

	Bahnanwendungen Telekommunikationstechnik, Signaltechnik und Datenverarbeitungssysteme Software für Eisenbahnsteuerungs- und Überwachungssysteme Deutsche Fassung EN 50128:2001	DIN EN 50128
VDE	Diese Norm ist zugleich eine VDE-Bestimmung im Sinne von VDE 0022. Sie ist nach Durchführung des vom VDE-Vorstand beschlossenen Genehmigungsverfahrens unter nebenstehenden Nummern in das VDE-Vorschriftenwerk aufgenommen und in der etz Elektrotechnische Zeitschrift bekannt gegeben worden.	Klassifikation VDE 0831 Teil 128
ICS 35.240.60; 45.020; 93.100 Railway applications – Communications, signalling and processing systems – Software for railway control and protection systems; German version EN 50128:2001 Applications ferroviaires – Systèmes de signalisation, de télécommunication et de traitement – Logiciels pour systèmes de commande et de protection ferroviaire; Version allemande EN 50128:2001 Die Europäische Norm EN 50128:2001 hat den Status einer Deutschen Norm. Beginn der Gültigkeit Die EN 50128 wurde 2000-11-01 angenommen. Nationales Vorwort Für die vorliegende Norm ist das nationale Arbeitsgremium UK 713.4 „Bahn-Signalanlagen“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE zuständig. Norm-Inhalt war veröffentlicht als E DIN EN 50128 (VDE 0831 Teil 128):1996-06. Fortsetzung Seite 2 und 102 Seiten EN DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE		

Nationaler Anhang NA (informativ) Zusammenhang mit Europäischen und Internationalen Normen

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf eine Norm ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils neueste gültige Ausgabe der in Bezug genommenen Norm.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe der Norm.

Der Zusammenhang der zitierten Normen mit den entsprechenden Deutschen Normen ist nachstehend wiedergegeben. Zum Zeitpunkt der Veröffentlichung dieser Norm waren die angegebenen Ausgaben gültig.

IEC hat 1997 die Benummerung der IEC-Publikationen geändert. Zu den bisher verwendeten Normnummern wird jeweils 60000 addiert. So ist zum Beispiel aus IEC 68 nun IEC 60068 geworden.

Tabelle NA.1

Europäische Norm	Internationale Norm	Deutsche Norm	Klassifikation im VDE-Vorschriftenwerk
EN 50126:1999	–	DIN EN 50126 (VDE 0115 Teil 103):2000-03	VDE 0115 Teil 103
EN 50129*	–	–	–
EN 50159-1:2001	–	DIN EN 50159-1 (VDE 0831 Teil 159-1):2001-11	VDE 0831 Teil 159-1
EN 50159-2:2001	–	DIN EN 50159-2 (VDE 0831 Teil 159-2)*	VDE 0831 Teil 159-2
EN ISO 9000-3:1997	ISO 9000-3:1997	DIN EN ISO 9000-3:1998-08	–
EN ISO 9001:2000	ISO 9001:2000	DIN EN ISO 9001:2000-12	–
* in Vorbereitung			

Nationaler Anhang NB (informativ) Literaturhinweise

DIN EN 50126 (VDE 0115 Teil 103), *Bahnanwendungen – Spezifikation und Nachweis der Zuverlässigkeit, Verfügbarkeit, Instandhaltbarkeit, Sicherheit (RAMS)*; Deutsche Fassung EN 50126:1999.

DIN EN 50159-1 (VDE 0831 Teil 159-1), *Bahnanwendungen – Telekommunikationstechnik, Signaltechnik und Datenverarbeitungssysteme – Teil 1: Sicherheitsrelevante Kommunikation in geschlossenen Übertragungssystemen*; Deutsche Fassung EN 50159-1:2001

DIN EN 50159-2 (VDE 0831 Teil 159-2), *Bahnanwendungen – Telekommunikationstechnik, Signaltechnik und Datenverarbeitungssysteme – Teil 2: Sicherheitsrelevante Kommunikation in offenen Übertragungssystemen*; Deutsche Fassung EN 50159-2:2001

DIN EN ISO 9000, *Qualitätsmanagementsysteme – Grundlagen und Begriffe (ISO 9000:2000)*; Dreisprachige Fassung EN ISO 9000:2000.

DIN EN ISO 9001, *Qualitätsmanagementsysteme – Anforderungen (ISO 9001:2000-09)*; Dreisprachige Fassung EN ISO 9001:2000.

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

Railway applications –
Communications, 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 2000-11-01 angenommen. Die 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 Zentralsekretariat 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 Zentralsekretariat mitgeteilt worden ist, hat den gleichen Status wie die offiziellen Fassungen.

CENELEC-Mitglieder sind die nationalen elektrotechnischen Komitees von Belgien, Dänemark, Deutschland, Finnland, Frankreich, Griechenland, Irland, Island, Italien, Luxemburg, Niederlande, Norwegen, Österreich, Portugal, Schweden, Schweiz, Spanien, der Tschechischen Republik und dem Vereinigten Königreich.

CENELEC

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

Zentralsekretariat: rue de Stassart 35, B-1050 Brüssel

Vorwort

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

Der Text des Entwurfs wurde der formellen Abstimmung unterworfen und von CENELEC am 2000-11-01 als EN 50128 angenommen.

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): 2001-11-01
- spätestes Datum, zu dem nationale Normen, die der EN entgegenstehen, zurückgezogen werden müssen (dow): 2003-11-01

Diese Europäische Norm sollte in Verbindung mit EN 50126: „Bahnanwendungen – Spezifikation und Nachweis der Zuverlässigkeit, Verfügbarkeit, Instandhaltbarkeit, Sicherheit (RAMS)“ und in Verbindung mit EN 50129: „Bahnanwendungen – Sicherheitsrelevante elektronische Systeme für Signaltechnik“ gelesen werden.

Anhänge, die als „normativ“ bezeichnet sind, gehören zum Norminhalt.

Anhänge, die als „informativ“ bezeichnet sind, enthalten nur Informationen.

In dieser Norm ist [Anhang A](#) normativ und ist [Anhang B](#) informativ.

Inhalt	Seite
Vorwort	2
Einleitung.....	5
1 Anwendungsbereich	7
2 Normative Verweisungen.....	7
3 Definitionen	8
4 Ziele und Konformität.....	11
5 Software-Sicherheitsanforderungsstufen	12
5.1 Ziel	12
5.2 Anforderungen	12
6 Personal und Verantwortlichkeiten	13
6.1 Ziele	13
6.2 Anforderungen	13
7 Fragen des Lebenszyklus und Dokumentation	15
7.1 Ziele	15
7.2 Anforderungen	15
8 Software-Anforderungsspezifikation	17
8.1 Ziele	17
8.2 Eingangsdokumente	17
8.3 Ausgangsdokumente	17
8.4 Anforderungen	17
9 Software-Architektur	18
9.1 Ziele	18
9.2 Eingangsdokumente	19
9.3 Ausgangsdokumente	19
9.4 Anforderungen	19
10 Software-Entwurf und Implementierung	20
10.1 Ziele	20
10.2 Eingangsdokumente	20
10.3 Ausgangsdokumente	21
10.4 Anforderungen	21
11 Software-Verifikation und Testen.....	23
11.1 Ziele	26
12.2 Eingangsdokumente	26
12.3 Ausgangsdokumente	26
12.4 Anforderungen	27
13 Software-Validierung.....	27
13.1 Ziele	27

13.2 Eingangsdokumente 28

13.3 Ausgangsdokumente 28

13.4 Anforderungen 28

14 Software-Begutachtung.....29

14.1 Ziele.....29

14.2 Eingangsdokumente 29

14.3 Ausgangsdokumente 29

14.4 Anforderungen 30

15 Software-Qualitätssicherung 30

15.1 Ziele.....30

15.2 Eingangsdokumente 31

15.3 Ausgangsdokumente 31

15.4 Anforderungen 31

16 Software-Wartung 33

16.1 Ziele.....33

16.2 Eingangsdokumente 33

16.3 Ausgangsdokumente 33

16.4 Anforderungen 33

17 Anwendungsspezifisch konfigurierbare Systeme 34

17.1 Ziele.....34

17.2 Eingangsdokumente 35

17.3 Ausgangsdokumente 35

17.4 Anforderungen 35

17.4.1 Lebenszyklus der Datengenerierung 35

17.4.2 Verfahren und Werkzeuge der Datengenerierung.....36

17.4.3 Software-Entwicklung 36

Anhang A (normativ) Kriterien für die Auswahl der Techniken und Maßnahmen.....43

Anhang B (informativ) Verfahrensübersicht 56

Bild 1 – Anforderungsstufen für sicherheitsrelevante Systeme 37

Bild 2 – Softwaresicherheit, Übersicht über das Vorgehen.....38

Bild 3 – Entwicklungs-Lebenszyklus 1 39

Bild 4 – Entwicklungs-Lebenszyklus 2 40

Bild 5 – Unabhängigkeit bei den verschiedenen Software-Anforderungsstufen.....41

Bild 6 – Beziehung zwischen der Entwicklung des generischen Systems und der Entwicklung der
Anwendung 42

Einleitung

Diese Norm ist Teil einer Gruppe verwandter Normen. Die anderen sind [EN 50126](#) „Bahnanwendungen – Spezifikation und Nachweis der Zuverlässigkeit, Verfügbarkeit, Instandhaltbarkeit, Sicherheit (RAMS)“ und [EN 50129](#) „Bahnanwendungen – Sicherheitsrelevante elektronische Systeme für Signaltechnik“. [EN 50126](#) behandelt Systemfragen im weitesten Sinn, während [EN 50129](#) den Zulassungsprozess für einzelne Systeme behandelt, die innerhalb des gesamten Eisenbahnsteuerungs- und Überwachungssystems existieren können. Die vorliegende Norm konzentriert sich auf die anzuwendenden Verfahren, um Software zu erhalten, welche die durch übergeordnete Betrachtungen an sie gestellten Sicherheitsanforderungen erfüllt.

Diese Norm verdankt einen großen Teil ihrer Ausrichtung den früheren Arbeiten der WG 9 des IEC/TC 65. Die Arbeit von WG 9 resultierte in einer Grundnorm für Software in Sicherheitssystemen, die nun Teil von IEC 61508 ist. Ein besonderer Aspekt der Arbeit von WG 9 ist die Einbeziehung der Software-Anforderungsstufe 0, die Software ohne Sicherheitsverantwortung abdeckt, als auch die Einbeziehung der Software-Anforderungsstufen 1 bis 4, die sicherheitsrelevante und sicherheitskritische Software abdecken. Diese Norm deckt auch alle fünf Software-Anforderungsstufen ab.

Ebenfalls berücksichtigt wurde die Arbeit der Institution of Railway Signal Engineers (IRSE), insbesondere ihr Technischer Bericht Nr. 1, welcher dasselbe Thema behandelt.

Das Grundkonzept dieser Europäischen Norm basiert auf Software-Sicherheitsanforderungsstufen. Je gefährlicher die Auswirkungen eines Softwarefehlers sind, desto höher ist die Software-Sicherheitsanforderungsstufe.

