SIST

SLOVENSKI STANDARD
oSIST prEN 50128:2009

01-oktober-2009

Ģelezniģke naprave - Komunikacijski, signalni in procesni sistemi - Programska oprema za ģelezniģke krmilne in zaģļ itne sisteme

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

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

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: prEN 50128:2009

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

oSIST prEN 50128:2009 **en,fr,de**

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

DRAFT
prEN 50128

July 2009

ICS 35.240.60; 45.020; 93.100

Will supersede EN 50128:2001

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

This draft European Standard is submitted to CENELEC members for CENELEC enquiry.
Deadline for CENELEC: 2010-01-08.

It has been drawn up by CLC/SC 9XA.

If this draft becomes a European Standard, 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.

This draft European Standard was established by CENELEC 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, Bulgaria, Cyprus, the Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and the United Kingdom.

Warning : This document is not a European Standard. It is distributed for review and comments. It is subject to change without notice and shall not be referred to as a European Standard.

CENELEC

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

Central Secretariat: Avenue Marnix 17, B - 1000 Brussels

Project: 20508

Ref. No. prEN 50128:2009 E

Foreword

This draft European Standard was prepared by SC 9XA, Communication, signalling and processing systems, of Technical Committee CENELEC TC 9X, Electrical and electronic applications for railways. It is submitted to the CENELEC enquiry.

This document will supersede EN 50128:2001.

The main changes with respect to the previous edition are listed below:

- requirements on software management and organisation, definition of roles and competencies, deployment and maintenance have been added;
- a new clause on tools has been inserted, based on EN 61508-2:2008;
- tables in Annex A have been updated.

This European Standard should be read in conjunction with EN 50126-1:1999 "*Railway applications – The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS) – Part 1: Basic requirements and generic process*" and EN 50129:2003 "*Railway applications – Communication, signalling and processing systems – Safety related electronic systems for signalling*".

This draft European Standard has been prepared under a mandate given to CENELEC by the European Commission and the European Free Trade Association and covers essential requirements of EC Directives 2001/16/EC and 96/48/EC.

Contents

Figures

Tables

Introduction

This European Standard is part of a group of related standards. The others are EN 50126-1:1999 "*Railway applications – The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS) – Part 1: Basic requirements and generic process*" and EN 50129:2003 "*Railway applications – Communication, signalling and processing systems – Safety related electronic systems for signalling*".

EN 50126-1 addresses system issues on the widest scale, while EN 50129 addresses the approval process for individual systems which can exist within the overall railway control and protection system. This European 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 European Standard provides a set of requirements with which the development, deployment and maintenance of any safety-related software intended for railway control and protection applications shall comply. It defines requirements concerning organisational structure, the relationship between organisations and division of responsibility involved in the deployment and maintenance activities. Criteria for the qualification and expertise of personnel are also recommended in this European Standard.

The key concept of this European Standard is that of levels of software safety integrity. This European Standard addresses five software safety integrity levels where 0 is the lowest and 4 the highest one. The higher the risk resulting from software failure, the higher the software safety integrity level will be.

This European Standard has identified techniques and measures for the five levels of software safety integrity. The required techniques and measures for software safety integrity levels 0-4 are shown in the normative Annex A 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 safety 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 functions and social and economic factors.

It is the within the scope of EN 50126-1 and EN 50129 to define the process of specifying the safety functions allocated to software.

This European Standard specifies those measures necessary to achieve these requirements.

EN 50126-1 and EN 50129 require that a systematic approach be 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 criteria,

c) define an overall System Safety Requirements Specification 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 Specification into a Safety-Related System of a validated 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 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 and
- appropriate consideration of organisation and personnel competency issues.

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

The System Safety Requirements Specification identifies all safety functions allocated to software and determines their system safety integrity level. The successive functional steps in the application of this European Standard are shown in Figure 1 and are as follows:

a) define the Software Requirements Specification and in parallel consider the software architecture. The software architecture is where the safety strategy is developed for the software and the software safety integrity level (7.2 and 7.3);

b) design, develop and test the software according to the Software Quality Assurance Plan, software safety integrity level and the software lifecycle (7.4 and 7.5);

c) integrate the software on the target hardware and verify functionality (7.6);

d) accept and deploy the software (7.7 and 9.1);

e) if software maintenance is required during operational life then re-activate this European Standard as appropriate (9.2).

