OVE EN 50716

Ausgabe: 2024-10-01

Bahnanwendungen – Anforderungen für die Softwareentwicklung

Railway Applications –
Requirements for software development

Applications ferroviaires –
Exigences pour le développement de logiciels

Copyright OVE

Medieninhaber und Hersteller:
OVE Österreichischer Verband für Elektrotechnik

Copyright © OVE – 2024.
Alle Rechte vorbehalten! Nachdruck oder Vervielfältigung, Aufnahme auf oder in sonstige Medien oder Datenträger nur mit Zustimmung gestattet!

OVE Österreichischer Verband für Elektrotechnik
Eschenbachgasse 9, 1010 Wien
E-Mail: verkauf@ove.at
Internet: http://www.ove.at
Webshop: www.ove.at/webshop
Tel.: +43 1 587 63 73

ICS	35.240.60
Ident (IDT) mit	EN 50716:2023
Ersatz für	siehe nationales Vorwort
zuständig	OVE/TK TM Traktion und Motorik

Nationales Vorwort

Diese Europäische Norm EN 50716:2023 hat den Status einer nationalen elektrotechnischen Norm gemäß ETG 1992. Bei ihrer Anwendung ist dieses Nationale Vorwort zu berücksichtigen.

Für den Fall einer undatierten normativen Verweisung (Verweisung auf einen Standard ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils neueste Ausgabe dieses Standards.
Für den Fall einer datierten normativen Verweisung bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe des Standards.

Der Rechtsstatus dieser nationalen elektrotechnischen Norm ist den jeweils geltenden Verordnungen zum Elektrotechnikgesetz zu entnehmen.

Bei mittels Verordnungen zum Elektrotechnikgesetz verbindlich erklärten rein österreichischen elektrotechnischen Normen ist zu beachten:
- Hinweise auf Veröffentlichungen beziehen sich, sofern nicht anders angegeben, auf den Stand zum Zeitpunkt der Herausgabe dieser rein österreichischen elektrotechnischen Norm. Zum Zeitpunkt der Anwendung dieser rein österreichischen elektrotechnischen Norm ist der durch die Verordnungen zum Elektrotechnikgesetz oder gegebenenfalls auf andere Weise festgelegte aktuelle Stand zu berücksichtigen.
- Informative Anhänge und Fußnoten sowie normative Verweise und Hinweise auf Fundstellen in anderen, nicht verbindlichen Texten werden von der Verbindlicherklärung nicht erfasst.

Europäische Normen (EN) von CENELEC werden gemäß den CENELEC-Regeln durch Veröffentlichung eines identen Titels und Textes in das Gesamtwerk der nationalen elektrotechnischen Normen übernommen, wobei der Nummerierung der Zusatz OVE vorangestellt wird.

Erläuterung zum Ersatzvermerk

Gemäß Vorwort zur EN wird das späteste Datum, zu dem nationale (elektrotechnische) Normen, die der vorliegenden Norm entgegenstehen, zurückgezogen werden müssen, mit dow (date of withdrawal) festgelegt. Bis zum Zurückziehungsdatum (dow) 2026-10-30 ist somit die Anwendung folgender Norm(en) noch erlaubt:

ÖVE/ÖNORM EN 50128:2012-04-01,
ÖVE/ÖNORM EN 50128/AC:2014-12-01,
OVE EN 50128 Beiblatt 1:2018-01-01,
OVE EN 50128/A1:2020-09-01,
OVE EN 50128/A2:2021-01-01,
OVE EN 50657:2017-12-01,
OVE EN 50657/A1:2024-05-01.

Copyright OVE

EUROPÄISCHE NORM **EN 50716**

EUROPEAN STANDARD

NORME EUROPÉENNE

November 2023

ICS 35.240.60

Ersatz für 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

Deutsche Fassung

Bahnanwendungen – Anforderungen für die Softwareentwicklung

Railway Applications – Requirements for software development

Applications ferroviaires – Exigences pour le développement de logiciels

Diese Europäische Norm wurde von CENELEC am 2023-10-30 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, der Republik Nordmazedonien, Rumänien, Schweden, der Schweiz, Serbien, der Slowakei, Slowenien, Spanien, der Tschechischen Republik, der Türkei, Ungarn, dem Vereinigten Königreich und Zypern.

Copyright OVE



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

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brüssel

© 2023 CENELEC Alle Rechte der Verwertung, gleich in welcher Form und in welchem Verfahren, sind weltweit den Mitgliedern von CENELEC vorbehalten.