Diese Europäische Norm führt Techniken und Maßnahmen für 5 Software-Sicherheitsanforderungsstufen auf, wobei 0 die niedrigste und 4 die höchste Stufe ist. Vier dieser Stufen, die Stufen 1 bis 4, betreffen sicherheitsrelevante Software, während die Stufe 0 nicht-sicherheitsrelevante Software betrifft. Diese Stufe wurde als normativ aufgenommen, um einen glatten Übergang zwischen Software-Entwicklungen für nicht-sicherheitsrelevante Systeme und jenen für sicherheitsrelevante Systeme zu ermöglichen. Die erforderlichen Techniken und Maßnahmen für jede Software-Sicherheitsanforderungsstufe und für die nicht-sicherheitsrelevante Stufe werden in den Tabellen gezeigt. In dieser Norm sind die geforderten Techniken für Stufe 1 dieselben wie die für Stufe 2 und die geforderten Techniken für Stufe 3 dieselben wie für Stufe 4. Diese Europäische Norm gibt keine Hinweise darauf, welche Software-Anforderungsstufe für ein gegebenes Risiko angemessen ist. Diese Entscheidung hängt von vielen Faktoren ab, wie der Art der Anwendung, dem Ausmaß, in dem andere Systeme Sicherheitsfunktionen übernehmen, und von sozialen und wirtschaftlichen Faktoren.

Es ist die Aufgabe der [EN 50126](#) und der [EN 50129](#) die der Software zugewiesenen Sicherheitsfunktionen festzulegen.

Diese Europäische Norm beschreibt die Maßnahmen, die notwendig sind, um diese Anforderungen zu erfüllen. Der Ablauf ist in [Bild 1](#) dargestellt.

[EN 50126](#) und [EN 50129](#) fordern eine systematische Vorgehensweise zwecks:

- i) Identifikation von Gefährdungen, Risiken und Risikokriterien;
- ii) Identifikation der notwendigen Risikominderung, um die Risikokriterien zu erreichen;
- iii) Definition einer umfassenden Spezifikation der System-Sicherheitsanforderungen für die Schutzmaßnahmen, die notwendig sind, um die erforderliche Risikoreduzierung zu erreichen;
- iv) Auswahl einer passenden Systemarchitektur;
- v) Planung, Überwachung und Kontrolle der technischen und Managementaktivitäten, die erforderlich sind, um die Spezifikation der System-Sicherheitsanforderungen in ein sicherheitsrelevantes System mit einem validierten Sicherheitsverhalten (oder Sicherheitsanforderungsstufe) umzusetzen.

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

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

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

- Top-Down-Entwurfsverfahren;
- Modularität;
- Verifikation jeder Phase des Entwicklungslebenszyklus;
- verifizierte Module und Modul-Bibliotheken;
- klare Dokumentation;
- auditierbare Dokumente und
- Validierungstests.

Diese und verwandte Prinzipien müssen korrekt angewendet werden. Der vorliegende Standard beschreibt den Vertrauensgrad, der notwendig ist, um dies für jede Software-Sicherheitsanforderungsstufe zu zeigen.

Nachdem die Spezifikation der System-Sicherheitsanforderungen, die alle der Software zugeordneten Sicherheitsfunktionen identifiziert und die System-Sicherheitsanforderungsstufe bestimmt, übernommen oder erstellt wurde, sind die funktionalen Schritte bei der Anwendung dieser Europäischen Norm in [Bild 2](#) dargestellt. Diese sind:

- i) Definieren der Software-Anforderungsspezifikation und parallel dazu die Überlegungen zur Software-Architektur. In der Software-Architektur wird die grundlegende Sicherheits-Strategie für die Software und die Software-Sicherheitsanforderungsstufe entwickelt ([Abschnitte 5, 8 und 9](#)).
- ii) Entwerfen, entwickeln und testen der Software entsprechend dem Software-Qualitätssicherungsplan, der Software-Sicherheitsanforderungsstufe und dem Software-Lebenszyklus ([Abschnitt 10](#)).
- iii) Integration der Software auf der Zielhardware ([Abschnitt 12](#)).
- iv) Validierung der Software ([Abschnitt 13](#)).
- v) Wenn innerhalb der Betriebslebensdauer eine Software-Wartung erforderlich ist, ist diese Europäische Norm in geeigneter Form erneut anzuwenden ([Abschnitt 16](#)).

Eine Anzahl von Aktivitäten erstreckt sich über die gesamte Software-Entwicklung. Diese schließen die Verifikation ([Abschnitt 11](#)), Begutachtung ([Abschnitt 14](#)) und Qualitätssicherung ([Abschnitt 15](#)) ein.

Es werden Anforderungen für anwendungsspezifisch konfigurierbare Systeme festgelegt ([Abschnitt 17](#)).

Anforderungen werden auch an die Fähigkeiten der Software-Entwicklungsarbeitsgruppe gestellt ([Abschnitt 6](#)).

Die Norm schreibt nicht den Gebrauch eines bestimmten Software-Entwicklungslebenszyklus vor. Ein empfohlener Lebenszyklus und ein Satz von Dokumenten sind jedoch angegeben ([Abschnitt 7](#) und [Bilder 3 und 4](#)).

Es wurden Tabellen erstellt, die die verschiedenen Techniken/Maßnahmen entsprechend den 5 Software-Sicherheitsanforderungsstufen bewerten. Die Tabellen befinden sich in [Anhang A](#). Ein Verzeichnis gibt, bezugnehmend auf die Tabellen, eine kurze Beschreibung jeder Technik/Maßnahme mit Hinweisen auf weiterführende Informationen. Diese Verfahrensübersicht befindet sich im [Anhang B](#).

1 Anwendungsbereich

1.1 Diese Europäische Norm beschreibt Verfahren und technische Anforderungen für die Entwicklung von programmierbaren elektronischen Systemen für Eisenbahnsteuerungs- und -überwachungsanwendungen. Sie soll auf den gesamten Bereich der Sicherheitsanwendungen angewandt werden. Dies kann sehr kritische Anwendungen umfassen, wie die sichere Signalisierung, bis zu nicht-kritischen Anwendungen, wie z. B. Management-Informationssysteme. Diese Systeme können implementiert sein unter Verwendung dedizierter Mikroprozessoren, speicherprogrammierbarer Steuerungen, verteilter Multiprozessor-Systeme, großer Zentralrechner-Systeme oder anderer Architekturen.

1.2 Diese Europäische Norm ist ausschließlich auf Software und die Wechselwirkung zwischen Software und dem System anwendbar, zu dem die Software gehört.

1.3 Software-Sicherheitsanforderungsstufen größer als Null sind für die Anwendung in solchen Systemen gedacht, in denen Fehlfunktionen den Tod von Personen nach sich ziehen können. Wirtschaftliche oder Umweltüberlegungen können jedoch auch die Anwendung von höheren Software-Sicherheitsanforderungsstufen rechtfertigen.

1.4 Diese Europäische Norm gilt für jegliche Software, die bei der Entwicklung und Realisierung von Eisenbahnsteuerungs- und -überwachungssystemen verwendet wird, einschließlich:

- Anwendungsprogrammierung
- Betriebssysteme
- unterstützende Werkzeuge
- Firmware

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

1.5 Die Nutzung kommerziell verfügbarer Standardsoftware und von Werkzeugen wird in dieser Europäischen Norm ebenfalls angesprochen.

1.6 Diese Europäische Norm behandelt auch Anforderungen für anwendungsspezifisch konfigurierbare Systeme.

1.7 Die Behandlung kommerzieller Probleme ist nicht Zweck dieser Europäischen Norm. Diese sollten als ein wesentlicher Teil von vertraglichen Vereinbarungen angesehen werden. Alle Abschnitte dieser Europäischen Norm müssen in jeder kommerziellen Situation berücksichtigt werden.

1.8 Diese Europäische Norm gilt nicht rückwirkend. Sie gilt daher in erster Linie für Neuentwicklungen und in ihrer Gesamtheit für vorhandene Systeme nur dann, wenn diese größeren Änderungen unterworfen werden sollen. Für kleinere Änderungen ist nur [Abschnitt 16](#) anzuwenden.

2 Normative Verweisungen

Diese Europäische Norm enthält durch datierte oder undatierte Verweisungen Festlegungen aus anderen Publikationen. Diese normativen Verweisungen sind an den jeweiligen Stellen im Text zitiert, und die Publikationen sind nachstehend aufgeführt. Bei datierten Verweisungen gehören spätere Änderungen oder Überarbeitungen dieser Publikationen nur zu dieser Europäischen Norm, falls sie durch Änderung oder Überarbeitung eingearbeitet sind. Bei undatierten Verweisungen gilt die letzte Ausgabe der in Bezug genommenen Publikation (einschließlich Änderungen).

EN 50126, *Railway applications – The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS)*.

EN 50129*, *Railway applications – Safety related electronic systems for signalling.*

EN 50159-1, *Railway applications – Communication, signalling and processing systems – Part 1: Safety-related communication in closed transmission systems.*

EN 50159-2, *Railway applications – Communication, signalling and processing systems – Part 2: Safety-related communication in open transmission systems.*

EN ISO 9001, *Quality systems – Model for quality assurance in design/development, production, installation and serving.*

EN 9000-3, *Quality management and quality assurance standards – Part 3: Guidelines for the application of ISO 9001:1994 to the development, supply, installation and maintenance of computer software.*

3 Definitionen

Für diese Europäische Norm gelten nachfolgende Definitionen. Für Begriffe, die nicht hier definiert sind, sollten folgende Unterlagen entsprechend der angegebenen Reihenfolge zu Rate gezogen werden:

EN ISO 8402	Qualitätsmanagement – Begriffe
IEC 60050-191	International Electrotechnical Vocabulary – Chapter 191: Dependability and quality of service
IEEE 610.12	IEEE standard glossary of software engineering terminology
ISO/IEC 2382	Information Technology Vocabulary
ISO/IEC 9126	Information Technology – Software Product Evaluation – Quality characteristics and guidelines for their use

3.1

Begutachtung

Analyseprozess zur Feststellung, ob die Entwurfsinstanz und der Validierer ein Produkt zustande gebracht haben, das die spezifizierten Anforderungen erfüllt, und um zu beurteilen, ob das Produkt für seinen gedachten Anwendungszweck geeignet ist

3.2

Gutachter

Person oder Beauftragte Stelle, die zur Durchführung der Begutachtung benannt ist

3.3

Verfügbarkeit

Fähigkeit eines Produkts, in einem Zustand zu sein, die geforderte Funktion unter vorgegebenen Bedingungen zu einer bestimmten Zeit oder über ein vorgegebenes Zeitintervall auszuführen, unter der Annahme, dass die erforderlichen externen Ressourcen zur Verfügung stehen

3.4

kommerzielle Standard-Software (commercial off-the-shelf (COTS) software)

durch Markterfordernisse bestimmte Software, die im Handel verfügbar ist und deren Einsatztauglichkeit durch ein breites Spektrum kommerzieller Anwender bewiesen ist

3.5

Entwurfsinstanz

Stelle, die verantwortlich ist für die Formulierung eines Entwurfs zur Erfüllung der spezifizierten Anforderungen und für die Beaufsichtigung der folgenden Entwicklung und Inbetriebsetzung eines Systems in seiner vorgesehenen Umgebung

* in Vorbereitung

3.6

Entwerfer

eine oder mehrere durch die Entwurfsinstanz bestimmte Personen zur Analyse spezifizierter Anforderungen und deren Umsetzung in akzeptable Entwürfe, die die geforderte Sicherheitsanforderungsstufe erfüllen

