

	DIN EN 61508-3 (VDE 0803-3)	
	Diese Norm ist zugleich eine VDE-Bestimmung im Sinne von VDE 0022. Sie ist nach Durchführung des vom VDE-Präsidium beschlossenen Genehmigungsverfahrens unter der oben angeführten Nummer in das VDE-Vorschriftenwerk aufgenommen und in der „etz Elektrotechnik + Automation“ bekannt gegeben worden.	
ICS 35.240.50 Ersatz für DIN EN 61508-3 (VDE 0803-3):2002-12 Siehe Anwendungsbeginn Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme – Teil 3: Anforderungen an Software (IEC 61508-3:2010); Deutsche Fassung EN 61508-3:2010 Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 3: Software requirements (IEC 61508-3:2010); German version EN 61508-3:2010 Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité – Partie 3: Exigences concernant les logiciels (CEI 61508-3:2010); Version allemande EN 61508-3:2010 Gesamtumfang 133 Seiten DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE		

Anwendungsbeginn

Anwendungsbeginn für die von CENELEC am 2010-05-01 angenommene Europäische Norm als DIN-Norm ist 2011-02-01.

Daneben darf DIN EN 61508-3 (VDE 0803-3):2002-12 noch bis 2013-05-01 angewendet werden.

Nationales Vorwort

Vorausgegangener Norm-Entwurf: E DIN EN 61508-3 (VDE 0803-3):2009-06.

Für diese Norm ist das nationale Arbeitsgremium GK 914 „Funktionale Sicherheit elektrischer, elektronischer und programmierbarer elektronischer Systeme (E, E, PES) zum Schutz von Personen und Umwelt“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE (www.dke.de) zuständig.

Die enthaltene IEC-Publikation wurde vom SC 65A „System aspects“ erarbeitet.

Das IEC-Komitee hat entschieden, dass der Inhalt dieser Publikation bis zu dem Datum (maintenance result date) unverändert bleiben soll, das auf der IEC-Website unter „<http://webstore.iec.ch>“ zu dieser Publikation angegeben ist. Zu diesem Zeitpunkt wird entsprechend der Entscheidung des Komitees die Publikation

- bestätigt,
- zurückgezogen,
- durch eine Folgeausgabe ersetzt oder
- geändert.

Änderungen

Gegenüber DIN EN 61508-3 (VDE 0803-3):2002-12 wurden folgende Änderungen vorgenommen:

- a) Einführung des Begriffs „Element“, dieser Begriff löst die „Komponente“ ab;
- b) Einführung eines neuen Anhangs C „Eigenschaften der systematischen Sicherheitsintegrität der Software („Properties“). Hierin werden Akzeptanzkriterien für Eigenschaften von Verfahren dargelegt, die im Lebenszyklus Verwendung finden;
- c) Einführung eines neuen Anhangs G „Datengesteuerte Systeme“. Über die Funktion und Konfigurierbarkeit wird auf den Komplexitätsgrad des Systems geschlossen. Es werden sechs Kategorien eingeführt, die über abnehmende Einschränkung der Beeinflussung des Systems durch die Konfiguration und die Funktionalität unterschieden werden.
- d) Die Architektur und das Design objektorientierter Software wurde erstmalig erwähnt.
- e) Einführung eines neuen Anhangs D „Sicherheitshandbuch“, um wiederverwendbare Software zu nutzen.
- f) Einführung eines neuen Anhangs F „Techniken für die Nicht-Beeinflussung von Software-Elementen auf einem Rechner“, um die Unabhängigkeit von Software-Elementen nachzuweisen.

Frühere Ausgaben

DIN EN 61508-3 (VDE 0803-3): 2002-12

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.

Eine Information über den Zusammenhang der zitierten Normen mit den entsprechenden Deutschen Normen ist in Tabelle NA.1 wiedergegeben.

Tabelle NA.1

Europäische Norm	Internationale Norm	Deutsche Norm	Klassifikation im VDE-Vorschriftenwerk
EN 60601 (alle Teile)	IEC 60601 (alle Teile)	DIN EN 60601 (VDE 0750-1) (alle Teile)	VDE 0750-1
EN 61131-3	IEC 61131-3	DIN EN 61131-3	–
EN 61508-1:2010	IEC 61508-1:2010	DIN EN 61508-1 (VDE 0803-1):2011-02	VDE 0803-1
EN 61508-2:2010	IEC 61508-2:2010	DIN EN 61508-2 (VDE 0803-2):2011-02	VDE 0803-2
EN 61508-4:2010	IEC 61508-4:2010	DIN EN 61508-4 (VDE 0803-4):2011-02	VDE 0803-4
EN 61508-5:2010	IEC 61508-5:2010	DIN EN 61508-5 (VDE 0803-5):2011-02	VDE 0803-5
EN 61508-6:2010	IEC 61508-6:2010	DIN EN 61508-6 (VDE 0803-6):2011-02	VDE 0803-6
EN 61508-7:2010	IEC 61508-7:2010	DIN EN 61508-7 (VDE 0803-7):2011-02	VDE 0803-7
EN 61511 (alle Teile)	IEC 61511 (alle Teile)	DIN EN 61511 (VDE 0810-1) (alle Teile)	VDE 0810-1
EN 61800-5-2	IEC 61800-5-2	DIN EN 61800-5-2 (VDE 0160-105-2)	VDE 0160-105-2
EN 62061	IEC 62061	DIN EN 62061 (VDE 0113-50)	VDE 0113-50
–	IEC Guide 104:1997	–	–
–	ISO/IEC Guide 51 :1999	–	–