Inhalt

Seite

Europäisches Vorwort ... 15

Einleitung ... 16

1 Anwendungsbereich ... 19

2 Normative Verweisungen ... 19

3 Begriffe und Abkürzungen ... 20

3.1 Begriffe ... 20

3.2 Abkürzungen ... 27

4 Konformität der Software-Integritätslevel ... 27

5 Softwaremanagement und -organisation ... 29

5.1 Organisation und Unabhängigkeit von Rollen ... 29

5.1.1 Ziel ... 29

5.1.2 Anforderungen ... 29

5.2 Kompetenz und Verantwortlichkeiten von Personal ... 31

5.2.1 Ziele ... 31

5.2.2 Anforderungen ... 31

5.3 Fragen des Lebenszyklus und Dokumentation ... 32

5.3.1 Ziele ... 32

5.3.2 Anforderungen ... 32

6 Softwaresicherung ... 33

6.1 Softwaretests ... 33

6.1.1 Ziel ... 33

6.1.2 Eingangsdokumente ... 33

6.1.3 Ausgangsdokumente ... 33

6.1.4 Anforderungen ... 33

6.2 Softwareverifizierung ... 34

6.2.1 Ziel ... 34

6.2.2 Eingangsdokumente ... 34

6.2.3 Ausgangsdokumente ... 35

6.2.4 Anforderungen ... 35

6.3 Softwarevalidierung ... 36

6.3.1 Ziel ... 36

6.3.2 Eingangsdokumente ... 36

6.3.3 Ausgangsdokumente ... 36

6.3.4 Anforderungen ... 37

6.4 Softwarebegutachtung ... 38

6.4.1 Ziel ... 38

6.4.2 Eingangsdokumente ... 38

6.4.3 Ausgangsdokumente ... 38

Copyright OVE

6.4.4 Anforderungen ... 38
6.5 Software-Qualitätssicherung ... 39
6.5.1 Ziele ... 39
6.5.2 Eingangsdokumente ... 40
6.5.3 Ausgangsdokumente ... 40
6.5.4 Anforderungen ... 40
6.6 Änderungen und Änderungsmanagement ... 42
6.6.1 Ziele ... 42
6.6.2 Eingangsdokumente ... 42
6.6.3 Ausgangsdokumente ... 42
6.6.4 Anforderungen ... 42
6.7 Unterstützende Werkzeuge und Sprachen ... 43
6.7.1 Ziele ... 43
6.7.2 Eingangsdokumente ... 43
6.7.3 Ausgangsdokumente ... 43
6.7.4 Anforderungen ... 43
7 Softwareentwicklung ... 46
7.1 Lebenszyklus und Dokumentation für Software ... 46
7.1.1 Ziele ... 46
7.1.2 Anforderungen ... 46
7.2 Softwareanforderungen ... 47
7.2.1 Ziele ... 47
7.2.2 Eingangsdokumente ... 47
7.2.3 Ausgangsdokumente ... 47
7.2.4 Anforderungen ... 47
7.3 Architektur und Entwurf ... 49
7.3.1 Ziele ... 49
7.3.2 Eingangsdokumente ... 49
7.3.3 Ausgangsdokumente ... 50
7.3.4 Anforderungen ... 50
7.4 Komponentenentwurf ... 56
7.4.1 Ziele ... 56
7.4.2 Eingangsdokumente ... 57
7.4.3 Ausgangsdokumente ... 57
7.4.4 Anforderungen ... 57
7.5 Implementierung und Test der Komponenten ... 58
7.5.1 Ziele ... 58
7.5.2 Eingangsdokumente ... 59
7.5.3 Ausgangsdokumente ... 59
7.5.4 Anforderungen ... 59

Copyright OVE

7.6 Integration ... 60
7.6.1 Ziele ... 60
7.6.2 Eingangsdokumente ... 60
7.6.3 Ausgangsdokumente ... 60
7.6.4 Anforderungen ... 60
7.7 Test der Gesamtsoftware/Abschließende Validierung ... 62
7.7.1 Ziele ... 62
7.7.2 Eingangsdokumente ... 62
7.7.3 Ausgangsdokumente ... 62
7.7.4 Anforderungen ... 62
8 Entwicklung von Anwendungsdaten: Systeme, die durch Anwendungsdaten konfiguriert werden ... 64
8.1 Ziele ... 64
8.2 Eingangsdokumente ... 64
8.3 Ausgangsdokumente ... 65
8.4 Anforderungen ... 65
8.4.1 Anwendungsentwicklungsprozess ... 65
8.4.2 Anwendungs-Anforderungsspezifikation ... 66
8.4.3 Architektur und Entwurf ... 67
8.4.4 Generierung von Anwendungsdaten ... 67
8.4.5 Integration der Anwendung und Testen ... 68
8.4.6 Validierung und Begutachtung der Anwendung ... 68
8.4.7 Verfahren und Werkzeuge der Anwendungsgenerierung ... 68
9 Bereitstellung und Wartung der Software ... 69
9.1 Bereitstellung der Software ... 69
9.1.1 Ziel ... 69
9.1.2 Eingangsdokumente ... 69
9.1.3 Ausgangsdokumente ... 69
9.1.4 Anforderungen ... 69
9.2 Softwarewartung ... 71
9.2.1 Ziel ... 71
9.2.2 Eingangsdokumente ... 71
9.2.3 Ausgangsdokumente ... 71
9.2.4 Anforderungen ... 71
Anhang A (normativ) Kriterien für die Auswahl der Techniken und Maßnahmen ... 74
A.1 Allgemeines ... 74
A.2 Abschnittstabellen ... 75
A.3 Detailtabellen ... 81
Anhang B (normativ) Software-Schlüsselrollen und Verantwortlichkeiten ... 86
Anhang C (informativ) Leitfaden für Softwareentwicklung ... 94
C.1 Beispiele für ein Lebenszyklusmodell ... 94