3.7

Element

Teil eines Produkts, das als Grund- oder Baueinheit dient. Ein Element kann einfach oder komplex sein

3.8

Fehler

Abweichung vom beabsichtigten Entwurf, die zu unerwünschtem Systemverhalten oder Ausfall führen kann

3.9

Fehlfunktion/Ausfall

Abweichung vom spezifizierten Verhalten des Systems. Eine Fehlfunktion/Ausfall ist die Folge einer Fehlerursache oder eines Fehlers im System

3.10

Fehlerursache

abnormaler Zustand, der zu einem Fehler oder einer Fehlfunktion/Ausfall in einem System führen kann. Eine Fehlerursache kann zufällig oder systematisch sein

3.11

Fehlervermeidung

Gebrauch von Entwurfstechniken, die auf die Vermeidung von Fehlerursachen während des Entwurfs und der Erstellung des Systems abzielen

3.12

Fehlertoleranz

eingebaute Fähigkeit eines Systems, weiterhin korrekt die spezifizierte Servicefunktion zur Verfügung zu stellen, obwohl eine begrenzte Anzahl von Hardware- oder Softwarefehlerursachen vorliegt

3.13

Firmware

geordneter Satz von Anweisungen und zugehörigen Daten, die in einer vom Hauptspeicher unabhängigen Art und Weise gespeichert sind, üblicherweise in einem ROM

3.14

generische Software

Software, die für eine Anzahl von Installationen durch einfaches Bereitstellen von anwendungsspezifischen Daten benutzt werden kann

3.15

Implementierer

eine oder mehrere durch die Entwurfsinstanz bestimmte Personen zur Umsetzung spezifizierter Entwürfe in ihre physikalische Realisierung

3.16

Produkt

Menge von Elementen, die in einer zur Erfüllung der spezifizierten Anforderungen geeigneten Art und Weise zu einem System, Subsystem oder Bestandteil einer Einrichtung zusammengeschaltet sind. In dieser Europäischen Norm kann ein Produkt als ausschließlich aus Softwareelementen oder Dokumenten aufgebaut angesehen werden

3.17

speicherprogrammierbare Steuerung

elektronisches Steuerungssystem, das einen anwenderprogrammierbaren Speicher für die Speicherung von Anweisungen besitzt, um spezifische Funktionen zu implementieren

3.18

Zuverlässigkeit

Fähigkeit eines Bauteils, die geforderte Funktion unter bestimmten Bedingungen für eine gewisse Zeit auszuführen

3.19

Anforderungs-Verfolgbarkeit

Anforderungs-Verfolgbarkeit soll sicherstellen, dass gezeigt werden kann, dass alle Anforderungen ordnungsgemäß erfüllt werden

3.20

Risiko

Kombination der Häufigkeit oder Wahrscheinlichkeit mit den Auswirkungen eines spezifizierten gefährlichen Ereignisses

3.21

Sicherheit

Freiheit von nicht-akzeptablen Risiken

3.22

Sicherheitsbehörde

Inстанz, die für die Zertifizierung verantwortlich ist, dass das sicherheitsrelevante System für den Einsatz geeignet ist und allen wesentlichen gesetzlichen Sicherheitsvorgaben und -regelungen entspricht

3.23

sicherheitsrelevante Software

Software, die Sicherheitsverantwortung trägt

3.24

Software

intellektuelle Schöpfung, die Programme, Verfahren, Regeln und alle dazugehörige Dokumentation umfasst, die zum Betrieb eines Systems gehören

ANMERKUNG Software ist unabhängig vom verwendeten Transportmedium.

3.25

Software-Lebenszyklus

Aktivitäten, die während eines Zeitraums anfallen, der mit der Konzeption der Software beginnt und endet, wenn die Software nicht länger für den Einsatz verfügbar ist. Der Software-Lebenszyklus umfasst typischerweise die Anforderungsphase, die Entwicklungsphase, die Testphase, die Integrationsphase, die Installations- und die Wartungsphase

3.26

Software-Wartbarkeit

Änderungsfähigkeit eines Systems zur Beseitigung von Fehlern, zur Verbesserung des Leistungsverhaltens oder anderer Eigenschaften oder zur Anpassung an eine andere Umgebung

3.27

Softwarewartung

einzelne oder mehrere an Software durchgeführte Aktionen nach deren Abnahme durch den Endnutzer. Ziel ist die Verbesserung, Erweiterung und/oder Berichtigung der Funktionalität

3.28

Software-Sicherheitsanforderungsstufe

Klassifizierungszahl, die die Techniken und Maßnahmen bestimmt, die zwecks Begrenzung der Software-Restfehler auf ein angemessenes Niveau angewendet werden müssen

3.29

System-Sicherheitsanforderungsstufe

Zahl, die den erforderlichen Vertrauensgrad angibt, dass ein System seine spezifizierten Sicherheitseigenschaften erfüllen wird

3.30

Verfolgbarkeit

das Maß für die Beziehung zwischen zwei oder mehr Ergebnissen eines Entwicklungsprozesses zueinander, vor allen Dingen solcher Ergebnisse, die einen Vorgänger/Nachfolger haben oder über- bzw. untergeordnet zueinander sind

3.31

Validierung

Analyse und Testen zur Demonstration, dass das Produkt in allen Belangen seine spezifizierten Anforderungen erfüllt

3.32

Validierer

Person oder Stelle, die zur Durchführung der Validierung benannt ist

3.33

Verifikation

Analyse und Testen, um festzustellen, ob das Ausgangsprodukt jeder Phase des Lebenszyklus die Anforderungen aus der vorhergehenden Phase erfüllt

3.34

Verifizierer

Person oder Stelle, die zur Durchführung der Verifikation benannt ist

4 Ziele und Konformität

4.1 In jedem der folgenden Abschnitte sind die Ziele und Anforderungen des jeweiligen Kapitels detailliert beschrieben.

4.2 Um mit dieser Europäischen Norm übereinzustimmen, muss gezeigt werden, dass jede der Anforderungen entsprechend der festgelegten Software-Sicherheitsanforderungsstufe erfüllt und daher das Ziel des Abschnitts erreicht wurde.

4.3 Wo eine Anforderung durch den Wortlaut „in dem durch die Software-Sicherheitsanforderungsstufe festgelegten Umfang“ beschrieben wird, bedeutet dies, dass eine Reihe von Techniken und Maßnahmen benutzt werden darf, um jene Anforderung zu erfüllen.

4.4 Liegt eine Formulierung gemäß 4.3 vor, sollten die in dieser Europäischen Norm detailliert aufgeführten Tabellen verwendet werden, um die Auswahl der für die Software-Sicherheitsanforderungsstufe geeigneten Techniken und Maßnahmen zu unterstützen.

4.5 Falls eine Technik oder Maßnahme in den Tabellen als dringend empfohlen (HR) bezeichnet wird, sollte die Begründung für eine Nichtbenutzung dieser Technik in angemessener Weise entweder im Software-Qualitätssicherungsplan oder einem anderen durch den Software-Qualitätssicherungsplan referenzierten Dokument genauer ausgeführt und dokumentiert werden. Dies ist nicht erforderlich, wenn eine in der entsprechenden Tabelle stehende zugelassene Kombination von Techniken benutzt wird.

4.6 Falls eine Technik oder Maßnahme zur Anwendung vorgeschlagen ist, die nicht in den Tabellen enthalten ist, dann ist ihre Wirksamkeit und Eignung zum Erreichen der besonderen Anforderung und des Gesamtziels des Abschnitts in geeigneter Weise entweder im Software-Qualitätssicherungsplan oder einem anderen durch den Software-Qualitätssicherungsplan referenzierten Dokument zu begründen und zu dokumentieren.

4.7 Die Übereinstimmung mit den Anforderungen eines bestimmten Abschnitts und den entsprechenden Techniken und Maßnahmen, die in den Tabellen näher beschrieben sind, muss durch die Überprüfung der in dieser Norm geforderten Unterlagen, durch sonstige objektive Beweise, Auditierung und durch das Bezeugen von Tests begutachtet werden.

4.8 Diese Europäische Norm erfordert die Benutzung eines Bündels von Techniken und deren richtige Anwendung. Diese Techniken werden in den Tabellen festgelegt und in der Verfahrensübersicht¹⁾ näher beschrieben.

5 Software-Sicherheitsanforderungsstufen

5.1 Ziel

Beschreibung der Zuweisung von Software-Sicherheitsanforderungsstufen an die Software.

5.2 Anforderungen

5.2.1 In Übereinstimmung mit **EN 50126** und **EN 50129** müssen erstellt werden:

- System-Anforderungsspezifikation;
- Spezifikation der System-Sicherheitsanforderungen;
- System-Architekturbeschreibung;
- System-Sicherheitsplan;

mit folgendem Inhalt:

- Sicherheitsfunktionen;
- Konfiguration oder Architektur des Systems;
- Zuverlässigkeitsanforderungen an die Hardware;
- Anforderungen an den Sicherheitsgrad.

Die Software-Sicherheitsanforderungsstufe muss durch das allgemeine in **EN 50126** beschriebene Verfahren zur Bestimmung einer Sicherheitsanforderungsstufe festgelegt werden.

5.2.2 Die erforderliche Software-Sicherheitsanforderungsstufe muss auf der Grundlage des Risikogrades festgelegt werden, der mit der Benutzung der Software im System und mit der System-Sicherheitsanforderungsstufe verbunden ist.

5.2.3 Ohne besondere Vorsorgemaßnahmen muss die Software-Sicherheitsanforderungsstufe mindestens so groß sein wie die System-Sicherheitsanforderungsstufe. Wenn jedoch Mechanismen existieren, die verhindern, dass der Fehler eines Softwaremoduls das System in einen unsicheren Zustand bringt, kann die Software-Sicherheitsanforderungsstufe des Moduls verringert werden.

5.2.4 Zu berücksichtigen sind Risiken mit folgenden Gefahrenausswirkungen:

- Tod einer oder mehrerer Personen;
- Verletzung oder Erkrankung von Personen;

¹⁾ Nationale Fußnote: [Anhang B](#).

- Umweltverschmutzung und
- Verlust oder Beschädigung von Eigentum.

5.2.5 Das Risiko kann quantifiziert werden. Es ist aber nicht möglich, die Software-Sicherheitsanforderung in der gleichen Weise festzulegen. Daher muss für diese Europäische Norm die Software-Sicherheitsanforderung als eine der folgenden fünf Stufen angegeben werden:

Software-Sicherheitsanforderungsstufe	Beschreibung der Software-Sicherheitsanforderung
4	sehr hoch
3	hoch
2	mittel
1	niedrig
0	nicht sicherheitsrelevant

5.2.6 Die Software-Sicherheitsanforderungsstufe muss in der Software-Anforderungsspezifikation ([Abschnitt 8](#)) festgelegt werden. Wenn unterschiedliche Softwarekomponenten unterschiedliche Software-Sicherheitsanforderungsstufen haben, so müssen diese in der Software-Architekturspezifikation beschrieben werden ([Abschnitt 9](#)).

6 Personal und Verantwortlichkeiten

6.1 Ziele

Sicherstellen, dass das gesamte Personal, das Verantwortung für die Software hat, auch so qualifiziert ist, dass es diese Verantwortung wahrnehmen kann.

