



Institut luxembourgeois de la normalisation
de l'accréditation, de la sécurité et qualité
des produits et services

ILNAS-EN 50128:2011

Bahnanwendungen - Telekommunikationstechnik, Signaltechnik und Datenverarbeitungssysteme -

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

06/2011



Nationales Vorwort

Diese Europäische Norm EN 50128:2011 wurde als luxemburgische Norm ILNAS-EN 50128:2011 übernommen.

Alle interessierten Personen, welche Mitglied einer luxemburgischen Organisation sind, können sich kostenlos an der Entwicklung von luxemburgischen (ILNAS), europäischen (CEN, CENELEC) und internationalen (ISO, IEC) Normen beteiligen:

- Inhalt der Normen beeinflussen und mitgestalten
- Künftige Entwicklungen vorhersehen
- An Sitzungen der technischen Komitees teilnehmen

<https://portail-qualite.public.lu/fr/normes-normalisation/participer-normalisation.html>

DIESES WERK IST URHEBERRECHTLICH GESCHÜTZT

Kein Teil dieser Veröffentlichung darf ohne schriftliche Einwilligung weder vervielfältigt noch in sonstiger Weise genutzt werden - sei es elektronisch, mechanisch, durch Fotokopien oder auf andere Art!

ILNAS-EN 50128:2011
EUROPÄISCHE NORM **EN 50128**
EUROPEAN STANDARD
NORME EUROPÉENNE Juni 2011

ICS 35.240.60; 45.020; 93.100

Ersatz für EN 50128:2001

Deutsche Fassung

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

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

Diese Europäische Norm wurde von CENELEC am 2011-04-25 angenommen. CENELEC-Mitglieder sind gehalten, die CEN/CENELEC-Geschäftsordnung zu erfüllen, in der die Bedingungen festgelegt sind, unter denen dieser Europäischen Norm ohne jede Änderung der Status einer nationalen Norm zu geben ist.

Auf dem letzten Stand befindliche Listen dieser nationalen Normen mit ihren bibliographischen Angaben sind beim CEN-CENELEC Management Centre oder bei jedem CENELEC-Mitglied auf Anfrage erhältlich.

Diese Europäische Norm besteht in drei offiziellen Fassungen (Deutsch, Englisch, Französisch). Eine Fassung in einer anderen Sprache, die von einem CENELEC-Mitglied in eigener Verantwortung durch Übersetzung in seine Landessprache gemacht und dem CEN-CENELEC Management Centre mitgeteilt worden ist, hat den gleichen Status wie die offiziellen Fassungen.

CENELEC-Mitglieder sind die nationalen elektrotechnischen Komitees von Belgien, Bulgarien, Dänemark, Deutschland, Estland, Finnland, Frankreich, Griechenland, Irland, Island, Italien, Kroatien, Lettland, Litauen, Luxemburg, Malta, den Niederlanden, Norwegen, Österreich, Polen, Portugal, Rumänien, Schweden, der Schweiz, der Slowakei, Slowenien, Spanien, der Tschechischen Republik, Ungarn, dem Vereinigten Königreich und Zypern.

CENELEC

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

Management Centre: Avenue Marnix 17, B - 1000 Brüssel

Inhalt

	Seite
Vorwort.....	- 7 -
Einleitung	- 8 -
1 Anwendungsbereich	- 11 -
2 Normative Verweisungen	- 12 -
3 Begriffe und Abkürzungen	- 12 -
3.1 Begriffe	- 12 -
3.2 Abkürzungen	- 17 -
4 Ziele, Konformität und Software-Sicherheits-Integritätslevel	- 17 -
5 Softwaremanagement und -organisation	- 19 -
5.1 Organisation, Rollen und Verantwortlichkeiten	- 19 -
5.2 Kompetenz der Mitarbeiter	- 22 -
5.3 Fragen des Lebenszyklus und Dokumentation	- 23 -
6 Software-Sicherung	- 26 -
6.1 Softwaretests	- 26 -
6.2 Software-Verifikation	- 27 -
6.3 Software-Validierung	- 29 -
6.4 Software-Begutachtung	- 31 -
6.5 Software-Qualitätssicherung	- 33 -
6.6 Änderungen und Änderungsmanagement	- 36 -
6.7 Unterstützende Werkzeuge und Sprachen	- 37 -
7 Entwicklung generischer Software	- 40 -
7.1 Lebenszyklus und Dokumentation für generische Software	- 40 -
7.2 Software-Anforderungen	- 40 -
7.3 Architektur und Entwurf	- 43 -
7.4 Komponentenentwurf	- 49 -
7.5 Implementierung und Test der Komponenten	- 51 -
7.6 Integration	- 53 -
7.7 Test der Gesamtsoftware/Abschließende Validierung	- 54 -
8 Entwicklung der Anwendungsdaten oder -algorithmen – Systeme, die durch Anwendungsdaten oder -algorithmen konfiguriert werden.....	- 56 -
8.1 Ziele	- 56 -
8.2 Eingangsdokumente	- 57 -
8.3 Ausgangsdokumente	- 57 -
8.4 Anforderungen	- 58 -
9 Bereitstellung und Wartung der Software.....	- 62 -
9.1 Bereitstellung der Software	- 62 -
9.2 Wartung der Software	- 64 -
Anhang A (normativ) Kriterien für die Auswahl der Techniken und Maßnahmen	- 68 -