Copyright OVE

C.1.1 Allgemeine Bemerkungen 94
C.1.2 Lineare Lebenszyklusmodelle 94
C.1.3 Iterative Lebenszyklusmodelle 96
C.2 Modellierung 100
C.2.1 Modellierung Allgemeines 100
C.2.2 Modellierung – Definition 100
C.2.3 Modellierung – Fragen bezüglich Lebenszyklus und Dokumentation 101
C.2.4 Modellierung – Softwaresicherung 102
C.2.5 Modellierung – Unterstützende Werkzeuge und Sprachen 102
C.2.6 Modellierung – Softwareentwicklung (Lebenszyklus und Dokumentation) 102
C.3 Künstliche Intelligenz und maschinelles Lernen 106
C.3.1 Allgemeines 106
C.3.2 Anwendung von KI und ML innerhalb von EN 50716 106
C.3.3 Herausforderungen für die Verwendung von KI und ML in EN 50716 107
Anhang D (informativ) Verfahrensübersicht 108
D.1 Bleibt frei 108
D.2 Analysierbare Programme 108
D.3 Avalanche-/Belastungstests 108
D.4 Grenzwertanalyse 109
D.5 Rückwärts-Regeneration 109
D.6 Ursache-Wirkungsdiagramme 109
D.7 Checklisten 110
D.8 Kontrollflussanalyse 110
D.9 Bleibt frei 111
D.10 Datenflussanalyse 111
D.11 Datenflussdiagramme 111
D.12 Datenaufzeichnung und-analyse 112
D.13 Entscheidungstabellen und Wahrheitstabellen 112
D.14 Defensive Programmierung 113
D.15 Codierstandards und Anleitung zum Programmierstil (en: Coding Standards and Style Guide) 113
D.16 Diversitäre Programmierung 115
D.17 Dynamische Rekonfiguration 115
D.18 Tests auf Basis von Äquivalenzklassen und Eingangsdaten-Unterteilung 116
D.19 Fehlererkennende und -korrigierende Codes 116
D.20 Fehlererwartung (en: Error Guessing) 116
D.21 Fehlereinstreuung (en: Error Seeding) 117
D.22 Ereignisbaumanalyse 117
D.23 Fagan-Inspektionen 118
D.24 „Failure Assertion“-Programmierung 118
D.25 Softwarefehler-Auswirkungsanalyse (en: Software Error Effect Analysis, SEEA) 118

Copyright OVE

D.26 Fehlererkennung und Diagnose ... 119
D.27 Endliche Zustandsautomaten/Zustandsübergangsdiagramme ... 120
D.28 Formale Methoden ... 120
D.29 Formaler Beweis ... 121
D.30 Vorwärts-Regeneration ... 121
D.31 Abgestufte Funktionseinschränkungen ... 122
D.32 Auswirkungsanalyse ... 122
D.33 Information-Hiding/Kapselung ... 122
D.34 Schnittstellentests ... 123
D.35 Bleibt frei ... 123
D.36 Aufzeichnung ausgeführter Fälle ... 123
D.37 Metriken ... 124
D.38 Modularer Ansatz ... 124
D.39 Leistungsmodellierung ... 125
D.40 Leistungsanforderungen ... 125
D.41 Bleibt frei ... 126
D.42 Prozesssimulation ... 126
D.43 Prototyping/Animation ... 126
D.44 „Recovery Block“ ... 126
D.45 Antwortzeiten und Speicherbeschränkungen ... 127
D.46 „Re-Try Fault Recovery“-Mechanismen ... 127
D.47 Externe Überwachungseinrichtung ... 127
D.48 Software-Konfigurationsmanagement ... 128
D.49 Bleibt frei ... 128
D.50 Strukturabhängige Tests ... 128
D.51 Strukturdiagramme ... 128
D.52 Strukturierte Methodik ... 129
D.53 Strukturierte Programmierung ... 130
D.54 Geeignete Programmiersprachen ... 130
D.55 Petri-Netze ... 132
D.56 Walkthroughs/Entwurfsüberprüfungen ... 133
D.57 Bleibt frei ... 133
D.58 Rückverfolgbarkeit ... 133
D.59 Bleibt frei ... 134
D.60 Bleibt frei ... 134
D.61 Bleibt frei ... 134
D.62 Bleibt frei ... 134
D.63 Bleibt frei ... 134
D.64 Bleibt frei ... 134
D.65 Datenmodellierung ... 134