6.2 Anforderungen

6.2.1 Der Lieferant und/oder Entwickler und der Kunde müssen mindestens die wichtigen Teile der EN ISO 9001 umsetzen, unter Beachtung der Leitlinien nach EN ISO 9000-3.

6.2.2 Außer in der Software-Sicherheitsanforderungsstufe 0 muss der Sicherheits-Prozess unter der Kontrolle einer geeigneten Sicherheits-Organisation implementiert werden, die der [EN 50129](#), Unterabschnitt „Sicherheits-Organisation“ im Abschnitt „Nachweis des Sicherheits-Managements“, entspricht.

6.2.3 Das gesamte Personal, das in allen Phasen des Software-Lebenszyklus, einschließlich der Managementaktivitäten, tätig ist, muss die entsprechende Ausbildung, Erfahrung und Qualifikation aufweisen.

6.2.4 Es wird dringend empfohlen, dass die Ausbildung, Erfahrung und Qualifikation des gesamten Personals, das in allen Phasen des Software-Lebenszyklus, einschließlich der Managementaktivitäten, tätig ist, unter Berücksichtigung der jeweiligen Anwendung begründet wird, außer für Software-Sicherheitsanforderungsstufe 0.

6.2.5 Die in 6.2.4 enthaltene Begründung muss im Software-Qualitätssicherungsplan festgehalten werden und muss den Nachweis einer angemessenen Kompetenz für die folgenden Bereiche einschließen:

- für den Anwendungsbereich geeignetes technisches Können;
- Softwaretechnik;
- Rechnersystemtechnik;
- Sicherheitstechnik;
- Gesetzes- und Vorschriftenwerke.

6.2.6 Ein unabhängiger Gutachter für die Software muss ernannt werden. Siehe auch 6.2.10 und [14.4.1](#).

6.2.7 Dem Gutachter muss die Autorität gegeben werden, um die Begutachtung der Software vornehmen zu können.

6.2.8 Während des Software-Lebenszyklus müssen die beteiligten Gruppen in dem durch die Software-Sicherheitsanforderungsstufe festgelegten Umfang unabhängig sein, in Übereinstimmung mit [Bild 5](#), das wie folgt interpretiert werden muss:

Auf allen Software-Sicherheitsanforderungsstufen muss der Gutachter durch die Sicherheitsbehörde zugelassen und, außer für die in 6.2.10 definierten Verhältnisse, unabhängig vom Lieferanten sein.

Entwerfer/Implementierer, Verifizierer und Validierer dürfen alle der gleichen Firma angehören, müssen aber die folgenden Regeln für ein Mindestmaß an Unabhängigkeit erfüllen:

Für die Software-Sicherheitsanforderungsstufe 0:

Hier gibt es keine Beschränkungen; Entwerfer, Implementierer, Verifizierer und Validierer dürfen dieselbe Person sein.

Für die Software-Sicherheitsanforderungsstufe 1 & 2:

Verifizierer und Validierer dürfen dieselbe Person, aber nicht identisch mit dem Entwerfer/Implementierer sein. Jedoch dürfen der Entwerfer/Implementierer, Verifizierer und Validierer dem gleichen Projektmanager berichten.

Für die Software-Sicherheitsanforderungsstufe 3 & 4 sind zwei Anordnungen zulässig:

- a) Verifizierer und Validierer dürfen dieselbe Person, aber nicht identisch mit dem Entwerfer/Implementierer sein. Zusätzlich dürfen Verifizierer und Validierer nicht dem Projektmanager berichten, wie der Entwerfer/Implementierer dies tun, und sie müssen das Recht haben, die Herausgabe des Produkts zu verhindern.
- b) Entwerfer/Implementierer, Verifizierer und Validierer müssen alle unterschiedliche Personen sein. Entwerfer/Implementierer und Verifizierer dürfen dem Projektmanager berichten, der Validierer nicht. Der Validierer muss das Recht haben, zu verhindern, dass das Produkt in den Verkehr gebracht wird.

6.2.9 Die für die verschiedenen Abschnitte verantwortlichen Parteien sind folgende:

Software-Anforderungsspezifikation (Abschnitt 8)	Entwerfer
Software-Anforderungstestspezifikation (Abschnitt 8)	Validierer
Software-Architektur (Abschnitt 9)	Entwerfer
Software-Entwurf und -Entwicklung (Abschnitt 10)	Entwerfer
Software-Verifikation und -Testen (Abschnitt 11)	Verifizierer
Software/Hardware-Integration (Abschnitt 12)	Entwerfer
Software-Validierung (Abschnitt 13)	Validierer
Software-Begutachtung (Abschnitt 14)	Gutachter

6.2.10 Es liegt im Ermessen der Sicherheitsbehörde, dass der Gutachter Teil der Organisation des Lieferanten oder des Kunden ist, jedoch muss in diesem Fall der Gutachter

- durch die Sicherheitsbehörde autorisiert sein;
- völlig unabhängig von der Projektarbeitsgruppe sein;
- direkt an die Sicherheitsbehörde berichten.

7 Fragen des Lebenszyklus und Dokumentation

7.1 Ziele

- 7.1.1 Gliederung der Entwicklung der Software in festgelegte Phasen und Aktivitäten.
- 7.1.2 Aufzeichnung aller der Software dienenden Informationen während des gesamten Lebenszyklus der Software.

7.2 Anforderungen

7.2.1 Es muss ein Lebenszyklusmodell für die Software-Entwicklung ausgewählt werden. Dieses muss im Software-Qualitätssicherungsplan entsprechend [Abschnitt 15](#) dieser Europäischen Norm detailliert beschrieben werden. Als Beispiele sind zwei Lebenszyklusmodelle in den [Bildern 3](#) und [4](#) dargestellt.

7.2.2 Qualitätssicherungsverfahren müssen parallel zu den Aktivitäten des Lebenszyklus ablaufen und dieselben Begriffe verwenden.

7.2.3 Alle Aktivitäten, die innerhalb einer Phase durchzuführen sind, müssen vor dem Phasenbeginn definiert werden. Jede Phase des Software-Lebenszyklus muss in Teilaufgaben mit genauer Definition des Ein- und Ausgangs sowie der jeweiligen Aktivität unterteilt werden.

7.2.4 Der Software-Qualitätssicherungsplan muss beschreiben, welche Verifikationsschritte und -berichte erforderlich sind.

7.2.5 Alle Dokumente müssen strukturiert werden, um eine fortlaufende Erweiterung parallel zum Entwurfsprozess zu erlauben.

7.2.6 Die Verfolgbarkeit der Dokumente muss gegeben sein, indem jedes Dokument eine eindeutige Bezugsnummer und einen festgelegten und dokumentierten Bezug zu anderen Dokumenten besitzt. Jeder Begriff, jedes Akronym oder jede Abkürzung müssen in jedem Dokument dieselbe Bedeutung haben. Wenn dies aus historischen Gründen nicht möglich ist, müssen die verschiedenen Bedeutungen aufgelistet und die Bezüge angegeben werden.

Zusätzlich muss jedes Dokument, außer Dokumente für COTS-Software (siehe [9.4.5](#)) oder früher entwickelte Software (siehe [9.4.6](#)), entsprechend den folgenden Regeln geschrieben sein:

- a) Es muss alle anwendbaren Bedingungen und Anforderungen des Vorgängerdokuments, mit dem es eine hierarchische Beziehung hat, enthalten oder realisieren.
- b) Es darf dem Vorgängerdokument nicht widersprechen.
- c) In jedem Dokument müssen alle Begriffe, Akronyme oder Abkürzungen die gleiche Bedeutung haben.
- d) In allen Dokumenten muss auf jeden Themenpunkt oder Konzept mit dem gleichen Namen oder derselben Beschreibung Bezug genommen werden.

7.2.7 Der Inhalt aller Dokumente muss in einer Form festgehalten werden, die für die Handhabung, Bearbeitung und Aufbewahrung geeignet ist.

7.2.8 Es müssen die in der Dokumenten-Cross-Referenz-Tabelle (siehe unten) aufgeführten Dokumente in dem durch die Software-Sicherheitsanforderungsstufe festgelegten Umfang erstellt werden.

7.2.9 In Abhängigkeit von der Größe, Komplexität und dem Lebenszyklus der zu entwickelnden Software wird die Anzahl der verschiedenen zu erstellenden Dokumente unterschiedlich sein. Einige Dokumente dürfen kombiniert werden (vorausgesetzt, es gibt keinen Verlust der notwendigen Darstellungsgenauigkeit des Prozesses). Bei großen Projekten kann es notwendig werden, die aufgelisteten Dokumente in einer hierarchischen Weise in eine Anzahl besser handhabbarer Tochterdokumente zu unterteilen. Dokumente,

die durch unabhängige Arbeitsgruppen oder Stellen produziert wurden, dürfen nicht in einem Dokument zusammengefasst werden.

7.2.10 Die Beziehung zwischen den verschiedenen in [Abschnitt 7](#) ausgewiesenen Dokumenten kann auch durch die Benutzung einer DCRT (Dokumenten-Cross-Referenz-Tabelle) definiert werden. Für jedes in der Spalte „Dokument“ aufgeführte Dokument kann die Phase und der Abschnitt, die mit der Erstellung des Dokuments zu tun haben, durch horizontales und vertikales Lesen aus dem Feld mit dem Symbol „■“ gefunden werden. Die Phasen, in denen es verwendet wird, können gefunden werden durch vertikales Lesen ausgehend von den durch „♦“ symbolisierten Feldern. Der Abschnitt oder andere Verweise auf die Definition des Dokuments sind in der „Definiert in“-Spalte zu finden. Wo ein Abschnitt angegeben ist, sollten die darauf folgenden Abschnitte auch überprüft werden, da sie weitere Informationen enthalten können. Es ist anzumerken, dass die Referenz für den Software-Konfigurationsmanagementplan in Klammern angegeben ist, weil dieser Abschnitt lediglich auf die EN ISO 9001 verweist.

