



**NSAI**  
Standards

Irish Standard  
I.S. EN 50128:2011&AC:2014&A1:2020

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

© CENELEC 2020 No copying without NSAI permission except as permitted by copyright law.

I.S. EN 50128:2011&AC:2014&A1:2020

*Incorporating amendments/corrigenda/National Annexes issued since publication:*

EN 50128:2011/A1:2020

EN 50128:2011/AC:2014

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:*

EN 50128:2011

*Published:*

2011-06-17

*This document was published under the authority of the NSAI and comes into effect on:*

2020-02-24

ICS number:

NOTE: If blank see CEN/CENELEC cover page

NSAI  
1 Swift Square,  
Northwood, Santry  
Dublin 9

T +353 1 807 3800  
F +353 1 807 3838  
E standards@nsai.ie  
W NSAI.ie

Sales:  
T +353 1 857 6730  
F +353 1 857 6729  
W standards.ie

Údarás um Chaighdeáin Náisiúnta na hÉireann

## National Foreword

I.S. EN 50128:2011&AC:2014&A1:2020 is the adopted Irish version of the European Document EN 50128:2011, Railway applications - Communication, signalling and processing systems - Software for railway control and protection systems

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

**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 is a free 20 page sample. Access the full version online.

This page is intentionally left blank

EUROPEAN STANDARD

**EN 50128:2011/A1**

NORME EUROPÉENNE

EUROPÄISCHE NORM

February 2020

ICS 35.240.60; 45.020; 93.100

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 amendment A1 modifies the European Standard EN 50128:2011; it was approved by CENELEC on 2019-07-23. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this amendment 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 amendment 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, Turkey 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**

## European foreword

This document (EN 50128:2011/A1:2020) has been prepared by CLC/SC 9XA "Communication, signalling and processing systems".

The following dates are fixed:

- latest date by which this document (dop) 2020-08-07 has to be implemented at national level by publication of an identical national standard or by endorsement
- latest date by which the national (dow) 2020-08-07 standards conflicting with this document have to be withdrawn

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.

**1 Addition of the following Annex ZZ****Annex ZZ**  
(informative)**Relationship between this European standard and the essential requirements of EU Directive 2016/797/EU [2016 OJ L138] aimed to be covered**

This European Standard has been prepared under a Commission's standardization request relating to harmonized standards in the field of the Interoperability of the rail system within the European Union, M/483, to provide one voluntary means of conforming to essential requirements of Directive 2016/797/EU of the European Parliament and of the Council of 11 May 2016 on the harmonization of the laws of the Member States relating to the interoperability of the rail system within the European Union [2016 OJ L138].

Once this standard is cited in the Official Journal of the European Union under that Directive, compliance with the normative clauses of this standard given in Table ZZ.1 for "Control-Command and Signalling" confers, within the limits of the scope of this standard, a presumption of conformity with the corresponding essential requirements of that Directive, and associated EFTA regulations.

**Table ZZ.1 - Correspondence between this European Standard, the TSI "Control-Command and Signalling" (REGULATION (EU) No 2016/919 of 27 May 2016) and Directive 2016/797/EU [2016 OJ L138]**

<b>Essential Requirements of Directive 2016/797/EU</b>	<b>Chapter / § / points / of CCS TSI</b>	<b>Clause(s) / sub-clause(s) of this EN</b>	<b>Remarks / Notes</b>
1. General Requirements	4.2. Functional and technical specifications of the Subsystems (software aspects)	Clause 4 Clause 5 Clause 6	The TSI includes (in 6.2.3) a reference to the standard EN 50128:2001 or 2011
1.1 Safety	4.5.1. Responsibility of the manufacturer of equipment (software aspects)	Clause 7	
1.1.1		Clause 8	References in the TSI Annex A Table A3 should be updated
2 Requirements specific to each subsystem	5.3 Constituents' performance and specifications (software aspects)	Clause 9	
2.3 Control-Command and Signalling		Annex A Annex B	
2.3.1 Safety	6.2.3 Assessment requirements		
2.3.2 Technical compatibility			

EN 50128:2011/A1:2020 (E)

**WARNING 1** — Presumption of conformity stays valid only as long as a reference to this European standard is maintained in the list published in the Official Journal of the European Union. Users of this standard should consult frequently the latest list published in the Official Journal of the European Union.