Copyright OVE

D.66 Kontrollflussdiagramm/Kontrollflussgraph 134
D.67 Ablaufdiagramm (en: Sequence Diagram) 136
D.68 Tabellarische Spezifikationsverfahren 136
D.69 Bleibt frei 137
D.70 Bleibt frei 137
D.71 Domänenspezifische Sprachen 137
Anhang ZZ (informativ) Zusammenhang zwischen dieser Europäischen Norm und den grundlegenden Anforderungen der abzudeckenden Richtlinie (EU) 2016/797 138
Literaturhinweise 140

Bilder

Bild 1 – Veranschaulichung einer Software-Planungsübersicht 18
Bild 2 – Veranschaulichung der Organisationsstruktur 30
Bild C.1 – Lineares Lebenszyklusmodell, Beispiel 1 (Wasserfallmodell) 95
Bild C.2 – Lineares Lebenszyklusmodell, Beispiel 2 (V-Modell) 96
Bild C.3 – Iteratives Lebenszyklusmodell, Beispiel 1 97
Bild C.4 – Iteratives Lebenszyklusmodell, Beispiel 2 98
Bild C.5 – Beispiel für iterative Entwicklung eines Arbeitsergebnisses 99
Bild C.6 – Iteratives Lebenszyklusmodell, Beispiel 3 99

Tabellen

Tabelle 1 – Zusammenhang zwischen Werkzeugklasse und anwendbaren Unterabschnitten 46
Tabelle A.1 – Fragen des Lebenszyklus und der Dokumentation (5.3) 75
Tabelle A.2 – Software-Anforderungsspezifikation (7.2) 77
Tabelle A.3 – Softwarearchitektur (7.3) 77
Tabelle A.4 – Software-Entwurf und -Implementierung (7.3, 7.4, und 7.5) 78
Tabelle A.5 – Analyse und Testen von Softwarekomponenten (6.2 und 7.4) 79
Tabelle A.6 – Analyse und Testen der Software-Integration (7.3 und 7.6) 79
Tabelle A.7 – Analyse und Testen der Gesamtsoftware (6.2 und 7.2) 79
Tabelle A.8 – Bleibt frei 80
Tabelle A.9 – Software-Qualitätssicherungsplan (6.5) 80
Tabelle A.10 – Softwarewartung (9.2) 80
Tabelle A.11 – Techniken für die Datengenerierung (8.4) 80
Tabelle A.12 – Codierstandards 81
Tabelle A.13 – Dynamische Analyse und Testen 81
Tabelle A.14 – Bleibt frei 82
Tabelle A.15 – Geeignete Programmiersprachen 82
Tabelle A.16 – Bleibt frei 82
Tabelle A.17 – Modellierung 82
Tabelle A.18 – Leistungstests 83
Tabelle A.19 – Statische Analyse 83

Copyright OVE

Tabelle A.20 – Komponenten..... 84

Tabelle A.21 – Testabdeckung für Codes..... 84

Tabelle B.1 – Spezifikation der Rolle des Anforderungsmanagers..... 86

Tabelle B.2 – Spezifikation der Rolle des Entwerfers..... 87

Tabelle B.3 – Spezifikation der Rolle des Implementierers..... 88

Tabelle B.4 – Spezifikation der Rolle des Testers..... 89

Tabelle B.5 – Spezifikation der Rolle des Verifizierers..... 90

Tabelle B.6 – Spezifikation der Rolle des Validierers..... 91

Tabelle B.7 – Spezifikation der Rolle des Gutachters..... 92

Tabelle B.8 – Spezifikation der Rolle des Projektmanagers..... 93

Tabelle B.9 – Spezifikation der Rolle des Konfigurationsmanagers..... 93

Tabelle C.1 – Typische Anpassung von Architektur und Entwurf für Modellierung..... 103

Tabelle C.2 – Typische Anpassung von Implementierung und Test der Komponenten für Modellierung..... 104

Tabelle C.3 – Typische Anpassung von Techniken/Maßnahmen von Codierstandards für Modellierung..... 105