	Seite
A.1 Tabellen zu den Abschnitten	- 69 -
A.2 Detaillierte Tabellen	- 77 -
Anhang B (normativ) Software-Schlüsselrollen und Verantwortlichkeiten	- 83 -
Anhang C (informativ) Zusammenfassung der Dokumentenkontrolle	- 92 -
Anhang D (informativ) Verfahrensübersicht	- 94 -
D.1 KI(Künstliche-Intelligenz)-Fehlerkorrektur (en: AI Fault Correction)	- 94 -
D.2 Analysierbare Programme	- 94 -
D.3 Avalanche-/Belastungstests (en: Avalanche/Stress Testing)	- 95 -
D.4 Grenzwertanalyse (en: Boundary Value Analysis).....	- 95 -
D.5 Rückwärts-Regeneration (en: Backward Recovery).....	- 96 -
D.6 Ursache-Wirkungsdiagramme (en: Cause Consequence Diagrams).....	- 96 -
D.7 Checklisten (en: Checklists).....	- 96 -
D.8 Steuerflussanalyse (en: Control Flow Analysis).....	- 97 -
D.9 Analyse gemeinsamer Fehler (en: Common Cause Failure Analysis)	- 97 -
D.10 Datenflussanalyse (en: Data Flow Analysis).....	- 98 -
D.11 Datenflussdiagramme (en: Data Flow Diagrams).....	- 98 -
D.12 Datenaufzeichnung und -analyse (en: Data Recording and Analysis)	- 99 -
D.13 Entscheidungstabellen (Wahrheitstabellen) (en: Decision Tables (Truth Tables)).....	- 99 -
D.14 Defensive Programmierung (en: Defensive Programming)	- 100 -
D.15 Codierstandards und Anleitung zum Programmierstil (en: Coding Standards and Style Guide).....	- 101 -
D.16 Diversitäre Programmierung (en: Diverse Programming).....	- 101 -
D.17 Dynamische Rekonfiguration (en: Dynamic Reconfiguration)	- 102 -
D.18 Tests auf Basis von Äquivalenzklassen und Eingangsdaten-Unterteilung (en: Equivalence Classes and Input Partitioning Testing)	- 102 -
D.19 Fehlererkennende und -korrigierende Codes (en: Error Detecting and Correcting Codes)	- 103 -
D.20 Fehlererwartung (en: Error Guessing)	- 103 -
D.21 Fehlereinstreuung (en: Error Seeding)	- 103 -
D.22 Ereignisbaumanalyse (en: Event Tree Analysis)	- 104 -
D.23 Fagan-Inspektionen (en: Fagan Inspections)	- 104 -
D.24 „Failure Assertion“-Programmierung (en: Failure Assertion Programming)	- 104 -
D.25 SEEA – Softwarefehler-Auswirkungsanalyse (en: Software Error Effect Analysis).....	- 105 -
D.26 Fehlererkennung und Diagnose (en: Fault Detection and Diagnosis).....	- 105 -
D.27 Finite-Zustandsmaschinen (FSM)/Zustands-Übergangsdiagramme (en: Finite State Machines/State Transition Diagrams).....	- 106 -
D.28 Formale Methoden (en: Formal Methods)	- 107 -
D.29 Formaler Beweis (en: Formal Proof).....	- 112 -
D.30 Vorwärts-Regeneration (en: Forward Recovery)	- 112 -
D.31 Abgestufte Funktionseinschränkungen (en: Graceful Degradation)	- 113 -