Nationaler Anhang NB (informativ)

Literaturhinweise

DIN EN 60601 (VDE 0750-1) (alle Teile), *Klassifikation und Kennzeichnung von Dokumenten für Anlagen, Systeme und Ausrüstungen*

DIN EN 61131-3, *Speicherprogrammierbare Steuerungen – Teil 3: Programmiersprachen*

DIN EN 61508-1 (VDE 0803-1):2011-02, *Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme – Allgemeine Anforderungen (IEC 61508-1:2010); Deutsche Fassung EN 61508-1:2010*

DIN EN 61508-2 (VDE 0803-2):2011-02, *Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme – Teil 2: Anforderungen an sicherheitsbezogene elektrische/elektronische/programmierbare elektronische Systeme (IEC 61508-2:2010); Deutsche Fassung EN 61508-2:2010*

DIN EN 61508-4 (VDE 0803-4):2011-02, *Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme – Teil 4: Begriffe und Abkürzungen (IEC 61508-4:2010); Deutsche Fassung EN 61508-4:2010*

DIN EN 61508-5 (VDE 0803-5):2011-02, *Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme – Teil 5: Beispiele von Methoden für die Bestimmung von Sicherheits-Integritätsleveln (IEC 61508-5:2010); Deutsche Fassung EN 61508-5:2010*

DIN EN 61508-6 (VDE 0803-6):2011-02, *Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme – Teil 6: Anwendungsrichtlinie für IEC 61508-2 und IEC 61508-3 (IEC 61508-6:2010); Deutsche Fassung EN 61508-6:2010*

DIN EN 61508-7 (VDE 0803-7):2011-02, *Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme – Teil 7: Überblick über Verfahren und Maßnahmen (IEC 61508-7:2010); Deutsche Fassung EN 61508-7:2010*

DIN EN 61511 (VDE 0810-1) (alle Teile), *Funktionale Sicherheit – Sicherheitstechnische Systeme für die Prozessindustrie*

DIN EN 61800-5-2 (VDE 0160-105-2), *Elektrische Leistungsantriebssysteme mit einstellbarer Drehzahl – Teil 5-2: Anforderungen an die Sicherheit – Funktionale Sicherheit*

DIN EN 62061 (VDE 0113-50), *Sicherheit von Maschinen – Funktionale Sicherheit sicherheitsbezogener elektrischer, elektronischer und programmierbarer elektronischer Steuerungssysteme*

Funktionale Sicherheit sicherheitsbezogener
elektrischer/elektronischer/programmierbarer elektronischer Systeme –
Teil 3: Anforderungen an Software
(IEC 61508-3:2010)

Functional safety of
electrical/electronic/programmable electronic
safety-related systems –
Part 3: Software requirements
(IEC 61508-3:2010)

Sécurité fonctionnelle des systèmes
électriques/électroniques/électroniques
programmables relatifs à la sécurité –
Partie 3: Exigences concernant les logiciels
(CEI 61508-3:2010)

Diese Europäische Norm wurde von CENELEC am 2010-05-01 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, Bulgarien, Dänemark, Deutschland, Estland, Finnland, Frankreich, Griechenland, Irland, Island, Italien, Kroatien, Lettland, Litauen, Luxemburg, Malta, den Niederlanden, Norwegen, Österreich, Polen, Portugal, Rumänien, Schweden, der Schweiz, der Slowakei, Slowenien, Spanien, der Tschechischen Republik, Ungarn, dem Vereinigten Königreich und Zypern.

CENELEC

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

Zentralsekretariat: Avenue Marnix 17, B-1000 Brüssel

© 2010 CENELEC – Alle Rechte der Verwertung, gleich in welcher Form und in welchem Verfahren, sind weltweit den Mitgliedern von CENELEC vorbehalten.

Ref. Nr. EN 61508-3:2010 D

Vorwort

Der Text des Schriftstücks 65A/550/FDIS, zukünftige 2. Ausgabe von IEC 61508-3, ausgearbeitet von dem SC 65A „System aspects“ des IEC TC 65 „Industrial-process measurement, control and automation“, wurde der IEC-CENELEC Parallelen Abstimmung unterworfen und von CENELEC am 2010-05-01 als EN 61508-3 angenommen.

Diese Europäische Norm ersetzt EN 61508-3:2001.

Sie ist eine Sicherheits-Grundnorm nach IEC-Guide 104.