Dokumenten-Cross-Referenz-Tabelle

Abschnitt - Titel -	8 SAS	9 SAr	10 SEE	11 SVer	12 SI/HI	13 SVal	14 Beg	15 Q	16 Wa	DOKUMENTE	Definiert in
PHASEN (*) parallel mit anderen											
SYSTEM-EINGABEN	♦			♦	♦					System-Anforderungsspezifikation	ENV 50129 Par. 5.4, Nr. 2.3 und Anhang B.2.3
	♦	♦		♦	♦	♦	♦			System-Sicherheitsanforderungsspezifikation	ENV 50129 Par. 5.4, Nr. 2.4 und Anhang B.2.4
	♦				♦					System-Architekturbeschreibung	ENV 50129 Par. 5.4 Nr. 2.4 und Anhang B.2.1
										System-Sicherheitsplan	ENV 50129 Par. 5.3.4 und EN 50126
SW-PLANUNG (*)	♦	♦	♦	♦		♦	♦	■		SW-Qualitätssicherungsplan	15.4.3
						♦	♦	■		SW-Konfigurationsmanagementplan	(15.4.2)
				■		♦	♦			SW-Verifikationsplan	11.4.1
				■		♦	♦			SW-Integrationstestplan	11.4.5
					■	♦	♦			SW/HW-Integrationstestplan	12.4.1
						■	♦			SW-Validierungsplan	13.4.3
							♦	■		SW-Wartungsplan	16.4.3
										Daten-Generierungsplan	17.4.2.1
										Daten-Testplan	17.4.2.4
SW- ANFORDERUNGEN	■	♦	♦	♦	♦	♦	♦			SW-Anforderungsspezifikation	8.4.1
										Installationsspezifische Anforderungsdefinition	17.4.1.1
	■			♦	♦	♦	♦			SW-Anforderungstestspezifikation	8.4.13
				■						SW-Anforderungsverifikationsbericht	11.4.11
SW-ENTWURF		■	♦	♦	♦	♦	♦			SW-Architekturspezifikation	9.4.1
			■	♦	♦	♦	♦			SW-Entwurfsspezifikation	10.4.3
				■						SW-Arch. und Entwurfsverifikationsbericht	11.4.12
SW-MODULENTWURF			■	♦	♦	♦	♦			SW-Modulentwurfsspezifikation	10.4.3
			■	♦	♦	♦	♦			SW-Modultestspezifikation	10.4.14
				■						SW-Modulverifikationsbericht	11.4.13
CODE			■	♦	♦	♦	♦			SW-Quellcode	
				■		♦	♦			SW-Quellcodeverifikationsbericht	11.4.14
MODULTEST			■	♦						SW-Modultestbericht	10.4.14
SW-INTEGRATION				■						SW-Integrationstestbericht	11.4.15
										Daten-Testbericht	17.4.2.4
SW/HW-INTEGRATION					■					SW/HW-Integrationstestbericht	12.4.8
VALIDIERUNG (*)						■				SW-Validierungsbericht	13.4.10
BEGUTACHTUNG (*)							■			SW-Gutachten	14.4.9
WARTUNG								■		SW-Änderungsaufzeichnung	16.4.9
								■		SW-Wartungsaufzeichnung	16.4.8

8 Software-Anforderungsspezifikation

8.1 Ziele

8.1.1 Beschreiben eines Dokuments, das einen vollständigen Satz von Anforderungen an die Software definiert, der alle Systemanforderungen in dem durch die Software-Sicherheitsanforderungsstufe festgelegten Umfang erfüllt. Es soll ein derart umfassendes Dokument für jeden Software-Ingenieur sein, dass es für ihn nicht erforderlich ist, zum Thema Anforderungen in irgendeinem anderen Dokument nachzusehen.

8.1.2 Beschreiben der Software-Anforderungstestspezifikation.

8.2 Eingangsdokumente

- 1) System-Anforderungsspezifikation;
- 2) Spezifikation der System-Sicherheitsanforderungen;
- 3) System-Architekturbeschreibung;
- 4) Software-Qualitätssicherungsplan.

8.3 Ausgangsdokumente

- 1) Software-Anforderungsspezifikation;
- 2) Software-Anforderungstestspezifikation.

8.4 Anforderungen

8.4.1 Die Software-Anforderungsspezifikation muss die erforderlichen Eigenschaften der zu entwickelnden Software zum Ausdruck bringen und nicht die Verfahren, sie zu entwickeln. Diese Eigenschaften, die alle (außer der Sicherheit) in ISO/IEC 9126 definiert sind, müssen einschließen:

- Funktionalität (einschließlich Kapazität und Antwortzeitverhalten);
- Zuverlässigkeit und Wartbarkeit;
- Sicherheit (einschließlich Sicherheitsfunktionen und ihre zugehörigen Software-Sicherheitsanforderungsstufen);
- Effizienz;
- Benutzbarkeit;
- Portierbarkeit.

Die Software-Sicherheitsanforderungsstufe muss nach der Definition in [Abschnitt 5](#) abgeleitet und in der Software-Anforderungsspezifikation festgehalten werden.

8.4.2 In dem durch die Software-Sicherheitsanforderungsstufe festgelegten Umfang muss die Software-Anforderungsspezifikation so formuliert und strukturiert werden, dass sie:

- i) vollständig, klar, genau, eindeutig, verifizierbar, testbar, wartbar und umsetzbar ist;
- ii) auf alle in 8.2 genannten Unterlagen rückverfolgbar ist.

8.4.3 Die Software-Anforderungsspezifikation muss Ausdrucksweisen und Beschreibungen verwenden, die für das verantwortliche Personal, das in den gesamten Lebenszyklus des Systems involviert ist, verständlich sind.

8.4.4 Die Software-Anforderungsspezifikation muss alle Schnittstellen zu beliebigen anderen Systemen identifizieren und beschreiben, sei es innerhalb oder außerhalb der zu steuernden Einrichtung, einschließlich der Bedienerchnittstelle, wo immer eine direkte Verbindung besteht oder geplant ist.

8.4.5 Alle relevanten Betriebsarten müssen in der Software-Anforderungsspezifikation detailliert dargestellt werden.

8.4.6 Alle relevanten Verhaltensweisen der programmierbaren Elektronik, insbesondere das Ausfallverhalten, müssen in der Software-Anforderungsspezifikation detailliert beschrieben werden.

8.4.7 Alle Zwangsbedingungen zwischen Hardware und Software müssen in der Software-Anforderungsspezifikation identifiziert und beschrieben werden.

8.4.8 Die Software-Anforderungsspezifikation muss auf den Umfang der Software-Selbsttests und den spezifizierten Grad der Hardware-Prüfungen durch die Software hinweisen. Der Software-Selbsttest besteht im Erkennen und Melden eigener Ausfälle und von Fehlern durch die Software selbst.

8.4.9 Die Software-Anforderungsspezifikation muss Anforderungen für das periodische Testen von Funktionen in dem durch die Spezifikation der System-Sicherheitsanforderungen festgelegten Umfang einschließen.

8.4.10 Die Software-Anforderungsspezifikation muss Anforderungen enthalten, die es ermöglichen, dass alle Sicherheitsfunktionen während des Betriebs des Gesamtsystems in dem durch die Spezifikation der System-Sicherheitsanforderungen festgelegten Umfang getestet werden können.

8.4.11 Wenn die Software Funktionen ausführen soll, die sich besonders darauf beziehen, die erforderliche System-Sicherheitsanforderungsstufe zu erreichen, müssen diese eindeutig in der Software-Anforderungsspezifikation gekennzeichnet werden.

8.4.12 Wenn die Software Funktionen ohne Sicherheitsanforderungen ausführen soll, müssen diese eindeutig in der Software-Anforderungsspezifikation gekennzeichnet werden.

8.4.13 Eine Software-Anforderungstestspezifikation muss aus der Software-Anforderungsspezifikation entwickelt werden. Diese Testspezifikation muss für die Verifikation aller in der Software-Anforderungsspezifikation beschriebenen Anforderungen verwendet werden und auch als eine Beschreibung der an der fertigen Software auszuführenden Tests.

8.4.14 Die Software-Anforderungstestspezifikation muss die Testfälle für jede geforderte Funktion aufführen, einschließlich:

- i) der erforderlichen Eingangsdaten mit ihrer Reihenfolge und ihren Werten;
- ii) der erwarteten Ausgangsdaten mit ihrer Reihenfolge und ihren Werten;
- iii) den Akzeptanzkriterien einschließlich der Leistungs- und Qualitätsaspekte.

8.4.15 Die Verfolgbarkeit der Anforderungen muss ein wichtiger Grundgedanke für die Validierung eines sicherheitsrelevanten Systems sein, und es sind Mittel bereitzustellen, um dies in allen Phasen des Lebenszyklus zeigen zu können.

8.4.16 Für evtl. vorhandenes nicht-verfolgbares Material muss gezeigt werden, dass es keine Bedeutung für die Sicherheit oder Integrität des Systems hat.

9 Software-Architektur

9.1 Ziele

9.1.1 Entwickeln einer Software-Architektur, die die Vorgaben der Software-Anforderungsspezifikation in dem durch die Software-Sicherheitsanforderungsstufe festgelegten Umfang erfüllt.

9.1.2 Überprüfen der Anforderungen, die durch die System-Architektur an die Software gestellt werden.

9.1.3 Feststellen und Bewerten, was die Wechselwirkungen zwischen Hardware und Software für die Sicherheit bedeuten.

9.1.4 Auswahl eines Entwurfsverfahren, falls dies nicht schon früher geschehen ist.

9.2 Eingangsdokumente

- 1) Software-Anforderungsspezifikation;
- 2) Spezifikation der System-Sicherheitsanforderungen;
- 3) System-Architekturbeschreibung;
- 4) Software-Qualitätssicherungsplan.

9.3 Ausgangsdokumente

Software-Architekturspezifikation.

9.4 Anforderungen

9.4.1 Die vorgeschlagene Software-Architektur muss durch den Software-Lieferanten und/oder Entwickler erstellt und in der Software-Architekturspezifikation detailliert beschrieben werden.

9.4.2 Die Software-Architekturspezifikation muss eine möglichst einfache Umsetzung der Software-Anforderungsspezifikation bei Einhaltung der erforderlichen Software-Sicherheitsanforderungsstufe berücksichtigen.

9.4.3 Die Software-Architekturspezifikation muss die Bedeutung aller Wechselwirkungen zwischen Hardware und Software identifizieren, bewerten und ausführlich beschreiben. Wie in [EN 50126](#) und [EN 50129](#) gefordert, müssen die vorläufigen Untersuchungen der Wechselwirkungen zwischen Hardware und Software in der System-Sicherheitsanforderungsspezifikation niedergelegt worden sein.

9.4.4 Die Software-Architekturspezifikation muss alle Software-Komponenten identifizieren und für diese Komponenten Folgendes kenntlich machen:

- i) ob diese Komponenten neu, bereits vorhanden oder proprietär sind;
- ii) ob diese Komponenten vorher validiert wurden und, wenn zutreffend, ihre Validierungsbedingungen;
- iii) die Software-Sicherheitsanforderungsstufe der Komponente.

9.4.5 Die Verwendung von COTS-Software muss folgenden Einschränkungen unterworfen werden:

- i) Für Software-Sicherheitsanforderungsstufe 0 ist die Verwendung von COTS-Software ohne weitere Vorsichtsmaßnahmen zu akzeptieren.
- ii) Wenn COTS-Software für die Software-Sicherheitsanforderungsstufen 1 oder 2 eingesetzt werden soll, muss sie in den Software-Validierungsprozess eingebunden werden.
- iii) Wenn COTS-Software für die Software-Sicherheitsanforderungsstufen 3 oder 4 eingesetzt werden soll, müssen folgende Vorsichtsmaßnahmen getroffen werden:
 - a) Die COTS-Software muss in die Validierungstests einbezogen werden.
 - b) Eine Analyse möglicher Fehlfunktionen muss durchgeführt werden.
 - c) Eine Vorgehensweise muss festgelegt werden, um Fehlfunktionen der COTS-Software zu erkennen und das System vor diesen Fehlern zu schützen.
 - d) Die Schutzstrategie muss Gegenstand der Validierungstests sein.
 - e) Fehleraufzeichnungen müssen vorhanden sein und ausgewertet werden.
 - f) Soweit praktikabel, dürfen nur die einfachsten Funktionen der COTS-Software benutzt werden.

