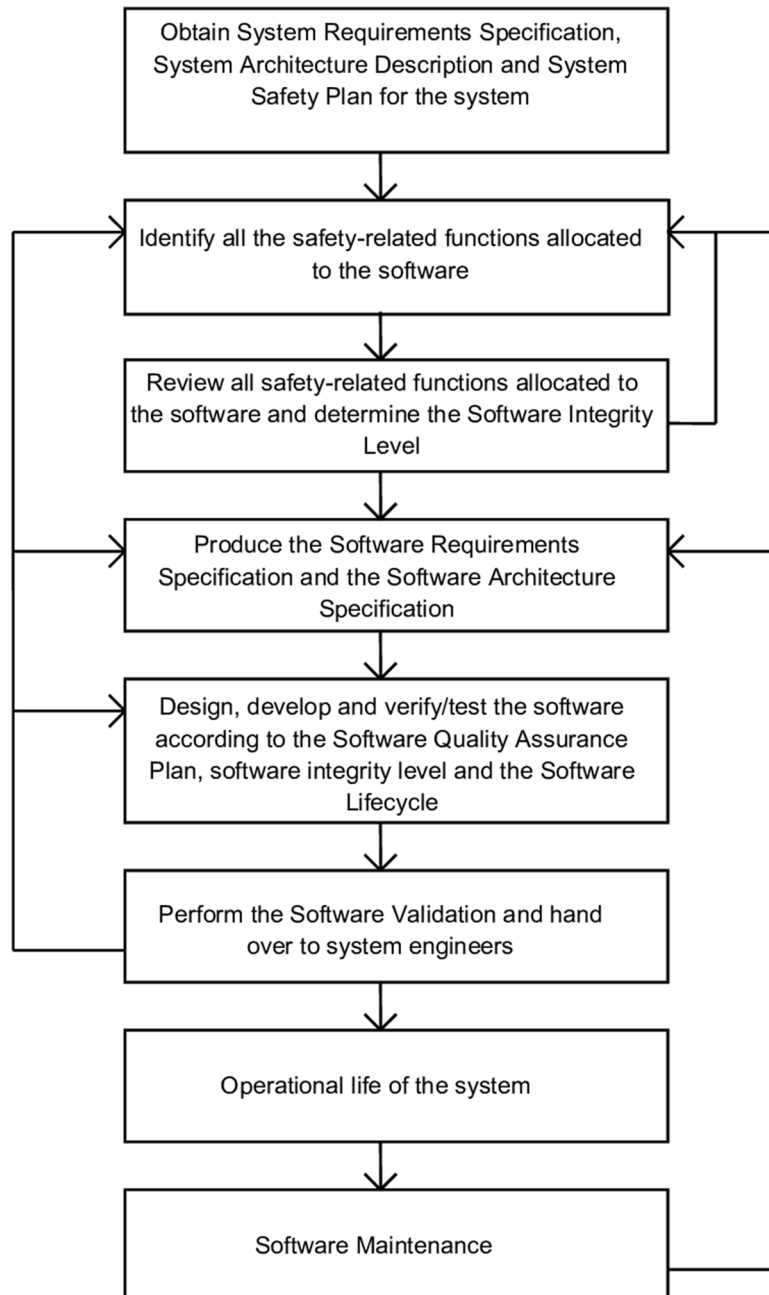
NOTE 2 Some entries within this document have been intentionally left blank. This ensures that the numbering is, as far as reasonably practicable and where not impacting readability, unchanged with respect to EN 50128 and EN 50657. This is meant to facilitate transition and adoption of this document, where relevant.

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

NOTE 3 ISO/IEC and CEN/CENELEC publications that address cybersecurity in depth are [3], [4], [5] and [17].

**EN 50716:2023 (E)**

It may be necessary to balance between measures against systematic errors and measures against security threats. An example is the need for fast security updates of software arising from security threats, whereas if such software is safety related, it should be thoroughly developed, tested, validated and approved before any update.



**Figure 1 — Illustrative Software Route Map**

## 1 Scope

1.1 This document specifies the process and technical requirements for the development of software for programmable electronic systems for use in:

- control, command for signalling applications,
- applications on-board of rolling stock.

This document is not intended to be applied in the area of electric traction power supply (fixed installations) or for power supply and control of conventional applications, e.g. station power supply for offices, shops. These applications are typically covered by standards for energy distribution and/or non-railway sectors and/or local legal frameworks.

1.2 This document is applicable exclusively to software and the interaction between software and the system of which it is part.

1.3 Intentionally left blank

1.4 This document applies to software as per subclause 1.1 of this document used in railway systems, including:

- application programming,
- operating systems,
- support tools,
- firmware.

Application programming comprises high level programming, low level programming and special purpose programming (for example: programmable logic controller ladder logic).

1.5 This document also addresses the use of pre-existing software (as defined in 3.1.16) and tools. Such software can be used if the specific requirements in 7.3.4.7 and 6.5.4.16 on pre-existing software and for tools in 6.7 are fulfilled.

1.6 Intentionally left blank

1.7 This document considers that modern application design often makes use of software that is suitable as a basis for various applications. Such software is then configured by application data for producing the executable software for the application.

1.8 Intentionally left blank

1.9 This document is not intended to be retrospective. It therefore applies primarily to new developments and only applies in its entirety to existing systems if these are subjected to major modifications. For minor changes, only 9.2 applies. However, application of this document during upgrades and maintenance of existing software is advisable.

1.10 For the development of User Programmable Integrated Circuits (e.g. field programmable gate arrays (FPGA) and complex programmable logic devices (CPLD)) guidance is provided in EN 50129:2018 Annex F for safety related functions and in EN 50155:2017 for non-safety related functions. Software running on softcore processors of User Programmable Integrated Circuits is within the scope of this document.

## 2 Normative references

There are no normative references in this document.

**koniec náhľadu – text ďalej pokračuje v platenej verzii STN**