BS EN 50716:2023

BSI Standards Publication

Railway Applications — Requirements for software development

bsi.

National foreword

This British Standard is the UK implementation of EN 50716:2023. It supersedes BS EN 50657:2017+A1:2023 and BS EN 50128:2011+A2:2020, which are withdrawn.

The UK participation in its preparation was entrusted to Technical Committee GEL/9, Railway Electrotechnical Applications.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

Published by BSI Standards Limited 2023

ISBN 978 0 539 21140 5

ICS 35.240.60; 35.240.60

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 November 2023.

Amendments/corrigenda issued since publication

Date	Text affected

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

EN 50716

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

Contents

Page

List of figures

List of tables

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- latest date by which the national standards conflicting with this document have to be withdrawn (dow) 2026-10-30

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.

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.

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].

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.

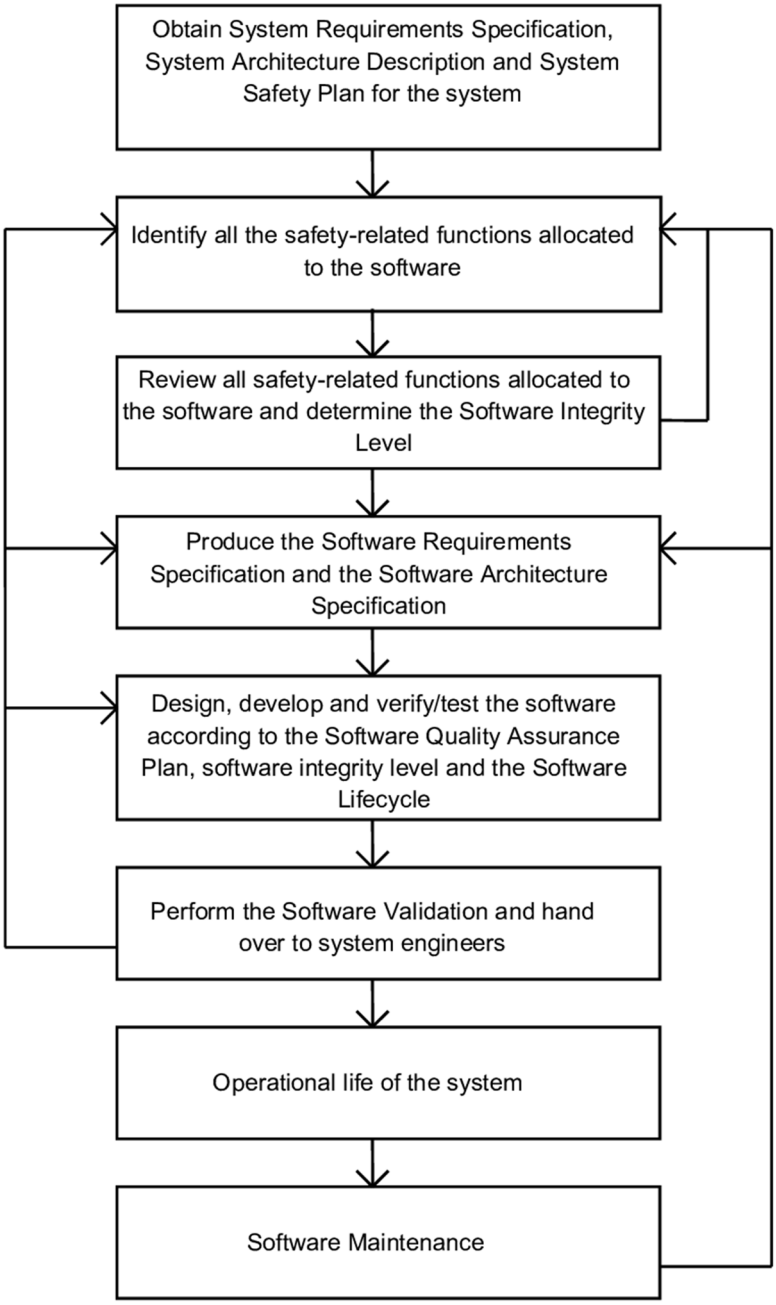

Figure 1 — Illustrative Software Route Map

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.