9.4.6 Wenn vorher entwickelte Software als Teil des Entwurfs verwendet werden soll, muss diese eindeutig gekennzeichnet und dokumentiert sein. Die Software-Architekturspezifikation muss die Eignung der Software zur Erfüllung der Software-Anforderungsspezifikation und der Software-Sicherheitsanforderungsstufe darlegen. Die Auswirkungen von Änderungen der Software auf das übrige System muss sorgfältig betrachtet werden, um zu entscheiden, ob diese eine Wiederholungsprüfung und Neubegutachtung erfordern. Es muss klar ersichtlich sein, dass Schnittstellenspezifikationen mit anderen nicht neu verifizierten, neu validierten oder neu begutachteten Modulen erfüllt werden.

9.4.7 Wenn immer möglich, müssen vorhandene verifizierte Software-Module, die nach dieser Norm entwickelt wurden, zum Entwurf verwendet werden.

9.4.8 Die Software-Architektur muss den sicherheitsrelevanten Teil der Anwendung minimieren.

9.4.9 Wo die Software aus Teilen unterschiedlicher Software-Sicherheitsanforderungsstufen besteht, müssen alle Softwareteile so betrachtet werden, als würden sie der höchsten Software-Anforderungsstufe angehören, es sei denn, die Unabhängigkeit zwischen den Teilen mit höherer bzw. niedrigerer Software-Sicherheitsanforderungsstufe ist klar ersichtlich. Dass dies so ist, muss in der Software-Architekturspezifikation festgehalten werden.

9.4.10 Die Software-Architekturspezifikation muss die Strategie für die Software-Entwicklung in dem durch die Software-Sicherheitsanforderungsstufe festgelegten Umfang definieren. Die Software-Architekturspezifikation muss so formuliert und strukturiert sein, dass sie:

- i) vollständig, klar, genau, eindeutig, verifizierbar, testbar, wartbar und leicht umsetzbar ist;
- ii) zur Software-Anforderungsspezifikation rückverfolgbar ist.

9.4.11 Die Software-Architekturspezifikation muss die Abwägung zwischen fehlervermeidenden und fehlerbeherrschenden Strategien begründen.

9.4.12 Die Software-Architekturspezifikation muss zeigen, dass die gewählten Techniken und Maßnahmen einen integrierten Satz bilden, der der Software-Anforderungsspezifikation auf der geforderten Software-Sicherheitsanforderungsstufe genügt.

10 Software-Entwurf und Implementierung

10.1 Ziele

10.1.1 Entwurf und Implementierung von Software einer definierten Software-Sicherheitsanforderungsstufe gemäß der Software-Anforderungsspezifikation und der Software-Architekturspezifikation.

10.1.2 Software, die analysierbar, testbar, verifizierbar und wartbar ist. Das Modultesten ist ebenfalls Teil dieser Phase. Da Verifikation und Testen bei der Validierung ein kritisches Element darstellen, muss der Verifikation und den Testerfordernissen schon beim Entwurf und der Entwicklung besondere Aufmerksamkeit geschenkt werden, um sicherzustellen, dass das resultierende System und seine Software von Anfang an leicht testbar sind.

10.1.3 Auswahl eines für die geforderte Software-Sicherheitsanforderungsstufe angemessenen Satzes von Werkzeugen, einschließlich Sprachen und Compilern, die während dem ganzen Lebenszyklus der Software die Verifikation, Validierung, Begutachtung und Wartung unterstützen.

10.1.4 Durchführung der Softwareintegration.

10.2 Eingangsdokumente

- 1) Software-Anforderungsspezifikation;
- 2) Software-Architekturspezifikation;
- 3) Software-Qualitätssicherungsplan.

10.3 Ausgangsdokumente

- 1) Software-Entwurfsspezifikation;
- 2) Software-Modulentwurfsspezifikation;
- 3) Software-Modultestspezifikation;
- 4) Software-Quellcode und unterstützende Dokumentation;
- 5) Software-Modultestbericht.

10.4 Anforderungen

10.4.1 Die Software-Anforderungsspezifikation und die Software-Architekturspezifikation müssen, wenn auch noch nicht unbedingt in endgültiger Form, vor dem Beginn des Entwurfsprozesses zur Verfügung stehen.

10.4.2 Der Umfang und die Komplexität der entwickelten Software muss so klein wie möglich gehalten werden.

10.4.3 Die Software-Entwurfsspezifikation muss den Software-Entwurf beschreiben, welcher auf einer Zerlegung in Module basiert, wobei jedes Modul eine Software-Modulentwurfsspezifikation und eine Software-Modultestspezifikation besitzt.

10.4.4 Die Software-Designspezifikation muss folgende Punkte behandeln:

- i) Softwarebestandteile rückverfolgt auf die Software-Architektur und ihre Sicherheitsanforderungsstufe;
- ii) Schnittstellen der Softwarebestandteile mit der Umgebung;
- iii) Schnittstellen zwischen den Softwareteilen;
- iv) Datenstrukturen;
- v) Aufteilung der Anforderungen auf die Bestandteile;
- vi) wesentliche Algorithmen und Abläufe;
- vii) Diagramme.

10.4.5 Die Software-Modulentwurfsspezifikation muss folgende Punkte ansprechen (eine Software-Modulentwurfsspezifikation je Modul):

- i) Identifizierung aller Software-Komponenten der niedersten Ebene (in der vorliegenden Norm „Module“ genannt), zurückverfolgt auf die darüber liegende Ebene;
- ii) ihre exakten Schnittstellen mit der Umgebung und anderen Modulen mit detaillierter Beschreibung der Eingänge und Ausgänge;
- iii) ihre Sicherheitsanforderungsstufe;
- iv) genaue Darstellung der Algorithmen und Datenstrukturen.

Jede Software-Modulentwurfsspezifikation sollte in sich schlüssig aufgebaut sein und die Codierung des betreffenden Moduls erlauben.

10.4.6 Jedes Software-Modul muss lesbar, verständlich und testbar sein.

10.4.7 Eine angemessene Auswahl von Werkzeugen, einschließlich Entwurfsmethoden, Sprachen und Compilern muss für die geforderte Software-Sicherheitsanforderungsstufe über den gesamten Lebenszyklus der Software ausgewählt werden.

10.4.8 Falls möglich, müssen automatische Testwerkzeuge und integrierte Entwicklungswerkzeuge verwendet werden. Die Anforderungen des Verifizierers und des Validierers müssen dabei berücksichtigt werden.

10.4.9 Entsprechend dem durch die Software-Sicherheitsanforderungsstufe festgelegten Umfang muss die ausgewählte Programmiersprache einen Übersetzer/Compiler besitzen, der eines der folgenden Merkmale aufweist:

- i) ein „Validierungszertifikat“ nach einer anerkannten nationalen/internationalen Norm;
- ii) ein Gutachten, das die Tauglichkeit für den Anwendungszweck bescheinigt;
- iii) einen zusätzlichen signaturbasierenden Kontrollmechanismus zur Erkennung von Übersetzungsfehlern.

10.4.10 Die ausgewählte Programmiersprache muss die folgenden Anforderungen erfüllen:

- i) Die gewählte Sprache muss Merkmale enthalten, die das Erkennen von Programmierfehlern erleichtern.
- ii) Die gewählte Programmiersprache muss Merkmale unterstützen, die zu dem Entwurfsverfahren passen.

10.4.11 Wenn 10.4.10 nicht erfüllt werden kann, muss eine Begründung für eine alternative Sprache gegeben werden und deren Tauglichkeit für die Anwendung muss in der Software-Architekturspezifikation oder im Software-Qualitätssicherungsplan detailliert aufgezeichnet werden.

10.4.12 Es müssen Codierstandards entwickelt und für die Entwicklung der gesamten Software verwendet werden. Auf diese muss im Software-Qualitätssicherungsplan verwiesen werden (siehe [15.4.5](#)).

10.4.13 Die Codierstandards müssen gute Programmierpraktiken spezifizieren, unsichere Sprachmerkmale verbieten und Verfahren für die Quellcodedokumentation beschreiben. Im Quellcode muss jedes Software-Modul mindestens folgende (nicht erschöpfende) Informationen enthalten:

- Autor;
- Konfigurationsgeschichte;
- Kurzbeschreibung.

Die Verwendung einer Standardform für diese Informationen wird empfohlen. Sie sollte für alle Module gleich sein.

10.4.14 Software-Modultests: Jedes Modul muss eine Software-Modultestspezifikation besitzen, gegen die das Modul zu testen ist. Diese Tests müssen zeigen, dass jedes Modul seine beabsichtigte Funktion erfüllt. Die Software-Modultestspezifikation muss den erforderlichen Testüberdeckungsgrad definieren.

Es muss ein Software-Modultestbericht mit folgenden Merkmalen erstellt werden:

- i) eine Aussage über die Testergebnisse und ob jedes Modul die Anforderungen seiner Software-Modul-entwurfsspezifikation erfüllt hat;
- ii) für jedes Modul muss eine Aussage über die Testüberdeckung gemacht werden, die zeigt, dass jede Quellcodezeile mindestens einmal ausgeführt wurde;
- iii) der Testbericht muss in auditierbarer Form vorliegen;
- iv) die Testfälle und ihre Ergebnisse müssen zwecks nachfolgender Analyse in maschinenlesbarer Form aufgezeichnet werden; die Tests sollten wiederholbar sein und, soweit praktikabel, mit automatischen Hilfsmitteln ausgeführt werden.

Die Überprüfung, dass das Modul seine Testspezifikation korrekt erfüllt hat, ist eine Verifikationstätigkeit (siehe [Abschnitt 11](#)).

10.4.15 Das gewählte Entwurfsverfahren muss Merkmale haben, die entsprechend der geforderten Software-Sicherheitsanforderungsstufe Folgendes unterstützen:

- i) Abstraktion, Modularität und andere Eigenschaften, die die Komplexität kontrollieren;

- ii) die klare und genaue Darstellung von:
 - Funktionalität;
 - Informationsfluss zwischen den Komponenten;
 - Reihenfolge und zeitbezogene Information;
 - Parallelverarbeitung;
 - Datenstrukturen und -eigenschaften;
- iii) Verständlichkeit für den Menschen;
- iv) Verifikation und Validierung.

10.4.16 Das gewählte Entwurfsverfahren muss Eigenschaften besitzen, die die Softwarewartung erleichtern. Solche Eigenschaften müssen Modularität, Information-Hiding und Einkapselung einschließen.

10.4.17 Die Integration von Software-Modulen muss ein Prozess der vorwärts gerichteten Kombination einzelner und vorher getesteter Software-Module zu einem verbundenen Ganzen (oder in eine Anzahl verbundener Teilsysteme) sein, damit die Modulschnittstellen und die zusammengebaute Software vor der Integration und dem Test des Systems angemessen geprüft werden können.

10.4.18 Im Zusammenhang dieser Norm und in dem der spezifizierten Software-Sicherheitsanforderungsstufe angemessenen Umfang muss die Verfolgbarkeit speziell Folgendes behandeln:

- i) Verfolgbarkeit der Anforderungen zum Entwurf oder zu anderen Objekten, die sie erfüllen.
- ii) Verfolgbarkeit von Entwurfsobjekten zu den Implementierungsobjekten, welche sie umsetzen.

Das Ergebnis des Verfolgbarkeits-Prozesses muss dem Konfigurationsmanagement unterzogen werden.