Tabelle ZZ.1 – Zusammenhang zwischen dieser Europäischen Norm, der Verordnung 2016/919 der Kommission über eine technische Spezifikation für Interoperabilität (TSI) des Subsystems „Zugsteuerung, Zugsicherung und Signalgebung“ des Eisenbahnsystems der Europäischen Union* und Richtlinie (EU) 2016/797..... 138

Tabelle ZZ.2 – Zusammenhang zwischen dieser Europäischen Norm, der Verordnung (EU) Nr. 1302/2014 der Kommission über eine technische Spezifikation für die Interoperabilität (TSI) des Subsystems „Fahrzeuge – Lokomotiven und Personenwagen“ des Eisenbahnsystems in der Europäischen Union* und der Richtlinie(EU) 2016/797..... 139

Copyright OVE

Europäisches Vorwort

Dieses Dokument (EN 50716:2023) wurde vom Technischen Komitee CLC/TC 9X „Elektrische und elektronische Anwendungen für Bahnen“ erarbeitet.

Nachstehende Daten wurden festgelegt:

- spätestes Datum, zu dem dieses Dokument auf nationaler Ebene durch Veröffentlichung einer identischen nationalen Norm oder durch Anerkennung übernommen werden muss (dop): 2024-10-30
- spätestes Datum, zu dem nationale Normen, die diesem Dokument entgegenstehen, zurückgezogen werden müssen (dow): 2026-10-30

Dieses Dokument ersetzt EN 50128:2011 und EN 50657:2017 und alle Änderungen und Berichtigungen (falls vorhanden).

EN 50716:2023 enthält die folgenden wesentlichen technischen Änderungen gegenüber EN 50128:2011 und EN 50657:2017:

- es erfolgte eine verbesserte Anpassung an EN 50126-1:2017 und EN 50126-2:2017, einschließlich Definitionen;
- Abschnitt 5: Anforderungen wurden umformuliert, um die Lesbarkeit zu verbessern (während vorhandene organisatorische Auswahlmöglichkeiten weitestgehend beibehalten werden);
- Anhang A wurde aktualisiert, um eine bessere Anpassung an Lebenszyklusphasen zu erreichen;
- im informativen Anhang C, wurde ein neuer Abschnitt C.1 mit zusätzlichen Hinweisen zu Lebenszyklusmodellen hinzugefügt;
- im informativen Anhang C, wurde ein neuer Abschnitt C.2 mit zusätzlichen Hinweisen zur Modellierung für Softwareentwicklung hinzugefügt;
- ein zusätzlicher Leitfaden für Softwarekomponenten unterschiedlicher Software-Integritätslevel wurde aufgenommen;
- Anforderungen an Programmiersprachen wurden allgemeiner gefasst.

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

Dieses Dokument ist in Verbindung mit EN 50126-1 “Bahnanwendungen - Spezifikation und Nachweis von Zuverlässigkeit, Verfügbarkeit, Instandhaltbarkeit und Sicherheit (RAMS) - Teil 1: Generischer RAMS-Prozess” [1] und EN 50126-2 “Bahnanwendungen - Spezifikation und Nachweis von Zuverlässigkeit, Verfügbarkeit, Instandhaltbarkeit und Sicherheit (RAMS) - Teil 2: Systembezogene Sicherheitsmethodik” [2] zu lesen.

Für Ortsfeste Anlagen (Leistungsregelung und Stromversorgung des elektrischen Zugbetriebs) ist EN 50562 “Bahnanwendungen - Ortsfeste Anlagen - Prozess, Schutzmaßnahmen und Nachweisführung für die Sicherheit für elektrische Bahnanlagen” [20] anzuwenden.

Dieses Dokument wurde im Rahmen eines Normungsauftrages erarbeitet, der von der Europäischen Kommission und der Europäischen Freihandelszone an CENELEC gegeben wurde. Diese Normungsaufträge werden anschließend durch das Ständige Komitee der EFTA-Länder für diese Mitgliedsländer angenommen.

Zum Zusammenhang mit EU-Richtlinien siehe den informativen Anhang ZZ, der Bestandteil dieses Dokuments ist.

Rückmeldungen oder Fragen zu diesem Dokument sollten an das jeweilige nationale Komitee des Anwenders gerichtet werden. Eine vollständige Liste dieser Gremien ist auf den Internetseiten des CENELEC abrufbar.

Copyright OVE

Einleitung

Dieses Dokument konzentriert sich auf Verfahren, die zu verwenden sind, um Software zu erhalten welche den Anspruch an die Software-Integrität erfüllt.

Dieses Dokument beschreibt eine Reihe von Anforderungen für Entwicklung, Bereitstellung und Wartung beliebiger Software, die für Bahnanwendungen vorgesehen ist. Es werden Anforderungen hinsichtlich der Organisationsstruktur, der Beziehung zwischen Organisationen und der Aufteilung von Verantwortlichkeiten derjenigen festgelegt, die in Entwicklungs-, Bereitstellungs- und Wartungsmaßnahmen eingebunden sind. Gleichermaßen werden in diesem Dokument Kriterien für die Qualifikation und für die Fachkenntnis des Personals angegeben.