A number of 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).

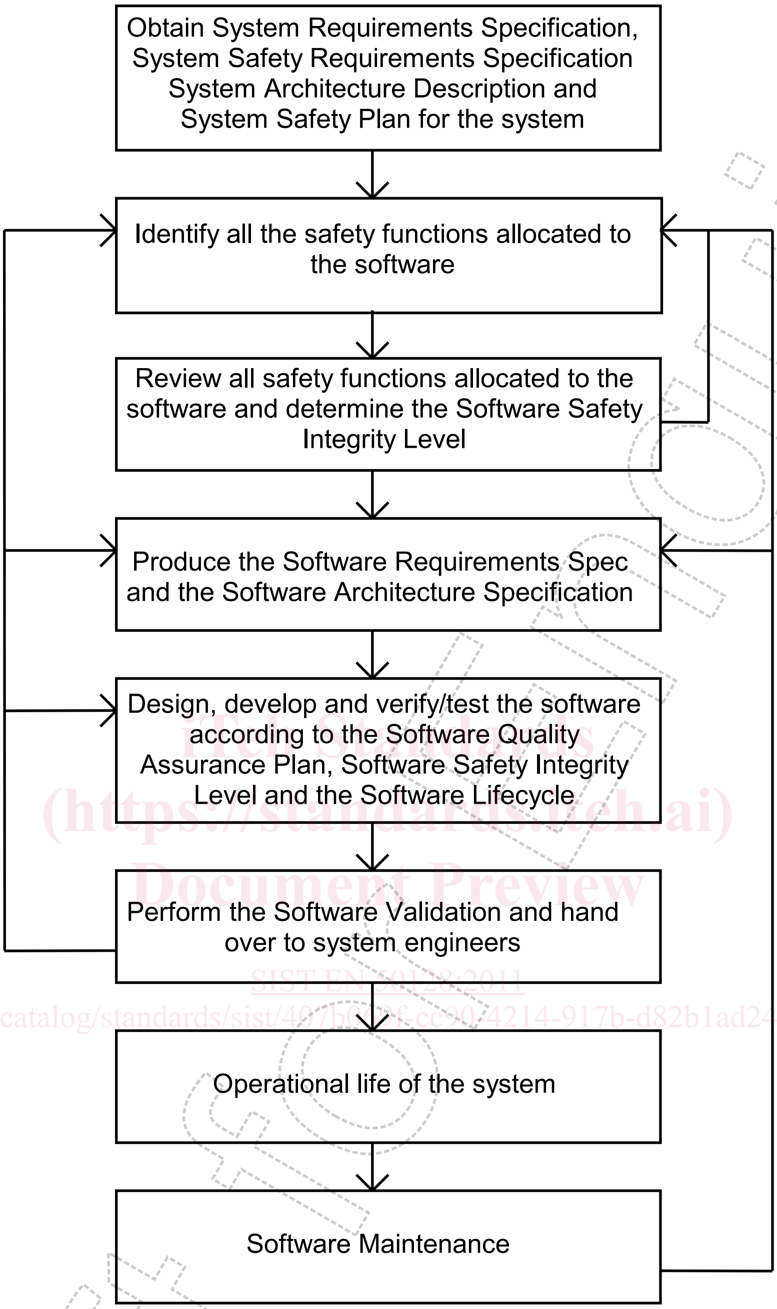
Requirements are given for support tools (6.7) and for systems which are configured by application data or algorithms (8.1).

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

This European Standard does not mandate the use of a particular software development lifecycle. However, illustrative lifecycle and documentation sets are given (5.3, Figures 2 and 3 and 7.1).

Tables have been formulated ranking various techniques/measures against the software safety integrity levels 1-4. 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 of techniques is in Annex D.

Finally, Annex C provides an example of how the flow between document generation and checking could take place.