11 Software-Verifikation und Testen

11.1 Ziele

Testen und Bewerten der Produkte einer bestimmten Phase in dem durch die Software-Sicherheitsanforderungsstufe festgelegten Umfang, um die Korrektheit und die Konsistenz hinsichtlich der Produkte und Standards zu gewährleisten, die eingangsseitig dieser Phase zur Verfügung stehen.

11.2 Eingangsdokumente

- 1) System-Anforderungsspezifikation;
- 2) Spezifikation der System-Sicherheitsanforderungen;
- 3) Software-Anforderungsspezifikation;
- 4) Software-Anforderungstestspezifikation;
- 5) Software-Architekturspezifikation;
- 6) Software-Entwurfsspezifikation;
- 7) Software-Modulentwurfsspezifikation;
- 8) Software-Modultestspezifikation;
- 9) Software-Quellcode und unterstützende Dokumentation;
- 10) Software-Qualitätssicherungsplan;
- 11) Software-Modultestbericht.

11.3 Ausgangsdokumente

- 1) Software-Verifikationsplan;

- 2) Software-Anforderungsverifikationsbericht;
- 3) Software-Architektur- und Entwurfsverifikationsbericht;
- 4) Software-Modulverifikationsbericht;
- 5) Software-Quellcodeverifikationsbericht;
- 6) Software-Integrationstestplan;
- 7) Software-Integrationstestbericht.

11.4 Anforderungen

11.4.1 Es muss ein Software-Verifikationsplan erstellt werden, damit die Verifikationstätigkeiten geeignet ausgerichtet und besondere Anforderungen an den Entwurf oder die Verifikation angemessen berücksichtigt werden können. Während der Entwicklung (und abhängig von der Größe des Systems) darf der Plan in eine Anzahl von Tochterdokumenten unterteilt und selbstverständlich erweitert werden, sobald die detaillierten Erfordernisse der Verifikation klarer werden.

11.4.2 Der Software-Verifikationsplan muss alle die Kriterien, Techniken und Werkzeuge beschreiben, die im Verifikationsprozess der jeweiligen Phase genutzt werden.

11.4.3 Der Software-Verifikationsplan muss die Aktivitäten beschreiben, die ausgeführt werden müssen, um Korrektheit und Konsistenz im Hinblick auf die Produkte und Standards zu gewährleisten, die eingangsseitig der Phase zur Verfügung stehen.

11.4.4 Der Software-Verifikationsplan muss folgende Punkte ansprechen:

- i) die Auswahl der Verifikationsstrategien und -techniken. Um eine unangemessene Komplexität in der Begutachtung der Verifikations- und Testtätigkeit zu vermeiden, sollten bevorzugt solche Testfälle und -verfahren usw. ausgewählt werden, die in sich leicht analysierbar sind;
- ii) die Auswahl und den Gebrauch der Software-Testeinrichtung;
- iii) die Auswahl und die Dokumentation der Verifikationsaktivitäten;
- iv) die Bewertung der gewonnenen Verifikationsergebnisse;
- v) die Bewertung hinsichtlich der Zuverlässigkeitsanforderungen;
- vi) die Rollen und Verantwortlichkeiten der in den Testprozess einbezogenen Personen;
- vii) den Grad der zu erreichenden Testabdeckung.

11.4.5 Der Software-Integrationstestplan muss Folgendes dokumentieren:

- i) Testfälle und Testdaten;
- ii) Arten von durchzuführenden Tests;
- iii) Testumgebung, -werkzeuge, -konfiguration und -programme;
- iv) Testkriterien, nach denen der Testabschluss beurteilt wird.

11.4.6 In jeder Entwicklungsphase muss gezeigt werden, dass die Anforderungen an Funktion, Zuverlässigkeit, Leistung und Sicherheit erreicht werden.

11.4.7 Die Verifikation muss von einer unabhängigen Stelle durchgeführt werden, und zwar in dem durch die Software-Sicherheitsanforderungsstufe festgelegten Umfang (siehe [Bild 5](#)).

11.4.8 Ein Test, der nicht vollständig dokumentiert und vom Entwerfer vor der Verifikation durchgeführt wird, darf nicht als Teil der Verifikation betrachtet werden.

11.4.9 Die Ergebnisse jeder Verifikation müssen in einer auditfähigen Form festgehalten werden, wie es im Software-Verifikationsplan definiert oder referenziert ist.

11.4.10 Nach jeder Verifikationstätigkeit muss ein Verifikationsbericht erstellt werden, in dem angegeben wird, ob die Software die Verifikation bestanden hat oder es müssen die Gründe für das Nichtbestehen genannt werden. Der Verifikationsbericht muss folgende Punkte berücksichtigen:

- i) Punkte, die nicht mit der Software-Anforderungsspezifikation, der Software-Entwurfsspezifikation oder den Software-Modulentwurfsspezifikationen übereinstimmen;
- ii) Punkte, die nicht mit dem Software-Qualitätssicherungsplan übereinstimmen;
- iii) Module, Daten, Strukturen und Algorithmen, die dem Problem schlecht angepasst sind;
- iv) entdeckte Fehler oder Mängel;
- v) Identität und Konfiguration der verifizierten Teile.

11.4.11 Software-Anforderungsverifikation: Nachdem die Software-Anforderungsspezifikation erstellt worden ist, muss eine Verifikation zu folgenden Punkten erfolgen:

- i) Eignung der Software-Anforderungsspezifikation, die in der System-Anforderungsspezifikation, der Spezifikation der System-Sicherheitsanforderungen und dem Software-Qualitätssicherungsplan festgelegten Anforderungen zu erfüllen.
- ii) Eignung der Software-Anforderungstestspezifikation als Test der Software-Anforderungsspezifikation.
- iii) Konsistenz der Software-Anforderungsspezifikation in sich.

Die Ergebnisse müssen in einem Software-Anforderungsverifikationsbericht aufgezeichnet werden.

11.4.12 Software-Architektur- und Entwurfsverifikation: Nachdem die Software-Architekturspezifikation und die Software-Entwurfsspezifikation erstellt sind, muss eine Verifikation zu folgenden Punkten erfolgen:

- i) Eignung der Software-Architekturspezifikation und der Software-Entwurfsspezifikation, die Software-Anforderungsspezifikation zu erfüllen.
- ii) Eignung der Software-Entwurfsspezifikation, die Software-Anforderungsspezifikation hinsichtlich Konsistenz und Vollständigkeit zu erfüllen.
- iii) Eignung des Software-Integrationstestplans als Satz von Testfällen für die Software-Architekturspezifikation und die Software-Entwurfsspezifikation.
- iv) Konsistenz der Software-Architektur- und Entwurfsspezifikation in sich.

Die Ergebnisse müssen in einem Software-Architektur- und Entwurfsverifikationsbericht aufgezeichnet werden.

11.4.13 Software-Modulverifikation: Nachdem alle Software-Modulentwurfsspezifikationen erstellt worden sind, muss eine Verifikation zu folgenden Punkten erfolgen:

- i) Eignung der Software-Modulentwurfsspezifikation, die Software-Entwurfsspezifikation zu erfüllen;
- ii) Eignung der Software-Modultestspezifikation als Satz von Testfällen für die Software-Modulentwurfsspezifikation;
- iii) Zerlegung der Software-Entwurfsspezifikation in Software-Module und in die Software-Modulentwurfsspezifikationen mit Bezug auf:
 - Erreichbarkeit der erforderlichen Leistung;
 - Testbarkeit für die weitere Verifikation;
 - Lesbarkeit durch die Entwicklungs- und Verifikationsarbeitsgruppe und
 - Wartbarkeit, um eine Weiterentwicklung zu ermöglichen;
- iv) Eignung der Software-Modultestberichte als Aufzeichnung, dass die Tests in Übereinstimmung mit der Software-Modultestspezifikation durchgeführt wurden.

Die Ergebnisse müssen in einem Software-Modulverifikationsbericht aufgezeichnet werden.

11.4.14 Software-Quellcodeverifikation: Um eine Übereinstimmung mit der Software-Modulentwurfsspezifikation und dem Software-Qualitätssicherungsplan zu gewährleisten, muss der Software-Quellcode in dem durch die Software-Sicherheitsanforderungsstufe festgelegten Umfang verifiziert werden. Dies muss eine Prüfung auf korrekte Anwendung der Codierstandards einschließen.

Die Ergebnisse müssen in einem Software-Quellcodeverifikationsbericht aufgezeichnet werden.

11.4.15 Es muss ein Software-Integrationstestbericht erstellt werden, der Folgendes enthält:

- i) Es muss ein Software-Integrationstestbericht erstellt werden, der die Testergebnisse darstellt und zeigt, ob die Ziele und Kriterien des Software-Integrationstestplans erfüllt wurden. Wenn ein Fehler vorliegt, müssen die Fehlergründe aufgezeichnet werden.
- ii) Der Software-Integrationstestplan muss in einer auditierbaren Form vorliegen.
- iii) Die Testfälle und ihre Ergebnisse müssen aufgezeichnet werden, vorzugsweise in maschinenlesbarer Form, um eine Nachanalyse zu ermöglichen.
- iv) Die Tests sollten wiederholbar sein und, falls praktikabel, mit automatischen Hilfsmitteln durchgeführt werden.
- v) Identität und Konfiguration der verifizierten Teile.

11.4.16 Bezüglich der Software/Hardwareintegration siehe [12.4.8](#).

12 Software/Hardware-Integration

12.1 Ziele

12.1.1 Zeigen, dass die Software und die Hardware richtig zusammenwirken, um ihre vorgesehenen Funktionen auszuführen.

12.1.2 Kombination der Software mit der Hardware und dabei sicherstellen, dass sie kompatibel sind, um die Spezifikation der System-Sicherheitsanforderungen sowie die Erfordernisse für die vorgesehene Software-Sicherheitsanforderungsstufe zu erfüllen.

12.2 Eingangsdokumente

- 1) System-Anforderungsspezifikation;
- 2) Spezifikation der System-Sicherheitsanforderungen;
- 3) System-Architekturbeschreibung;
- 4) Software-Anforderungsspezifikation;
- 5) Software-Anforderungstestspezifikation;
- 6) Software-Architekturspezifikation;
- 7) Software-Entwurfsspezifikation;
- 8) Software-Modulentwurfsspezifikation;
- 9) Software-Modultestspezifikation;
- 10) Software-Quellcode und unterstützende Dokumentation;
- 11) Hardware-Dokumentation.

12.3 Ausgangsdokumente

- 1) Software/Hardware-Integrationstestplan;
- 2) Software/Hardware-Integrationstestbericht.

12.4 Anforderungen

12.4.1 Für Software-Sicherheitsanforderungsstufen größer Null wird frühzeitig im Entwicklungszyklus ein Software/Hardware-Integrationstestplan erstellt, damit die Integrationsaktivitäten sinnvoll ausgerichtet und besondere Anforderungen an den Entwurf oder die Integration angemessen berücksichtigt werden können. Während der Entwicklung (und abhängig von der Größe des Systems) darf der Plan in eine Anzahl von Tochterdokumenten unterteilt und selbstverständlich erweitert werden, sobald die Hardware- und Software-Entwürfe herausgearbeitet sind und die detaillierten Erfordernisse der Integration deutlicher werden.

12.4.2 Im Software/Hardware-Integrationstestplan müssen folgende Punkte dokumentiert werden:

- i) Testfälle und Testdaten;
- ii) auszuführende Testarten;
- iii) Testumgebung, einschließlich der Werkzeuge, der Hilfssoftware und Konfigurationsbeschreibung;
- iv) Testkriterien, an denen die Vollständigkeit der Tests beurteilt werden kann.