Das Grundkonzept dieses Dokuments basiert auf Software-Integritätsleveln. Dieses Dokument behandelt fünf Software-Integritätslevel, von denen die Basisintegrität der niedrigste und 4 der höchste Level ist. Je größer das Risiko ist, das von einem Softwareausfall ausgeht, desto höher ist der Software-Integritätslevel.

ANMERKUNG 1 Der in diesem Dokument verwendete Begriff der Basisintegrität (en: basic integrity) wurde erstmals in der Normenreihe EN 50126 ([1] [2]) eingeführt.

Dieses Dokument führt Techniken und Maßnahmen für fünf Software-Integritätslevel auf. Die geforderten Techniken und Maßnahmen für die Basisintegrität und für die Sicherheits-Integritätslevel 1 bis 4 werden in den Tabellen im normativen Anhang A angegeben. Die geforderten Techniken für Level 1 sind dieselben wie für Level 2 und die geforderten Techniken für Level 3 dieselben wie für Level 4. Dieses Dokument gibt keine Hinweise darauf, welcher Software-Integritätslevel für ein gegebenes Risiko angemessen ist. Diese Entscheidung hängt von vielen Faktoren ab, wie der Art der Anwendung, dem Ausmaß, in dem andere Systeme sicherheitsrelevante Funktionen übernehmen, und von sozialen und wirtschaftlichen Faktoren.

Es fällt in den Anwendungsbereich von EN 50126-1 und EN 50126-2, das Verfahren zur Festlegung von der Software zugewiesenen sicherheitsrelevanten Funktionen zu definieren.

Dieses Dokument legt die Maßnahmen fest, die notwendig sind, um diese Anforderungen zu erfüllen.

Die Normenreihe EN 50126 ([1] [2]) fordert eine systematische Vorgehensweise in Bezug auf:

a) Identifikation von Gefährdungen, Bewertung von Risiken und risikobasierte Entscheidungsfindung;

b) Identifikation der notwendigen Risikominderung, um die Risikoakzeptanzkriterien zu erfüllen;

c) Definition der übergreifenden System-Sicherheitsanforderungen für die Schutzmaßnahmen, die notwendig sind, um die erforderliche Risikominderung zu erreichen;

d) Auswahl einer passenden Systemarchitektur;

e) Planung, Überwachung und Steuerung der technischen Aktivitäten und der Managementaktivitäten, die notwendig sind, um die System-Sicherheitsanforderungen in ein sicherheitsrelevantes System mit einem validierten Sicherheits-Integritätslevel umzusetzen.

Mit der Aufteilung der Spezifikation in einen Entwurf sicherheitsrelevanter Systeme und Komponenten erfolgt eine weitergehende Zuordnung von Sicherheits-Integritätsleveln. Schließlich führt dies zu den erforderlichen Software-Integritätsleveln.

Der aktuelle Stand der Technik ist der, dass weder die Anwendung von Qualitätssicherungsverfahren (so genannte fehlervermeidende Maßnahmen und fehlererkennende Maßnahmen) noch die Anwendung fehlertoleranter Softwareverfahren die absolute Sicherheit der Software sicherstellen können. Es ist kein Weg bekannt, die Fehlerfreiheit in einer vergleichsweise komplexen sicherheitsrelevanten Software zu beweisen. Dies gilt insbesondere für Spezifikations- und Entwurfsfehler.

Die bei der Entwicklung von Software mit hoher Integrität anzuwendenden Prinzipien beinhalten, sind aber nicht beschränkt auf:

— Top-Down-Entwurfsverfahren;

— Modularität;

Copyright OVE

— Verifizierung jeder Phase des Entwicklungslebenszyklus;

— verifizierte Komponenten und Komponentenbibliotheken;

— klare Dokumentation und Rückverfolgbarkeit;

— auditierbare Dokumente;

— Validierung;

— Begutachtung;

— Konfigurationsmanagement und Änderungsmanagement;

— geeignete Betrachtung von Fragen der Organisation und der Kompetenz des Personals.