	Seite
D.32 Auswirkungsanalyse (en: Impact Analysis)	- 113 -
D.33 Information-Hiding/Einkapselung (en: Information Hiding/Encapsulation)	- 113 -
D.34 Schnittstellentests (en: Interface Testing)	- 114 -
D.35 Untermenge der Programmiersprache (en: Language Subset)	- 114 -
D.36 Aufzeichnung ausgeführter Fälle (en: Memorising Executed Cases)	- 115 -
D.37 Metriken (en: Metrics)	- 115 -
D.38 Modularer Ansatz (en: Modular Approach)	- 116 -
D.39 Leistungs-Modellierung (en: Performance Modelling)	- 116 -
D.40 Leistungsanforderungen (en: Performance Requirements)	- 117 -
D.41 Wahrscheinlichkeits-Tests (en: Probabilistic Testing)	- 117 -
D.42 Prozesssimulation (en: Process Simulation)	- 118 -
D.43 Prototyping/Animation	- 118 -
D.44 Recovery Block	- 119 -
D.45 Antwortzeiten und Speichergrenzen (en: Response Timing and Memory Constraints)	- 119 -
D.46 „Re-Try Fault Recovery“-Mechanismen (en: Re-Try Fault Recovery Mechanisms)	- 119 -
D.47 Externe Überwachungseinrichtung (en: Safety Bag)	- 120 -
D.48 Software-Konfigurationsmanagement (en: Software Configuration Management)	- 120 -
D.49 Streng typisierte Programmiersprache (en: Strongly Typed Programming Languages)	- 120 -
D.50 Strukturabhängige Tests (en: Structure Based Testing)	- 121 -
D.51 Strukturdiagramme (en: Structure Diagrams)	- 121 -
D.52 Strukturierte Methodik (en: Structured Methodology)	- 122 -
D.53 Strukturierte Programmierung (en: Structured Programming)	- 122 -
D.54 Geeignete Programmiersprachen (en: Suitable Programming Languages)	- 123 -
D.55 Zeit-Petri-Netze (en: Time Petri Nets)	- 124 -
D.56 Walkthroughs/Entwurfsüberprüfungen (en: Walkthroughs/Design Reviews)	- 124 -
D.57 Objektorientierte Programmierung (en: Object Oriented Programming)	- 124 -
D.58 Rückverfolgbarkeit (en: Traceability)	- 125 -
D.59 Metaprogrammierung (en: Metaprogramming)	- 126 -
D.60 Prozedurale Programmierung (en: Procedural programming)	- 126 -
D.61 Sequentielle Funktionslisten (en: Sequential Function Charts – SFC)	- 126 -
D.62 Kontaktplan (en: Ladder Diagram)	- 127 -
D.63 Funktionsblockdiagramm (en: Functional Block Diagram)	- 127 -
D.64 Zustandsliste oder Zustandsdiagramm (en: State Chart or State Diagram)	- 127 -
D.65 Datenmodellierung (en: Data modelling)	- 127 -
D.66 Kontrollflussdiagramm/Kontrollflussgraph (en: Control Flow Diagram/Control Flow Graph)	- 128 -
D.67 Ablaufdiagramm (en: Sequence diagram)	- 129 -
D.68 Tabellarische Spezifikationsverfahren (en: Tabular Specification Methods)	- 129 -

