



# SLOVENSKI STANDARD SIST EN 50128:2002

01-marec-2002

---

Železniške aplikacije - Telekomunikacijska, signalna in obrambna sistemstva - Programski kod za sisteme za nadzor in zaščito železnice

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

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

(standards.iteh.ai)

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

Ta slovenski standard je istoveten z: **EN 50128:2001**

---

## ICS:

35.240.60	Uporabniške rešitve IT v transportu in trgovini	IT applications in transport and trade
45.020	Železniška tehnika na splošno	Railway engineering in general

**SIST EN 50128:2002**

en

**iTeh STANDARD PREVIEW**  
**(standards.iteh.ai)**

SIST EN 50128:2002

<https://standards.iteh.ai/catalog/standards/sist/2c35101a-844c-4645-8a87-b564fa4f9970/sist-en-50128-2002>

EUROPEAN STANDARD

**EN 50128**

NORME EUROPÉENNE

EUROPÄISCHE NORM

March 2001

ICS 29.280; 45.060.10

English version

**Railway applications -  
Communications, 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, Signal-  
technik und Datenverarbeitungssysteme -  
Software für Eisenbahnsteuerungs- und  
Überwachungssysteme

**iTeh STANDARD PREVIEW**  
**(standards.iteh.ai)**

SIST EN 50128:2002

This European Standard was approved by CENELEC on 2000-11-01. 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 Central Secretariat 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 Central Secretariat has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Czech Republic, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and United Kingdom.

# CENELEC

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

**Central Secretariat: rue de Stassart 35, B - 1050 Brussels**

## Foreword

This European Standard was prepared by SC 9XA, Communication, signalling and processing systems, of Technical Committee CENELEC TC 9X, Electrical and electronic applications for railways.

The text of the draft was submitted to the formal vote and was approved by CENELEC as EN 50128 on 2000-11-01.

The following dates were fixed:

- latest date by which the EN has to be implemented at national level by publication of an identical national standard or by endorsement (dop) 2001-11-01
- latest date by which the national standards conflicting with the EN have to be withdrawn (dow) 2003-11-01

This European Standard should be read in conjunction with EN 50126: "Railway applications - The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS)" and EN 50129: "Railway applications - Safety related electronic systems for signalling".

Annexes designated "normative" are part of the body of the standard.

Annexes designated "informative" are given for information only.

In this standard, annex A is normative and annex B is informative.

**iTeh STANDARD PREVIEW**  
**(standards.iteh.ai)**

SIST EN 50128:2002

<https://standards.iteh.ai/catalog/standards/sist/2c35101a-844c-4645-8a87-b564fa4f9970/sist-en-50128-2002>

## Contents

	<b>Pages</b>
Introduction .....	5
1 Scope .....	7
2 Normative references.....	7
3 Definitions.....	8
4 Objectives and conformance.....	11
5 Software safety integrity levels.....	11
5.1 Objective.....	11
5.2 Requirements.....	12
6 Personnel and responsibilities .....	13
6.1 Objective.....	13
6.2 Requirements.....	13
7 Lifecycle issues and documentation .....	14
7.1 Objectives.....	14
7.2 Requirements.....	14
8 Software requirements specification.....	17
8.1 Objectives.....	17
8.2 Input documents.....	17
8.3 Output documents.....	17
8.4 Requirements.....	17
9 Software architecture .....	18
9.1 Objectives.....	18
9.2 Input documents.....	19
9.3 Output documents.....	19
9.4 Requirements.....	19
10 Software design and implementation.....	20
10.1 Objectives.....	20
10.2 Input documents.....	21
10.3 Output documents.....	21
10.4 Requirements.....	21
11 Software verification and testing .....	24
11.1 Objective.....	24
11.2 Input documents.....	24
11.3 Output documents.....	24
11.4 Requirements.....	24
12 Software/hardware integration.....	27
12.1 Objectives.....	27
12.2 Input documents.....	27
12.3 Output documents.....	28
12.4 Requirements.....	28

iTech STANDARD PREVIEW  
(standards.iteh.ai)

SIST EN 50128:2002

<https://standards.iteh.ai/catalog/standards/sist/2c35101a-844c-4645-8a87-b564fa4f9970/sist-en-50128-2002>