Die Zuweisung von Systemanforderungen zu Softwarefunktionen findet auf Systemebene statt. Dies beinhaltet die Definition der geforderten Integritätslevel für die Softwarefunktionen. Die aufeinander folgenden funktionalen Schritte bei der Anwendung dieses Dokuments sind in Bild 1 dargestellt und sind die folgenden:

f) Definieren der Software-Anforderungsspezifikation und parallel dazu Überlegungen zur Softwarearchitektur anstellen. Mit der Softwarearchitektur wird die Strategie für die Software und den Software-Integritätslevel entwickelt (7.2 und 7.3);

g) Entwerfen, Implementieren und Testen der Software entsprechend dem Software-Qualitätssicherungsplan, dem Software-Integritätslevel und dem Software-Lebenszyklus (7.4 und 7.5);

h) Integrieren der Software auf der Zielhardware und Verifizierung der Funktionalität (7.6);

i) Validieren und Bereitstellen der Software (7.7 und 9.1);

j) Softwarewartung, sofern während der Betriebslebensdauer erforderlich (9.2).

Einige Qualitätssicherungsmaßnahmen erstrecken sich über die gesamte Softwareentwicklung. Diese schließen Tests (6.1), Verifizierung (6.2), Validierung (6.3), Begutachtung (6.4), Qualitätssicherung (6.5) sowie Änderungen und Änderungsmanagement (6.6) ein.

Es werden Anforderungen für unterstützende Werkzeuge (6.7) und für Systeme festgelegt, die durch Anwendungsdaten (Abschnitt 8) konfiguriert werden.

Außerdem werden Anforderungen an die Unabhängigkeit der Rollen und die Kompetenz des an der Softwareentwicklung beteiligten Personals gestellt (5.1, 5.2 und Anhang B).

Dieses Dokument schreibt nicht den Gebrauch eines bestimmten Software-Entwicklungslebenszyklus vor. Ein empfohlener Lebenszyklus und ein Satz von Dokumenten sind jedoch angegeben in 5.3, 7.1 und in C.1 .

Es wurden Tabellen erstellt, die die verschiedenen Techniken/Maßnahmen entsprechend den Software-Integritätsleveln 1 bis 4 und für die Basisintegrität zuordnen. Die Tabellen sind in Anhang A angegeben. Ein Literaturverzeichnis, auf das in den Tabellen verwiesen wird, gibt eine kurze Beschreibung jeder Technik/Maßnahme mit Hinweisen auf weiterführende Informationsquellen. Die Literaturhinweise zu Techniken befinden sich in Anhang D.

ANMERKUNG 2 Einige Einträge in diesem Dokument wurden absichtlich frei gelassen. Damit wird sichergestellt, dass die Nummerierung in Bezug auf EN 50128 und EN 50657 unverändert bleibt, soweit dies in angemessener Weise praktikabel ist und die Lesbarkeit nicht beeinträchtigt. Dies dient der Unterstützung des Umstiegs auf und die Übernahme dieses Dokuments, sofern zutreffend.

Dieses Dokument legt nicht die Anforderungen an die Entwicklung, Implementierung, Wartung und/oder Anwendung von Sicherheitsrichtlinien (en: security policies) oder -diensten (en: security services) fest, die zur Erfüllung der Anforderungen an Cybersicherheit erforderlich sind, die möglicherweise von sicherheitsrelevanten Systemen benötigt werden. Cyberangriffe können nicht nur den Betrieb, sondern auch die funktionale Sicherheit eines Systems beeinflussen. Für Cybersicherheit sollten geeignete Normen angewendet werden.

ANMERKUNG 3 ISO-/IEC-Normen und CEN/CENELEC-Publikationen, welche Cybersicherheit ausführlich behandeln, sind [3], [4], [5] und [17].

Copyright OVE

Es ist möglicherweise notwendig, zwischen Maßnahmen gegen systematische Fehler und Maßnahmen gegen Sicherheitsbedrohungen (en: security threats) abzuwägen. Ein Beispiel dafür ist die Notwendigkeit schneller Sicherheits-Updates von Software bei Sicherheitsbedrohungen (en: security threats), wobei sicherheitsrelevante Software vor jedem Update entwickelt, getestet, validiert und freigegeben werden sollte.