**WARNING 2** — Other Union legislation may be applicable to the product(s) falling within the scope of this standard.





Corrigendum to EN 50128:2011

English version

---

Following CLC/BT Decision D146/028, please extend in the Foreword the DOW of EN 50128:2011 from DOR + 36 months to DOR + 72 months, to read 2017-04-25.

---

February 2014

This is a free 20 page sample. Access the full version online.

This page is intentionally left blank

EUROPEAN STANDARD  
NORME EUROPÉENNE  
EUROPÄISCHE NORM

**EN 50128**

June 2011

ICS 35.240.60; 45.020; 93.100

Supersedes 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 European Standard was approved by CENELEC on 2011-04-25. 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, 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, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and the United Kingdom.

**CENELEC**

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

**Management Centre: Avenue Marnix 17, B - 1000 Brussels**

## Contents

Foreword .....	6
Introduction .....	7
1 Scope .....	10
2 Normative references .....	11
3 Terms, definitions and abbreviations .....	11
3.1 Terms and definitions.....	11
3.2 Abbreviations .....	15
4 Objectives, conformance and software safety integrity levels .....	16
5 Software management and organisation.....	17
5.1 Organisation, roles and responsibilities .....	17
5.2 Personnel competence.....	20
5.3 Lifecycle issues and documentation .....	21
6 Software assurance .....	23
6.1 Software testing .....	23
6.2 Software verification.....	25
6.3 Software validation .....	27
6.4 Software assessment .....	28
6.5 Software quality assurance.....	30
6.6 Modification and change control.....	33
6.7 Support tools and languages .....	34
7 Generic software development.....	37
7.1 Lifecycle and documentation for generic software .....	37
7.2 Software requirements .....	37
7.3 Architecture and Design.....	40
7.4 Component design .....	46
7.5 Component implementation and testing .....	49
7.6 Integration.....	50
7.7 Overall Software Testing / Final Validation .....	52
8 Development of application data or algorithms: systems configured by application data or algorithms.....	54

<b>8.1 Objectives</b> .....	<b>54</b>
<b>8.2 Input documents</b> .....	<b>55</b>
<b>8.3 Output documents</b> .....	<b>55</b>
<b>8.4 Requirements</b> .....	<b>55</b>
<b>9 Software deployment and maintenance</b> .....	<b>60</b>
<b>9.1 Software deployment</b> .....	<b>60</b>
<b>9.2 Software maintenance</b> .....	<b>62</b>
<b>Annex A (normative) Criteria for the Selection of Techniques and Measures</b> .....	<b>65</b>
A.1 Clauses tables .....	66
A.2 Detailed tables .....	73
<b>Annex B (normative) Key software roles and responsibilities</b> .....	<b>79</b>
<b>Annex C (informative) Documents Control Summary</b> .....	<b>88</b>
<b>Annex D (informative) Bibliography of techniques</b> .....	<b>90</b>
D.1 Artificial Intelligence Fault Correction.....	90
D.2 Analysable Programs .....	90
D.3 Avalanche/Stress Testing .....	91
D.4 Boundary Value Analysis .....	91
D.5 Backward Recovery .....	92
D.6 Cause Consequence Diagrams.....	92
D.7 Checklists .....	92
D.8 Control Flow Analysis.....	93
D.9 Common Cause Failure Analysis .....	93
D.10 Data Flow Analysis.....	94
D.11 Data Flow Diagrams .....	94
D.12 Data Recording and Analysis.....	95
D.13 Decision Tables (Truth Tables).....	95
D.14 Defensive Programming .....	96
D.15 Coding Standards and Style Guide.....	96
D.16 Diverse Programming .....	97
D.17 Dynamic Reconfiguration.....	98
D.18 Equivalence Classes and Input Partition Testing.....	98
D.19 Error Detecting and Correcting Codes.....	98
D.20 Error Guessing.....	99
D.21 Error Seeding.....	99
D.22 Event Tree Analysis .....	99
D.23 Fagan Inspections.....	100
D.24 Failure Assertion Programming .....	100
D.25 SEEA – Software Error Effect Analysis.....	100
D.26 Fault Detection and Diagnosis .....	101
D.27 Finite State Machines/State Transition Diagrams.....	102
D.28 Formal Methods .....	102
D.29 Formal Proof .....	108