13	Software validation.....	29
13.1	Objective.....	29
13.2	Input documents.....	29
13.3	Output documents.....	29
13.4	Requirements.....	29
14	Software assessment.....	31
14.1	Objective.....	31
14.2	Input documents.....	31
14.3	Output documents.....	31
14.4	Requirements.....	31
15	Software quality assurance.....	32
15.1	Objectives.....	32
15.2	Input documents.....	32
15.3	Output documents.....	32
15.4	Requirements.....	32
16	Software maintenance.....	34
16.1	Objective.....	34
16.2	Input documents.....	34
16.3	Output documents.....	35
16.4	Requirements.....	35
17	Systems configured by application data.....	36
17.1	Objectives.....	36
17.2	Input documents.....	36
17.3	Output documents.....	36
17.4	Requirements.....	37
17.4.1	Data Preparation Lifecycle.....	37
17.4.2	Data Preparation Procedures and Tools.....	37
17.4.3	Software Development.....	38
Annex A	(normative) Criteria for the Selection of Techniques and Measures.....	45
Annex B	(informative) Bibliography of Techniques.....	60

## Figures

Figure 1	– Integrity Levels for Safety-Related Systems.....	39
Figure 2	– Software Safety Route Map.....	40
Figure 3	– Development Lifecycle 1.....	41
Figure 4	– Development Lifecycle 2.....	42
Figure 5	- Independence Versus Software Integrity Level.....	43
Figure 6	– Relationship between Generic System Development and Application Development.....	44

## Introduction

This Standard is part of a group of related Standards. The others are EN 50126 "Railway applications - The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS)" and EN 50129 "Railway applications - Safety related electronic systems for signalling". EN 50126 addresses system issues on the widest scale, while EN 50129 addresses the approval process for individual systems which may exist within the overall railway control and protection system. This Standard concentrates on the methods which need to be used in order to provide software which meets the demands for safety integrity which are placed upon it by these wider considerations.

This Standard owes much of its direction to earlier work done by Working Group 9 of IEC/TC 65. The work of WG 9 resulted in a generic standard for software for safety systems which is now part of IEC 61508. A particular aspect of the work by WG 9 is its inclusion of Software Integrity Level 0, which covers non-safety software, as well as Software Integrity Levels 1 to 4, which cover safety-related and safety-critical software. This Standard also covers all five Software Integrity Levels.

Account has also been taken of the work of the Institution of Railway Signal Engineers (the IRSE), in particular its Technical Report Number 1, which addressed the same topic.

The key concept of this European Norm is that of levels of software safety integrity. The more dangerous the consequences of a software failure, the higher the software safety integrity level will be.

This European Standard has identified techniques and measures for 5 levels of software safety integrity where 0 is the minimum level and 4 the highest level. Four of these levels, 1 to 4, refer to safety-related software, whilst level 0 refers to non safety-related software. This level has been included as normative in order to allow a smooth transition between software developments for non-safety related systems and those for safety-related systems. The required techniques and measures for each software safety integrity level and for the non safety-related level are shown in the tables. In this version, 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 European Standard does not give guidance on which level of software integrity is appropriate for a given risk. This decision will depend upon the many factors including the nature of the application, the extent to which other systems carry out safety functions and social and economic factors.

It is the function of EN 50126 and EN 50129 to specify the safety functions allocated to software.

This European Standard specifies those measures necessary to achieve these requirements. The process is illustrated in Figure 1.

EN 50126 and EN 50129 require that a systematic approach be taken to:

- i) identifying hazards, risks and risk criteria;
- ii) identifying the necessary risk reduction to meet the risk criteria;
- iii) defining an overall System Safety Requirements Specification for the safeguards necessary to achieve the required risk reduction;
- iv) selecting a suitable system architecture;
- v) planning, monitoring and controlling the technical and managerial activities necessary to translate the System Safety Requirements Specification into a Safety-Related System of a validated safety performance (or safety integrity).

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 safety integrity levels.

The current state-of-the-art is such that neither the application of quality assurance methods (so-called fault avoiding measures) nor the application of software fault tolerant approaches can guarantee the

absolute safety of the system. 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 modules and module libraries;
- clear documentation;
- auditable documents; and
- validation testing.

These and related principles must be correctly applied. This standard specifies the level of assurance required to demonstrate this at each software safety integrity level.