Obtain System Requirements Specification, System Safety Requirements Specification System Architecture Description and System Safety Plan for the system
Identify all the safety functions allocated to the software
Review all safety functions allocated to the software and determine the Software Safety Integrity Level
Produce the Software Requirements Spec and the Software Architecture Specification
Design, develop and verify/test the software according to the Software Quality Assurance Plan, Software Safety Integrity Level and the Software Lifecycle
Perform the Software Validation and hand over to system engineers
Operational life of the system
Software Maintenance

265

266 **Figure 1 – Illustrative Software Route Map**

1 Scope

1.1 This European Standard specifies the process and technical requirements for the development of software for 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 systems can 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 This European Standard is not relevant for software that has been identified as having no impact on safety, i.e. software of which failures cannot affect the identified safety functions.

1.4 This European Standard applies to all safety related software used in 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 This European Standard also addresses the use of pre-existing software and tools. Such software may be used, if the specific requirements in 7.3.4.7 and 6.5.4.13 on pre-existing software and for tools in 6.7 are fulfilled.

1.6 Software developed according to any version of this European Standard will be considered as compliant and not subject to the requirements on pre-existing software.

1.7 This European Standard considers that modern application design often makes use of generic software that is suitable as a basis for various applications. Such generic software is then configured by data, algorithms, or both, for producing the executable software for the application. The general (sub)clauses 1 to 6.7, 9.1 and 9.2 of this European Standard apply to generic software as well as for application data or algorithms. The specific subclauses 7.1 to 7.7 apply only for generic software while 8.1 provides the specific requirements for application data or algorithms.

1.8 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.

1.9 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 9.2 applies. The determination of the nature and scope of change will be at the discretion of the assessor. However, application of this European Standard during upgrades and maintenance of existing software is highly recommended.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies. In case of clauses in the latest edition of the publication referred that are in contradiction to clauses in this European Standard, this European Standard will overrule the referenced clauses.

EN 50126-1:1999	Railway applications – The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS) – Part 1: Basic requirements and generic process
EN 50129:2003	Railway applications – Communication, signalling and processing systems – Safety related electronic systems for signalling
EN ISO 9000	Quality management systems – Fundamentals and vocabulary (ISO 9000)
EN ISO 9001	Quality management systems – Requirements (ISO 9001)
ISO/IEC 90003:2004	Software engineering – Guidelines for the application of ISO 9001:2000 to computer software
ISO/IEC 9126 series	Software engineering – Product quality

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1.1
assessment
process of analysis to determine whether software, which may include process, documentation, system, subsystem hardware and/or software components, meets the specified requirements and to form a judgement as to whether the software is fit for its intended purpose. Safety assessment is assessment focused on but not limited to the safety properties of a system

3.1.2
assessor
entity that carries out an assessment

3.1.3
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

3.1.4
component
component is a constituent part of software which has well-defined interfaces and behaviour with respect to the software architecture and design and fulfils the following criteria:

- it is designed according to “Components” (see Table A.19);
- it covers a specific subset of software requirements;
- it is clearly identified and has an independent version inside the configuration management system or is a part of a collection of components (e. g. subsystems) which have an independent version

3.1.5
customer
entity which purchases a railway control and protection system including the software

3.1.6
designer
entity that analyses and transforms specified requirements into acceptable design solutions which have the required safety integrity

3.1.7
entity
person, group or organisation who fulfils a role as defined in this European Standard

3.1.8
error, fault
defect, mistake or inaccuracy which could result in failure or in a deviation from the intended performance or behaviour

3.1.9
failure
unacceptable difference between required and observed performance

3.1.10
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

3.1.11
firmware
ordered set of instructions and associated data stored in a way that is functionally independent of main storage

3.1.12
generic software
software which can be used for a variety of installations purely by the provision of application-specific data and/or algorithms

3.1.13
implementer
entity that transforms specified designs into their physical realisation

3.1.14
integration
process of assembling software and/or hardware items, according to the architectural and design specification, and testing the integrated unit

3.1.15
integrator
entity that carries out software integration

3.1.16
pre-existing software
all software developed prior to the application currently in question is classed as pre-existing software including

- COTS (commercial off-the-shelf) and open-source software,
- software previously developed but not in accordance with this European Standard

3.1.17
programmable logic controller
solid-state control system which has a user programmable memory for storage of instructions to implement specific functions