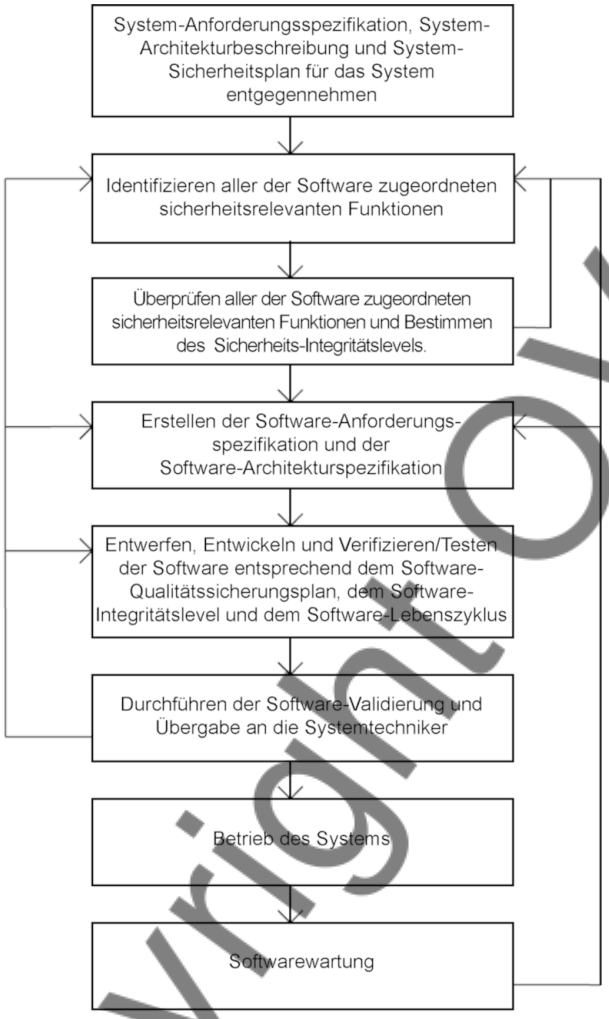


Bild 1 – Veranschaulichung einer Software-Planungsübersicht

Copyright OVE

1 Anwendungsbereich

1.1 Dieses Dokument legt den Prozess und die technischen Anforderungen für die Entwicklung von Software für programmierbare elektronische Systeme fest, zur Verwendung in:

— Anwendungen der Zugsteuerung, Zugsicherung und Signalgebung (en: control, command and signalling);

— Anwendungen auf Bahnfahrzeugen.

Dieses Dokument ist weder für eine Anwendung im Bereich der Stromversorgung zur elektrischen Zugförderung (ortsfeste Anlagen) anzuwenden, noch für Stromversorgung und Steuerung von herkömmlichen Anwendungen, z. B. Stromversorgungen auf Bahnhöfen für Büros, Geschäfte, usw. Diese Anwendungen werden üblicherweise in Normen für Energieverteilung und/oder Bereiche außerhalb von Bahnanwendungen und/oder in lokalen gesetzlichen Rahmenbedingungen abgedeckt.

1.2 Dieses Dokument ist ausschließlich auf Software und die Wechselwirkung zwischen Software und dem System anwendbar, zu dem die Software gehört.

1.3 Bleibt frei

1.4 Dieses Dokument ist anzuwenden auf Software entsprechend 1.1 dieses Dokuments, die in Bahnsystemen verwendet wird, einschließlich:

— Anwendungsprogrammierung;

— Betriebssysteme;

— unterstützende Werkzeuge;

— Firmware.

Anwendungsprogrammierung umfasst Programmierung in Hochsprache, Maschinensprache und speziellen Anwendungssprachen (z. B. ladder logic bei speicherprogrammierbaren Steuerungen).

1.5 Die Nutzung bereits bestehender (en: pre-existing) Software (wie in 3.1.16 definiert) und von Werkzeugen wird in diesem Dokument ebenfalls behandelt. Derartige Software kann eingesetzt werden, wenn die spezifischen Anforderungen in 7.3.4.7 und 6.5.4.16 für bereits bestehende Software und für Werkzeuge in 6.7 erfüllt sind.

1.6 Bleibt frei

1.7 In diesem Dokument wird berücksichtigt, dass moderner Anwendungsentwurf häufig Software verwendet, die als Basis für verschiedene Anwendungen geeignet ist. Diese Software wird anschließend durch Anwendungsdaten konfiguriert, um damit die ausführbare Software für die Anwendung anzufertigen.

1.8 Bleibt frei

1.9 Dieses Dokument ist nicht rückwirkend anwendbar. Es ist daher in erster Linie für Neuentwicklungen und in ihrer Gesamtheit für vorhandene Systeme nur dann anzuwenden, wenn diese größeren Änderungen unterworfen werden. Für kleinere Änderungen ist nur 9.2 anzuwenden. Die Anwendung dieses Dokuments bei der Erweiterung und der Wartung vorhandener Software wird jedoch empfohlen.

1.10 Für die Entwicklung anwenderprogrammierbarer integrierter Schaltungen (z. B. feldprogrammierbare Gate-Arrays (FPGA) und komplexe programmierbare Logikschaltungen (CPLD; en: complex programmable logic devices) ist ein Leitfaden für sicherheitsrelevante Funktionen in EN 50129:2018, Anhang F, und für nicht-sicherheitsrelevante Funktionen in EN 50155:2017 enthalten. Software, die auf Softcore-Prozessoren von anwenderprogrammierbaren integrierten Schaltungen läuft, liegt im Anwendungsbereich dieses Dokuments.

2 Normative Verweisungen

Es gibt keine normativen Verweisungen in diesem Dokument.

Copyright OVE