After the System Safety Requirements Specification, which identifies all safety functions allocated to software and determines the system safety integrity level, has been obtained or produced, the functional steps in the application of this European Standard are shown in Figure 2 and are as follows:

- i) define the Software Requirements Specification and in parallel consider the software architecture. the software architecture is where the basic safety strategy is developed for the software and the software safety integrity level (clauses 5, 8 and 9);
- ii) design, develop and test the software according to the Software Quality Assurance Plan, software safety integrity level and the software lifecycle (clause 10);
- iii) integrate the software on the target hardware (clause 12);
- iv) validate the software (clause 13);
- v) if software maintenance is required during operational life then re-activate this European Standard as appropriate (clause 16).

A number of activities run across the software development. These include verification (clause 11), assessment (clause 14) and quality assurance (clause 15).

Requirements are given for systems which are configured by application data (clause 17).

Requirements are also given for the competency of staff involved in software development (clause 6).

The standard does not mandate the use of a particular software development lifecycle. However a recommended lifecycle and documentation set are given (clause 7 and Figures 3 and 4).

Tables have been formulated ranking various techniques/measures against the 5 software safety integrity levels. The tables are in annex A. Cross-referenced to the tables is a bibliography giving a brief description of each technique/measure with references to further sources of information. The bibliography is in annex B.



## 1 Scope

1.1 This European Standard specifies procedures and technical requirements for the development of programmable electronic systems for use in railway control and protection applications. It is aimed at use in any area where there are safety implications. These may range from the very critical, such as safety signalling to the non-critical, such as management information systems. These systems may be implemented using dedicated microprocessors, programmable logic controllers, multiprocessor distributed systems, larger scale central processor systems or other architectures.

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

1.3 Software safety integrity levels above zero are for use in systems in which the consequences of failure could include loss of life. Economic or environmental considerations, however, may also justify the use of higher software safety integrity levels.

1.4 This European Standard applies to all software used in development and implementation of railway control and protection 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 The use of standard, commercially available software and tools is also addressed in this European Standard.

1.6 This European Standard also addresses the requirements for systems configured by application data.

1.7 This European Standard is not intended to address commercial issues. These should be addressed as an essential part of any contractual agreement. All the clauses of this European Standard will need careful consideration in any commercial situation.

1.8 This European Standard 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 clause 16 applies.

## 2 Normative references

This European Standard incorporates by dated or undated reference, provisions from other publications. These normative references are cited at the appropriate places in the text and the publications are listed hereafter. For dated references, subsequent amendments to or revisions of any of these publications apply to this European Standard only when incorporated in it by amendment or revision. For undated references the latest edition of the publication referred to applies (including amendments).

EN 50126            Railway applications - The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS)

EN 50129\*         Railway applications - Safety related electronic systems for signalling

---

\* at draft stage

EN 50159-1	Railway applications - Communication, signalling and processing systems Part 1: Safety-related communication in closed transmission systems
EN 50159-2	Railway applications - Communication, signalling and processing systems Part 2: Safety-related communication in open transmission systems
EN ISO 9001	Quality systems - Model for quality assurance in design/development, production, installation and servicing
EN ISO 9000-3	Quality management and quality assurance standards – Part 3: Guidelines for the application of ISO 9001:1994 to the development, supply, installation and maintenance of computer software

### 3 Definitions

For the purposes of this European Standard, the following definitions apply. For terms not defined here, the following references should be consulted in order of priority:

EN ISO 8402	Quality management and quality assurance – Vocabulary
IEC 60050-191	International Electrotechnical Vocabulary of Chapter 191: Dependability and quality of service
IEEE 610.12	IEEE standard glossary of software engineering terminology
ISO/IEC 2382	Information Technology Vocabulary
ISO/IEC 9126	Information Technology – Software Product Evaluation – Quality characteristics and guidelines for their use

#### 3.1 assessment

process of analysis to determine whether the Design Authority and the Validator have achieved a product that meets the specified requirements and to form a judgement as to whether the product is fit for its intended purpose

#### 3.2 assessor

person or agent appointed to carry out the assessment

#### 3.3 availability

ability of a product to be in a state to perform a required function under given conditions at a given instant of time or over a given time interval, assuming the required external resources are provided

