Irish Standard
I.S. EN 50657:2017

Railways Applications - Rolling stock applications - Software on Board Rolling Stock

I.S. EN 50657:2017

Incorporating amendments/corrigenda/National Annexes issued since publication:

The National Standards Authority of Ireland (NSAI) produces the following categories of formal documents:

I.S. xxx: Irish Standard — national specification based on the consensus of an expert panel and subject to public consultation.

S.R. xxx: Standard Recommendation — recommendation based on the consensus of an expert panel and subject to public consultation.

SWiFT xxx: A rapidly developed recommendatory document based on the consensus of the participants of an NSAI workshop.

This document replaces/revises/consolidates the NSAI adoption of the document(s) indicated on the CEN/CENELEC cover/Foreword and the following National document(s):

NOTE: The date of any NSAI previous adoption may not match the date of its original CEN/CENELEC document.

This document is based on:	*Published:*
EN 50657:2017	2017-08-11

This document was published under the authority of the NSAI and comes into effect on:	ICS number:
2017-08-29	35.080 35.240.60
	NOTE: If blank see CEN/CENELEC cover page

NSAI	T +353 1 807 3800	Sales:
1 Swift Square,	F +353 1 807 3838	T +353 1 857 6730
Northwood, Santry	E standards@nsai.ie	F +353 1 857 6729
Dublin 9	W NSAI.ie	W standards.ie

Údarás um Chaighdeáin Náisiúnta na hÉireann

National Foreword

I.S. EN 50657:2017 is the adopted Irish version of the European Document EN 50657:2017, Railways Applications - Rolling stock applications - Software on Board Rolling Stock

This document does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

For relationships with other publications refer to the NSAI web store.

Compliance with this document does not of itself confer immunity from legal obligations.

In line with international standards practice the decimal point is shown as a comma (,) throughout this document.

This page is intentionally left blank

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

EN 50657

August 2017

ICS 35.080; 35.240.60

English Version

Railways Applications - Rolling stock applications - Software on Board Rolling Stock

Applications ferroviaires - Applications du matériel roulant - Logiciels embarqués

Bahnanwendungen - Anwendungen für Schienenfahrzeuge - Software auf Schienenfahrzeugen

This European Standard was approved by CENELEC on 2017-05-08. 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, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

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

CEN-CENELEC Management Centre: Avenue Marnix 17, B-1000 Brussels

Contents

Page

Figures

Tables

European foreword

This document (EN 50657:2017) has been prepared by CLC/SC 9XB, “Electrical, electronic and electromechanical material on board rolling stock, including associated software”.

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)	2018-05-08
• latest date by which the national standards conflicting with this document have to be withdrawn	(dow)	2020-05-08

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 has been prepared under a mandate given to CENELEC by the European Commission and the European Free Trade Association, and supports essential requirements of EU Directive(s).

For the relationship with EU Directive(s) see informative Annex ZZ, which is an integral part of this document.

This document adapts EN 50128:2011 (prepared by CLC/SC 9XA “Communication, signalling and processing systems”) for the application in the Rolling Stock domain. It uses the same structure and section numbering as EN 50128:2011. Where requirements of EN 50128:2011 do not apply to rolling stock, the respective text is replaced by the term “intentionally left empty”.

The main changes with respect to EN 50128:2011 are listed in Annex E.

Introduction

This European Standard is related to, and should be read in conjunction with the EN 50126 series, *Railway applications — The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS)*.

This European Standard concentrates on the methods which need to be used in order to provide software which meets the demands for software integrity which are placed upon it by these wider considerations.

This European Standard provides a set of requirements for the development, deployment and maintenance of any software intended for railway rolling stock 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 European Standard.

The key concept of this European Standard is that of levels of software integrity. This European Standard 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 European Standard was first introduced in the EN 50126 series.

This European Standard has identified techniques and measures for the five levels of software integrity. The required techniques and measures for basic integrity and for the safety integrity levels 1-4 are shown in the normative tables of Annex A. 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-related functions and social and economic factors.

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

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

The EN 50126 series 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 Specification 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, and

— 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 European Standard 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 safety strategy is developed for the software and the software integrity level (7.2 and 7.3);

g) design, develop 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) accept and deploy the software (7.7 and 9.1);

j) 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 (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 European Standard does not mandate the use of a particular software development lifecycle. However, illustrative lifecycle and documentation sets are given in 5.3, Figure 3 and Figure 4 and in 7.1.

Tables have been formulated ranking various techniques/measures against the safety integrity levels 1-4 and for basic integrity. 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.

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

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.