D.30	Forward Recovery.....	108
D.31	Graceful Degradation.....	108
D.32	Impact Analysis.....	109
D.33	Information Hiding / Encapsulation .....	109
D.34	Interface Testing .....	110
D.35	Language Subset.....	110
D.36	Memorising Executed Cases .....	110
D.37	Metrics .....	111
D.38	Modular Approach.....	111
D.39	Performance Modelling .....	112
D.40	Performance Requirements.....	112
D.41	Probabilistic Testing.....	113
D.42	Process Simulation .....	113
D.43	Prototyping / Animation .....	114
D.44	Recovery Block .....	114
D.45	Response Timing and Memory Constraints.....	114
D.46	Re-Try Fault Recovery Mechanisms.....	115
D.47	Safety Bag .....	115
D.48	Software Configuration Management .....	115
D.49	Strongly Typed Programming Languages .....	115
D.50	Structure Based Testing .....	116
D.51	Structure Diagrams .....	116
D.52	Structured Methodology.....	117
D.53	Structured Programming.....	117
D.54	Suitable Programming languages.....	118
D.55	Time Petri Nets .....	119
D.56	Walkthroughs / Design Reviews .....	119
D.57	Object Oriented Programming .....	119
D.58	Traceability.....	120
D.59	Metaprogramming.....	121
D.60	Procedural programming .....	121
D.61	Sequential Function Charts.....	121
D.62	Ladder Diagram .....	122
D.63	Functional Block Diagram .....	122
D.64	State Chart or State Diagram .....	122
D.65	Data modelling .....	122
D.66	Control Flow Diagram/Control Flow Graph.....	123
D.67	Sequence diagram.....	124
D.68	Tabular Specification Methods .....	124
D.69	Application specific language.....	124
D.70	UML (Unified Modeling Language) .....	125
D.71	Domain specific languages.....	126
	<b>Bibliography .....</b>	<b>127</b>

**Figures**

Figure 1 – Illustrative Software Route Map .....	9
Figure 2 – Illustration of the preferred organisational structure .....	18
Figure 3 – Illustrative Development Lifecycle 1 .....	22
Figure 4 – Illustrative Development Lifecycle 2 .....	23

**Tables**

Table 1 - Relation between tool class and applicable sub-clauses .....	37
Table A.1– Lifecycle Issues and Documentation (5.3) .....	66
Table A.2 – Software Requirements Specification (7.2).....	68
Table A.3 – Software Architecture (7.3).....	69
Table A.4– Software Design and Implementation (7.4).....	70
Table A.5 – Verification and Testing (6.2 and 7.3) .....	71
Table A.6 – Integration (7.6).....	71
Table A.7 – Overall Software Testing (6.2 and 7.7).....	71
Table A.8 – Software Analysis Techniques (6.3).....	72
Table A.9 – Software Quality Assurance (6.5).....	72
Table A.10 – Software Maintenance (9.2) .....	72
Table A.11 – Data Preparation Techniques (8.4) .....	73
Table A.12 – Coding Standards.....	73
Table A.13 – Dynamic Analysis and Testing .....	74
Table A.14 – Functional/Black Box Test.....	74
Table A.15 – Textual Programming Languages .....	75
Table A.16 – Diagrammatic Languages for Application Algorithms .....	75
Table A.17 – Modelling .....	76
Table A.18 – Performance Testing.....	76
Table A.19 – Static Analysis .....	76
Table A.20 – Components .....	77
Table A.21 – Test Coverage for Code .....	77
Table A.22 – Object Oriented Software Architecture.....	78
Table A.23 – Object Oriented Detailed Design.....	78
Table B.1 – Requirements Manager Role Specification.....	79
Table B.2 – Designer Role Specification .....	80
Table B.3 – Implementer Role Specification.....	81
Table B.4 – Tester Role Specification .....	82
Table B.5 – Verifier Role Specification .....	83
Table B.6 – Integrator Role Specification .....	84
Table B.7 – Validator Role Specification.....	85
Table B.8 – Assessor Role Specification.....	86
Table B.9 – Project Manager Role Specification .....	87
Table B.10 – Configuration Manager Role Specification .....	87
Table C.1 – Documents Control Summary.....	88

## 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. .

It was submitted to the Formal Vote and was approved by CENELEC as EN 50128 on 2011-04-25.

This document supersedes EN 50128:2001.

The main changes with respect to EN 50128:2001 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 61508-2:2010;
- tables in Annex A have been updated.

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

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) | 2012-04-25 |
| – latest date by which the national standards conflicting with the EN have to be withdrawn   | (dow) | 2014-04-25 |

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*".