#### 3.4 commercial off-the-shelf (COTS) software

software defined by market-driven need, commercially available and whose fitness for purpose has been demonstrated by a broad spectrum of commercial users

#### 3.5 design authority

body responsible for the formulation of a design solution to fulfil the specified requirements and for overseeing the subsequent development and setting to work of a system in its intended environment

#### 3.6 designer

one or more persons assigned by the Design Authority to analyse and transform specified requirements into acceptable design solutions which have the required safety integrity

**3.7****element**

part of a product that has been determined to be a basic unit or building block. An element may be simple or complex

**3.8****error**

deviation from the intended design which could result in unintended system behaviour or failure

**3.9****failure**

deviation from the specified performance of a system. A failure is the consequence of a fault or error in a system

**3.10****fault**

abnormal condition that could lead to an error or a failure in a system. A fault can be random or systematic

**3.11****fault avoidance**

use of design techniques which aim to avoid the introduction of faults during the design and construction of the system

**3.12****fault tolerance**

built-in capability of a system to provide continued correct provision of service as specified, in the presence of a limited number of hardware or software faults

**3.13****firmware**

ordered set of instructions and associated data stored in a way that is functionally independent of main storage, usually in a ROM

**3.14****generic software**

generic software is software which can be used for a variety of installations purely by the provision of application-specific data

**3.15****implementer**

one or more persons assigned by the Design Authority to transform specified designs into their physical realisation

**3.16****product**

collection of elements, interconnected to form a system, sub-system or item of equipment, in a manner which meets the specified requirements. In this European Standard, a product may be considered to consist entirely of elements of software or documentation

**3.17****programmable logic controller (PLC)**

solid-state control system which has a user programmable memory for storage of instructions to implement specific functions

**3.18****reliability**

ability of an item to perform a required function under given conditions for a given period of time

**3.19****requirements traceability**

objective of requirements traceability is to ensure that all requirements can be shown to have been properly met

**3.20****risk**

combination of the frequency, or probability, and the consequence of a specified hazardous event

**3.21****safety**

Freedom from unacceptable levels of risk

**3.22****safety authority**

body responsible for certifying that the safety-related system is fit for service and complies with relevant statutory and regulatory safety requirements

**3.23****safety-related software**

software which carries responsibility for safety

**3.24****software**

intellectual creation comprising the programs, procedures, rules and any associated documentation pertaining to the operation of a system

NOTE Software is independent of the media used for transport.

**3.25****software life-cycle**

activities occurring during a period of time that starts when software is conceived and ends when the software is no longer available for use. The software lifecycle typically includes a requirements phase, development phase, test phase, integration phase, installation phase and a maintenance phase

**3.26****software maintainability**

capability of a system to be modified to correct faults, improve performance or other attributes, or adapt it to a different environment

**3.27****software maintenance**

Action, or set of actions, carried out on software after its acceptance by the final user. The aim is to improve, increase and/or correct its functionality

**3.28****software safety integrity level**

classification number which determines the techniques and measures that have to be applied in order to reduce residual software faults to an appropriate level

**3.29****system safety integrity level**

number which indicates the required degree of confidence that a system will meet its specified safety features

**3.30****traceability**

degree to which a relationship can be established between two or more products of a development process, especially those having a predecessor/successor or master/subordinate relationship to one another

**3.31  
validation**

activity of demonstration, by analysis and test, that the product meets, in all respects, its specified requirements

**3.32  
validator**

person or agent appointed to carry out validation

**3.33  
verification**

activity of determination, by analysis and test, that the output of each phase of the life-cycle fulfils the requirements of the previous phase

**3.34  
verifier**

person or agent appointed to carry out verification

**4 Objectives and conformance**

4.1 In each of the following clauses, the objectives and requirements of the clause are detailed.

4.2 To conform to this European Standard it shall be shown that each of the requirements have been satisfied to the software safety integrity level defined and therefore the clause objective has been met.

4.3 Where a requirement is qualified by the words "To the extent required by the software safety integrity level", this indicates that a range of techniques and measures can be used to satisfy that requirement.

4.4 Where 4.3 applies the tables detailed in this European Standard should be used to assist in the selection of techniques and measures appropriate to the software safety integrity level.