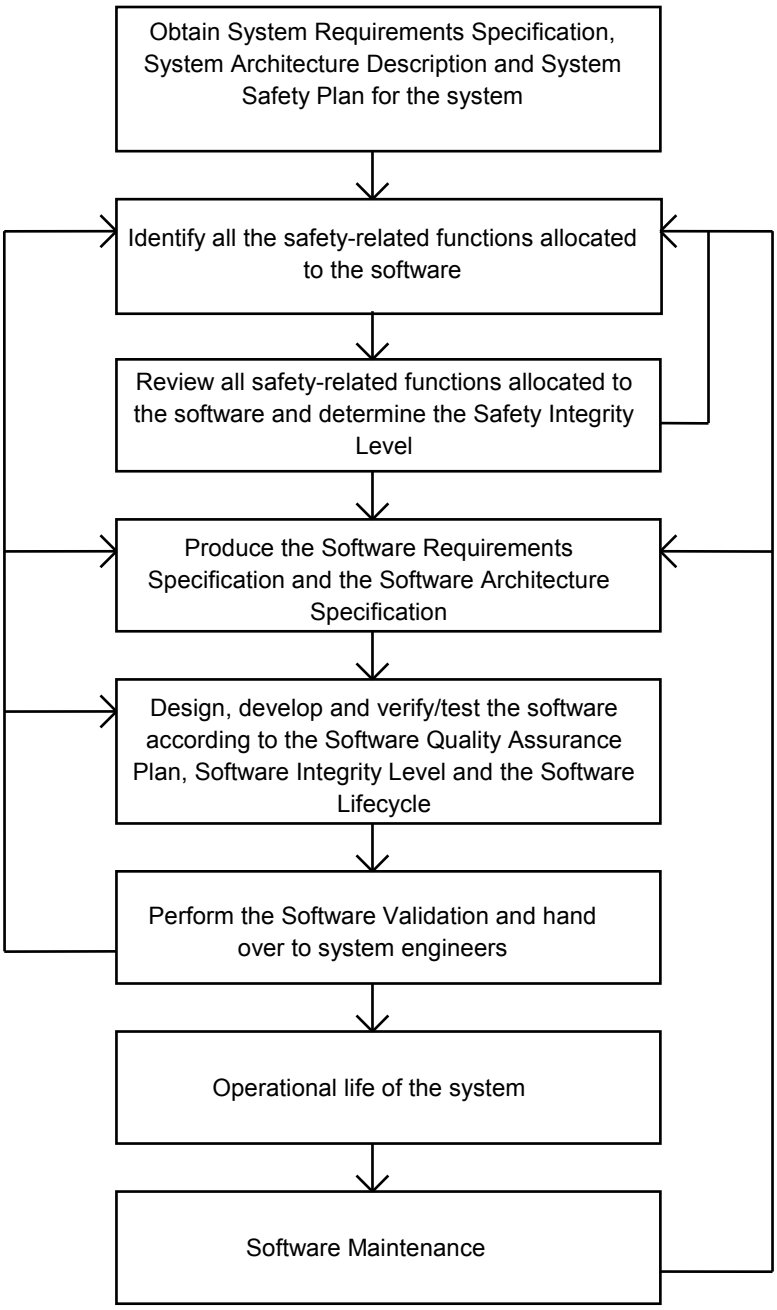

Figure 1 — Illustrative Software Route Map

1 Scope

1.1 This European Standard specifies the process and technical requirements for the development of software for programmable electronic systems for use in rolling stock applications.

Outside the scope of this standard is software that:

— is part of signalling equipment (CENELEC sub-committee SC9XA applications) installed on board trains, or

— does not contribute to, and is segregated from Rolling Stock operational functions.

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 Entry intentionally left empty

1.4 This European Standard applies to safety-related as well as non-safety-related software, including for example:

— 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 6.5.4.16 on pre-existing software and for tools in 6.7 are fulfilled.

1.6 Software developed according to a valid version of EN 50128 is considered as compliant to this standard. Software previously developed in accordance with any version of EN 50128 is also considered as compliant and not subject to the requirements on pre-existing software. For SIL1-SIL4 software under the scope of this standard, requirements included in this European Standard are equivalent to the SIL1-SIL4 software requirements of EN 50128:2011.

1.7 This European Standard 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. This European Standard applies to such software. In addition, specific requirements for application data will be given.

1.8 Entry intentionally left empty.

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. However, application of this European Standard during upgrades and maintenance of existing software is recommended.

1.10 The relevant sections of this software standard are also applicable to programmable components (e.g. FPGA and CPLD), in addition to the applicable hardware standard (e.g. EN 50129, EN 50155, EN 61508-2). However, requirements of this software standard that are already covered by the applicable hardware standard do not need to be re-addressed.

When it is possible to exhaustively test the programmable logic for all possible inputs and internal logic states, this European Standard does not apply.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

EN ISO 9000:2015, *Quality management systems — Fundamentals and vocabulary (ISO 9000:2015)*

ISO/IEC 90003:2014, *Software engineering — Guidelines for the application of ISO 9001:2008 to computer software*

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 to form a judgement on whether a product, system or process meet the specified requirements, based on evidence

Note 1 to entry: With regards to software, assessment may include development processes, documentation, system, subsystem hardware and/or software components. Assessment is focused on but not limited to the safety properties of the software being assessed.

3.1.2
Assessor
appointed independent entity that carries out an assessment

Note 1 to entry: This definition is based on EN 50126–1:2017 but in this European Standard, independence of the Assessor is always required, see 5.1.2.

Note 2 to entry: The specific meaning for the term “Assessor” in this standard is defined in 5.1.2.4, 5.1.2.5, 5.1.2.6 and Annex B, Table B.8 in combination herein. This is a software specific role and should not be confused with different types of Assessor specified in other standards.

3.1.3
commercial off-the-shelf (COTS) software
software defined by market-driven need, commercially available and whose fitness for purpose has been deemed acceptable by a broad spectrum of commercial users

[SOURCE: EN 50126-1:2017, 3.10, modified]

3.1.4
component
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.20);

— 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