12.4.3 Der Software/Hardware-Integrationstestplan muss zwischen jenen Aktivitäten unterscheiden, die vom Entwickler im eigenen Haus ausgeführt werden können, und jenen, die einen Zugang zur Anlage des Betreibers erfordern.

12.4.4 Der Software/Hardware-Integrationstestplan muss zwischen folgenden Aktivitäten unterscheiden:

- i) Portierung der Software auf die Zielhardware;
- ii) Systemintegration.

12.4.5 Werkzeuge und Hilfsmittel, die im Software/Hardware-Integrationstestplan beschrieben werden, sollten zum frühest möglichen Termin verfügbar sein.

12.4.6 Während der Software/Hardware-Integration muss jede Modifizierung oder Veränderung des integrierten Systems einer Auswirkungsstudie unterzogen werden, in der alle betroffenen Module und die erforderlichen Aktivitäten zur Reverification identifiziert werden müssen.

12.4.7 Die Testfälle und deren Ergebnisse müssen aufgezeichnet werden, für nachfolgende Analysen vorzugsweise in maschinenlesbarer Form.

12.4.8 Es muss ein Software/Hardware-Integrationstestbericht folgenden Inhalts erstellt werden:

- i) Der Software/Hardware-Integrationstestbericht muss die Testergebnisse aufführen und ob die Ziele und Kriterien des Software/Hardware-Integrationstestplans erfüllt wurden. Fehler sind aufzuzeichnen.
- ii) Der Software/Hardware-Integrationstestbericht muss eine auditable Form haben.
- iii) Testfälle und ihre Ergebnisse müssen aufgezeichnet werden, für nachfolgende Auswertungen bevorzugt in maschinenlesbarer Form.
- iv) Der Software/Hardware-Integrationstestbericht muss auch die Zustimmung des Verifizierers zeigen, dass der Software/Hardware-Integrationstestbericht eine angemessene Aufzeichnung der ausgeführten Tests im Sinne des Software/Hardware-Integrationstestplans ist.
- v) Der Software/Hardware-Integrationstestbericht muss die Identität und Konfiguration aller betroffenen Bestandteile dokumentieren.

13 Software-Validierung

13.1 Ziele

Analyse und Test des integrierten Systems, um die Übereinstimmung mit der Software-Anforderungsspezifikation sicherzustellen, wobei die besondere Betonung auf den funktionalen und Sicherheitsaspekten entsprechend der Software-Sicherheitsanforderungsstufe liegt.

13.2 Eingangsdokumente

- 1) Software-Anforderungsspezifikation;
- 2) alle Hardware- und Softwareunterlagen;
- 3) Spezifikation der System-Sicherheitsanforderungen.

13.3 Ausgangsdokumente

- 1) Software-Validierungsplan;
- 2) Software-Validierungsbericht.

13.4 Anforderungen

13.4.1 Analyse und Testen müssen die Hauptaktivitäten der Validierung sein.

13.4.2 Simulation und Modellierung dürfen verwendet werden, um den Validierungsprozess zu ergänzen.

13.4.3 Es muss ein Software-Validierungsplan erstellt und in geeigneter Form detailliert dokumentiert werden.

13.4.4 Der Software-Validierungsplan muss entwickelt, ausgeführt und die Ergebnisse durch eine unabhängige Stelle in dem durch die Software-Sicherheitsanforderungsstufe festgelegten Umfang bewertet werden.

13.4.5 Wenn es aufgrund der Software-Sicherheitsanforderungsstufe notwendig ist, müssen Geltungsbereich und Inhalt des Software-Validierungsplans mit dem Gutachter vereinbart werden. Diese Vereinbarung muss auch eine Aussage bez. der Anwesenheit des Gutachters während der Tests enthalten.

13.4.6 Der Software-Validierungsplan muss eine Zusammenfassung enthalten, die die gewählte Validierungsstrategie begründet. Diese Begründung muss, entsprechend der geforderten Software-Sicherheitsanforderungsstufe, folgende Betrachtungen einschließen:

- i) manuelle oder automatische Techniken oder beides;
- ii) statische oder dynamische Techniken oder beides;
- iii) analytische oder statistische Techniken oder beides.

13.4.7 Der Software-Validierungsplan muss die notwendigen Schritte aufführen, um zu demonstrieren, dass die

- Software-Anforderungsspezifikation;
- Software-Architekturspezifikation;
- Software-Entwurfsspezifikation;
- Software-Modulentwurfsspezifikation

angemessen sind, um die Sicherheitsanforderungen zu erfüllen, die in der Spezifikation der System-Sicherheitsanforderungen aufgestellt wurden. Der Validierer muss überprüfen, dass der Verifikationsprozess vollständig ausgeführt wurde.

13.4.8 Die für die Validierung verwendete Messeinrichtung muss angemessen kalibriert sein. Es muss dargestellt werden, dass alle für die Validierung benutzten Hardware- oder Software-Werkzeuge für den Zweck geeignet sind.

13.4.9 Die Software muss durch Nachbildung der Eingangssignale bei normalem Betrieb, bei vorhersehbaren besonderen Ereignissen und unter unerwünschten Bedingungen, die eine Systemreaktion erfordern, ausgeführt werden.

13.4.10 Die Ergebnisse der Validierung müssen in auditfähiger Form im Software-Validierungsbericht dokumentiert werden.

13.4.11 Nach Abschluss der Hardware/Software-Integration muss ein Software-Validierungsbericht folgenden Inhalts erstellt werden:

- i) Er muss eine Aussage enthalten, ob die Ziele und Kriterien des Software-Validierungsplans erfüllt wurden. Wenn eine Fehlfunktion vorliegt, so muss diese aufgezeichnet werden.
- ii) Er muss eine Aussage bezüglich der Testergebnisse enthalten und ob die Gesamtsoftware auf dem Zielrechner die in der Software-Anforderungsspezifikation aufgestellten Anforderungen erfüllt.
- iii) Es muss eine Bewertung der Testüberdeckung bezüglich der Anforderungen in der Software-Anforderungsspezifikation erfolgen.
- iv) Der Software-Validierungsbericht muss in auditierbarer Form vorliegen.
- v) Die Testfälle und ihre Ergebnisse müssen für eine nachfolgende Analyse in maschinenlesbarer Form aufgezeichnet sein.
- vi) Die Tests sollten wiederholbar und, wenn praktisch möglich, mit automatischen Hilfsmitteln ausführbar sein.

13.4.12 Der Software-Validierungsbericht muss die Identität und Konfiguration folgender Punkte dokumentieren:

- i) verwendete Hardware und Software;
- ii) verwendete Geräte;
- iii) Kalibrierung der Geräte;
- iv) verwendete Simulationsmodelle;
- v) festgestellte Mängel;
- vi) ausgeführte Korrekturmaßnahmen.

13.4.13 Alle gefundenen Mängel, einschließlich entdeckter Fehler, müssen klar in einem eigenen Abschnitt des Software-Validierungsberichts aufgeführt und in einer Mitteilung enthalten sein, die mit der gelieferten Software herausgegeben wird.

13.4.14 Die Software muss gegen die Software-Anforderungstestspezifikation getestet werden. Diese Tests müssen zeigen, dass alle Anforderungen der Software-Anforderungsspezifikation richtig ausgeführt werden.

Die Ergebnisse müssen in einem Software-Validierungsbericht aufgezeichnet werden.

14 Software-Begutachtung

14.1 Ziele

Bewertung, ob die Prozesse des Lebenszyklus und die sich daraus ergebenden Produkte den Schluss zulassen, dass die Software die festgelegte Software-Sicherheitsanforderungsstufe aufweist und für ihre vorgesehene Anwendung tauglich ist.

14.2 Eingangsdokumente

- 1) Spezifikation der System-Sicherheitsanforderungen;
- 2) komplette Hardware- und Software-Dokumentation.

14.3 Ausgangsdokumente

Software-Gutachten.

14.4 Anforderungen

14.4.1 Für Software der Sicherheitsanforderungsstufe 0:

- i) Der Gutachter ist nur gefordert, um zu bestätigen, dass dies die geeignete Software-Sicherheitsanforderungsstufe ist.
- ii) Es ist auch möglich, dass sich Lieferant und Anwender in der Angebotsphase über diese Anforderungsstufe abstimmen.

14.4.2 Software, für die bereits ein Gutachten eines anderen Gutachters existiert, muss nicht komplett neu begutachtet werden. Der zweite Gutachter muss prüfen, dass die Software die geforderte Software-Sicherheitsanforderungsstufe erfüllt und dass sie für die vorgesehene Anwendung auf dem beabsichtigten Zielrechner tauglich ist.

14.4.3 Der Gutachter muss Zugang zum Entwurfs- und Entwicklungsprozess und zu allen Projektunterlagen haben.

14.4.4 Die Begutachtung der Software muss durch einen Gutachter vorgenommen werden, der von der Entwurfsarbeitsgruppe unabhängig ist.

14.4.5 Der Gutachter muss feststellen, ob die Software des Systems für den vorgesehenen Zweck tauglich ist und die aus der Spezifikation der System-Sicherheitsanforderungen abgeleiteten Sicherheitsanforderungen richtig abdeckt.

14.4.6 Gemäß der festgelegten Software-Sicherheitsanforderungsstufe muss der Gutachter entscheiden, ob geeignete Verfahren ausgewählt und in jeder Phase des Software-Lebenszyklus angewendet wurden.

14.4.7 Wenn es gemäß Software-Sicherheitsanforderungsstufe gefordert ist, muss der Gutachter dem Geltungsbereich und Inhalt des Software-Validierungsplans zustimmen. Diese Zustimmung muss auch eine Angabe enthalten bez. der Anwesenheit des Gutachters bei den Tests.

14.4.8 Der Gutachter darf zusätzliche Verifikations- und Validierungsschritte nach seiner Wahl verlangen.

14.4.9 Der Gutachter muss für jede Überprüfung einen Bericht erstellen, der das Ergebnis der Begutachtung detailliert darstellt.

14.4.10 Wenn nach Meinung des Gutachters die Software für die vorgesehene Anwendung tauglich ist, muss das abschließende Software-Gutachten eine Angabe darüber enthalten, dass die Software-Sicherheitsanforderungsstufe durch die Software erreicht wird.

14.4.11 Falls die Software für ihren Einsatzzweck nicht tauglich ist oder die notwendige Software-Sicherheitsanforderungsstufe nicht erreicht hat, darf der Gutachter im Software-Gutachten die Mängel nur auflisten und keine technische Lösung angeben.

15 Software-Qualitätssicherung

15.1 Ziele

15.1.1 Identifizieren, Überwachen und Steuern aller technischen und Managementaktivitäten, die erforderlich sind, um sicherzustellen, dass die Software die erforderliche Qualität erreicht. Dies ist notwendig, um die geforderte qualitative Abwehr gegen systematische Fehler zu gewährleisten und um sicherzustellen, dass ein Auditwesen eingerichtet werden kann, um die effiziente Durchführung von Verifikations- und Validierungsmaßnahmen zu ermöglichen.

15.1.2 Beweisen, dass die oben genannten Aktivitäten ausgeführt wurden.