	Seite
D.69 Anwendungsspezifische Sprache (en: Application specific language).....	- 130 -
D.70 UML (Unified Modelling Language).....	- 130 -
D.71 Domänenspezifische Sprachen (en: Domain specific languages).....	- 131 -
Literaturhinweise	- 132 -
Bilder	
Bild 1 – Software, Übersicht über das Vorgehen	- 10 -
Bild 2 – Darstellung der bevorzugten Organisationsstruktur.....	- 20 -
Bild 3 – Beispielhafter Entwicklungs-Lebenszyklus 1	- 25 -
Bild 4 – Beispielhafter Entwicklungs-Lebenszyklus 2	- 26 -
Tabellen	
Tabelle 1 – Beziehung zwischen Werkzeugklasse und anwendbarem Abschnitt	- 40 -
Tabelle A.1 – Fragen des Lebenszyklus und der Dokumentation (5.3)	- 69 -
Tabelle A.2 – Software-Anforderungsspezifikation (7.2).....	- 71 -
Tabelle A.3 – Software-Architektur (7.3)	- 72 -
Tabelle A.4 – Software-Entwurf und -Implementierung (7.4)	- 73 -
Tabelle A.5 – Verifikation und Testen (6.2 und 7.3).....	- 74 -
Tabelle A.6 – Integration (7.6).....	- 74 -
Tabelle A.7 – Testen der Gesamtsoftware (6.2 und 7.7)	- 75 -
Tabelle A.8 – Software-Analysetechniken (6.3)	- 75 -
Tabelle A.9 – Software-Qualitätssicherung (6.5)	- 76 -
Tabelle A.10 – Software-Wartung (9.2).....	- 76 -
Tabelle A.11 – Techniken für die Datengenerierung (8.4)	- 76 -
Tabelle A.12 – Codierstandards.....	- 77 -
Tabelle A.13 – Dynamische Analyse und Testen	- 77 -
Tabelle A.14 – Funktions-/Black-Box-Tests	- 78 -
Tabelle A.15 – Text-Programmiersprachen	- 78 -
Tabelle A.16 – Diagrammartige Sprachen für Anwendungsalgorithmen	- 79 -
Tabelle A.17 – Modellierung.....	- 79 -
Tabelle A.18 – Leistungstests	- 79 -
Tabelle A.19 – Statische Analyse.....	- 80 -
Tabelle A.20 – Komponenten	- 80 -
Tabelle A.21 – Testabdeckung für Code.....	- 81 -
Tabelle A.22 – Objektorientierte Software-Architektur	- 82 -
Tabelle A.23 – Objektorientierter detaillierter Entwurf.....	- 82 -
Tabelle B.1 – Spezifikation der Rolle des Anforderungsmanagers.....	- 83 -
Tabelle B.2 – Spezifikation der Rolle des Entwerfers	- 84 -
Tabelle B.3 – Spezifikation der Rolle des Implementierers.....	- 85 -
Tabelle B.4 – Spezifikation der Rolle des Testers	- 86 -
Tabelle B.5 – Spezifikation der Rolle des Verifizierers	- 87 -

	Seite
Tabelle B.6 – Spezifikation der Rolle des Integrators.....	- 88 -
Tabelle B.7 – Spezifikation der Rolle des Validierers.....	- 89 -
Tabelle B.8 – Spezifikation der Rolle des Gutachters	- 90 -
Tabelle B.9 – Spezifikation der Rolle des Projektmanagers	- 91 -
Tabelle B.10 – Spezifikation der Rolle des Konfigurationsmanagers.....	- 91 -
Tabelle C.1 – Zusammenfassung der Dokumentenkontrolle	- 92 -

Vorwort

Diese Europäische Norm wurde vom SC 9XA „Kommunikation, Signaltechnik und Datenverarbeitungssysteme“ des Technischen Komitees CENELEC/TC 9X „Elektrische und elektronische Anwendungen für Bahnen“ ausgearbeitet.

Sie wurde der formellen Abstimmung unterworfen und von CENELEC am 2011-04-25 als EN 50128 angenommen.

Dieses Schriftstück ersetzt EN 50128:2001.

Nachfolgend sind die wesentlichen Änderungen gegenüber EN 50128:2001 aufgeführt:

- Anforderungen an Softwaremanagement und -organisation, Festlegung von Rollen und Kompetenzen, Bereitstellung und Wartung wurden ergänzt;
- ein neuer Abschnitt zu Werkzeugen, der auf EN 61508-2:2010 beruht, wurde eingefügt;
- die Tabellen in Anhang A wurden aktualisiert.

Es wird auf die Möglichkeit hingewiesen, dass einige Elemente dieses Dokuments Patentrechte berühren können. CEN und CENELEC sind nicht dafür verantwortlich, einige oder alle diesbezüglichen Patentrechte zu identifizieren.

Nachstehende Daten wurden festgelegt:

- spätestes Datum, zu dem die EN auf nationaler Ebene durch Veröffentlichung einer identischen nationalen Norm oder durch Anerkennung übernommen werden muss (dop): 2012-04-25
- spätestes Datum, zu dem nationale Normen, die der EN entgegenstehen, zurückgezogen werden müssen (dow): 2014-04-25

Diese Europäische Norm sollte in Verbindung mit EN 50126-1:1999 „*Bahnanwendungen – Spezifikation und Nachweis der Zuverlässigkeit, Verfügbarkeit, Instandhaltbarkeit, Sicherheit (RAMS) – Teil 1: Grundlegende Anforderungen und genereller Prozess*“ und EN 50129:2003 „*Bahnanwendungen – Telekommunikationstechnik, Signaltechnik und Datenverarbeitungssysteme – Sicherheitsrelevante elektronische Systeme für Signaltechnik*“ gelesen werden.