4.5 If a technique or measure is ranked as highly recommended (HR) in the tables then the rationale for not using that technique should be detailed and recorded either in the Software Quality Assurance Plan or in another document referenced by the Software Quality Assurance Plan. This is not necessary if an approved combination of techniques given in the corresponding table is used.

4.6 If a technique or measure is proposed to be used that is not contained in the tables then its effectiveness and suitability in meeting the particular requirement and overall objective of the clause shall be justified and recorded in either the Software Quality Assurance Plan or in another document referenced by the Software Quality Assurance Plan.

4.7 Compliance with the requirements of a particular clause and their respective techniques and measures detailed in the tables shall be assessed by the inspection of documents required by this standard, other objective evidence, auditing and the witnessing of tests.

4.8 This European Standard requires the use of a package of techniques and their correct application. These techniques are required from the tables and detailed in the bibliography.

**5 Software safety integrity levels****5.1 Objective**

To describe the assignment of software safety integrity levels to the software.

## 5.2 Requirements

5.2.1 There shall be produced, in accordance with EN 50126 and EN 50129

- System Requirements Specification,
- System Safety Requirements Specification,
- System Architecture Description,
- System Safety Plan,

which include:

- safety functions;
- configuration or architecture of the system;
- hardware reliability requirements;
- safety integrity requirements.

The software safety integrity level shall be specified through following the general process for obtaining a safety integrity level identified in EN 50126.

5.2.2 The required software safety integrity level shall be decided on the basis of the level of risk associated with the use of the software in the system and the system safety integrity level.

5.2.3 Without further precautions, the software safety integrity level shall be, as a minimum, identical to the system safety integrity level. However, if mechanisms exist to prevent the failure of a software module from causing the system to go to an unsafe state, the software safety integrity level of the module may be reduced.

5.2.4 Risks which shall be taken into account are those associated with the following hazard consequences:

- loss of human life or lives;
- injuries to or illness of persons;
- environmental pollution; and
- loss of or damage to property.

5.2.5 Risk may be quantified but it is not possible to specify the software safety integrity in the same manner. Therefore for this European Standard the software safety integrity shall be specified as one of the following five levels:

Software safety integrity level	Description of software safety integrity
4	Very High
3	High
2	Medium
1	Low
0	Non safety-related

5.2.6 The software safety integrity level shall be specified in the Software Requirements Specification (clause 8). If different software components have different software safety integrity levels, these shall be specified in the Software Architecture Specification (clause 9).

## 6 Personnel and responsibilities

### 6.1 Objective

To ensure that all personnel who have responsibilities for the software are competent to discharge those responsibilities.

### 6.2 Requirements

6.2.1 As a minimum, the supplier and/or developer and the customer shall implement the relevant parts of EN ISO 9001, in accordance with the guidelines contained in EN ISO 9000-3.

6.2.2 Except at software safety integrity level zero, the safety process shall be implemented under the control of an appropriate safety organisation which is compliant with the "Safety Organisation" sub-clause in the "Evidence of Safety Management" clause of EN 50129.

6.2.3 All personnel involved in all the phases of the Software Lifecycle, including management activities, shall have the appropriate training, experience and qualifications.

6.2.4 It is highly recommended that the training, experience and qualifications of all personnel involved in all the phases of the Software Lifecycle, including management activities, be justified with respect to the particular application, except at software safety integrity level zero.

6.2.5 The justification contained in 6.2.4 shall be recorded in the Software Quality Assurance Plan, and shall include evidence of competency in the following areas, as appropriate:

- i) engineering appropriate to the application area;
- ii) software engineering;
- iii) computer-systems engineering;
- iv) safety engineering;
- v) legal and regulatory framework.

6.2.6 An independent assessor for the software shall be appointed. See also 6.2.10 and 14.4.1.

6.2.7 The assessor shall be given authority to perform the assessment of the software.

6.2.8 Throughout the Software Lifecycle, the parties involved shall be independent, to the extent required by the software safety integrity level, in accordance with Figure 5, which shall be interpreted as follows.

At all software safety integrity levels, the Assessor shall be approved by the Safety Authority and independent from the supplier except in the circumstances defined in 6.2.10.

The Designer/Implementer, Verifier and Validator can all belong to the same company but the following rules for minimum independence shall be complied with:

At software safety integrity level 0:

There are no constraints; the Designer/Implementer, Verifier and Validator can all be the same person.