---



## 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 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 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 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 functions and social and economic factors.

It is 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 acceptance 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.

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 (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 software safety integrity levels 0-4. 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.

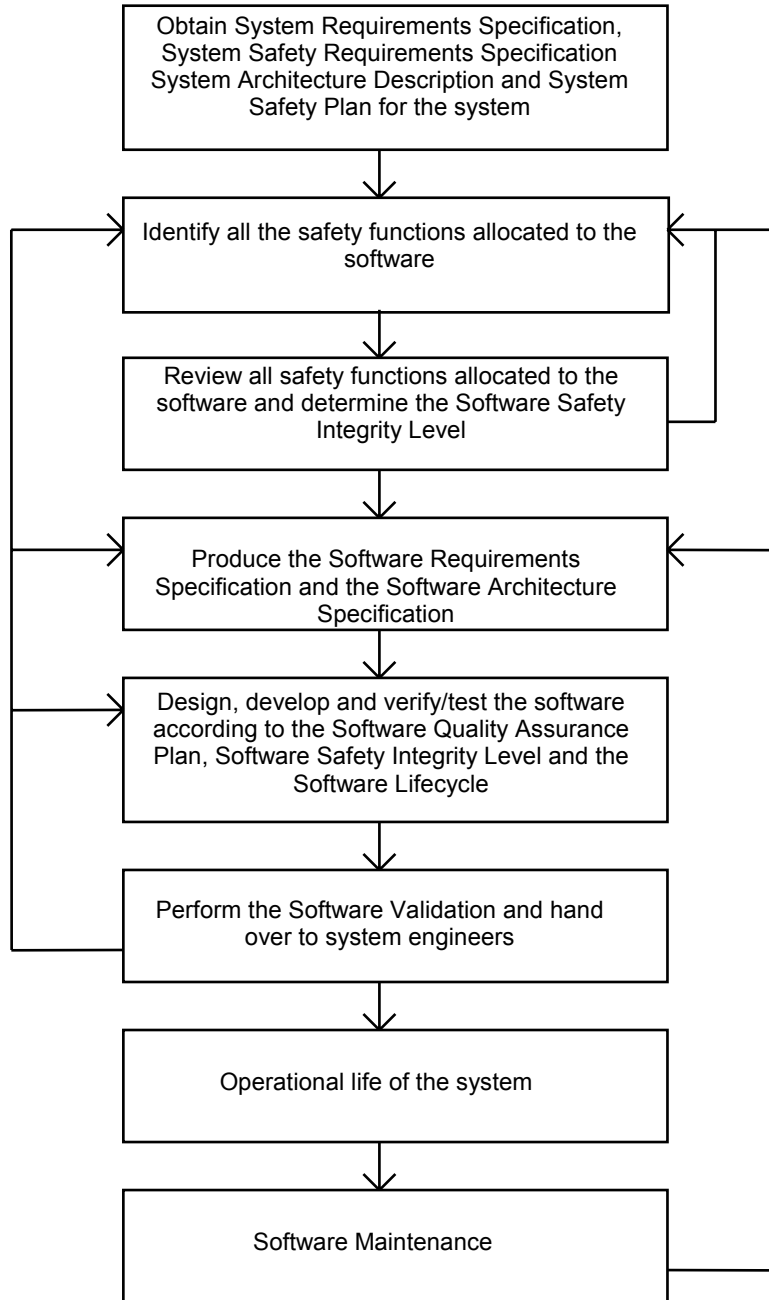


Figure 1 – Illustrative Software Route Map

## 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 any 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 6.5.4.16 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 Clauses 1 to 6 and 9 of this European Standard apply to generic software as well as for application data or algorithms. The specific Clause 7 applies only for generic software while Clause 8 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 assessor has to analyse the evidences provided in the software documentation to confirm whether the determination of the nature and scope of software changes is adequate. However, application of this European Standard during upgrades and maintenance of existing software is highly recommended.

This is a free preview. Purchase the entire publication at the link below:

**I.S. EN 50128 : 2011 : INC : AMD 1 : 2020 : EN :  
COMBINED PDF**

- 
- ⤵ Looking for additional Standards? Visit SAI Global Infostore
  - ⤵ Learn about LexConnect, All Jurisdictions, Standards referenced in Australian legislation
- 

Need to speak with a Customer Service Representative - Contact Us