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

Nachstehende Daten wurden festgelegt:

- spätestes Datum, zu dem die EN auf nationaler Ebene durch Veröffentlichung einer identischen nationalen Norm oder durch Anerkennung übernommen werden muss (dop): 2011-02-01
- spätestes Datum, zu dem nationale Normen, die der EN entgegenstehen, zurückgezogen werden müssen (dow): 2013-05-01

Der Anhang ZA wurde von CENELEC hinzugefügt.

Anerkennungsnotiz

Der Text der Internationalen Norm IEC 61508-3:2010 wurde von CENELEC ohne irgendeine Abänderung als Europäische Norm angenommen.

In der offiziellen Fassung sind unter „Literaturhinweise“ zu den aufgelisteten Normen die nachstehenden Anmerkungen einzutragen:

[1] IEC 61511 Reihe	ANMERKUNG	Harmonisiert in der Reihe EN 61511.
[2] IEC 62061	ANMERKUNG	Harmonisiert als EN 62061.
[3] IEC 61800-5-2	ANMERKUNG	Harmonisiert als EN 61800-5-2.
[4] IEC 61508-5:2010	ANMERKUNG	Harmonisiert als EN 61508-5:2010 (nicht modifiziert).
[5] IEC 61508-6:2010	ANMERKUNG	Harmonisiert als EN 61508-6:2010 (nicht modifiziert).
[6] IEC 61508-7:2010	ANMERKUNG	Harmonisiert als EN 61508-7:2010 (nicht modifiziert).
[7] IEC 60601 Reihe	ANMERKUNG	Harmonisiert in der Reihe EN 60601 (teilweise modifiziert).
[8] IEC 61131-3	ANMERKUNG	Harmonisiert als EN 61131-3.

Inhalt

	Seite
Vorwort.....	2
Einleitung	7
1 Anwendungsbereich	9
2 Normative Verweisungen	13
3 Begriffe und Abkürzungen	13
4 Übereinstimmung mit dieser Norm	13
5 Dokumentation	13
6 Zusätzliche Anforderungen an das Management der sicherheitsbezogenen Software.....	13
6.1 Ziele	13
6.2 Anforderungen	13
7 Anforderungen des Software-Sicherheitslebenszyklus	14
7.1 Allgemeines	14
7.2 Spezifikation der Anforderungen an die Sicherheit der Software.....	22
7.3 Validierungsplan für die Softwareaspekte der Systemsicherheit	26
7.4 Softwareentwurf und Entwicklung	27
7.5 Integration der programmierbaren Elektronik (Hardware und Software)	39
7.6 Software-Betriebs- und Modifikationsverfahren	40
7.7 Softwareaspekte bezüglich der Validierung der Sicherheit des Systems	40
7.8 Softwaremodifikation	42
7.9 Softwareverifikation	44
8 Beurteilung der funktionalen Sicherheit.....	48
Anhang A (normativ) Leitfaden für die Auswahl der Verfahren und Maßnahmen.....	50
Anhang B (informativ) Detailtabellen	59
Anhang C (informativ) Eigenschaften für die systematische Eignung der Software	64
C.1 Einleitung	64
C.1.1 Struktur des Anhangs C hinsichtlich der Anhänge A und B	64
C.1.2 Methode der Anwendung – 1	66
C.1.3 Methode der Anwendung – 2	67
C.2 Eigenschaften der systematischen Sicherheitsintegrität.....	69
C.3 Eigenschaften für die systematische Sicherheitsintegrität – Ausführliche Tabellen	101
Anhang D (normativ) Sicherheitshandbuch für konforme Objekte – zusätzliche Anforderungen an Softwareelemente.....	114
D.1 Zweck des Sicherheitshandbuchs	114
D.2 Inhalt des Sicherheitshandbuchs für ein Softwareelement	114
D.3 Begründung der Ansprüche im Sicherheitshandbuch für konforme Elemente	115
Anhang E (informativ) Beziehungen zwischen IEC 61508-2 und IEC 61508-3	117
Anhang F (informativ) Verfahren zum Erreichen der Nicht-Beeinflussung zwischen Softwareelementen auf einem einzelnen Rechner.....	119

	Seite
F.1 Einleitung	119
F.2 Bereiche des Verhaltens.....	119
F.3 Analyse ursächlicher Faktoren	119
F.4 Erreichung räumlicher Unabhängigkeit.....	120
F.5 Erreichung zeitlicher Unabhängigkeit	120
F.6 Anforderungen an die Hilfssoftware.....	121
F.7 Unabhängigkeit von Softwaremodulen – Aspekte zu Programmiersprachen	121
Anhang G (informativ) Leitlinien zur Anpassung des Lebenszyklus im Zusammenhang mit datengesteuerten Systemen.....	124
G.1 Datengesteuert – Systemteil und Anwendungsteil	124
G.2 Eingeschränkter Freiheitsgrad der Konfiguration, eingeschränkte Anwendungskonfigurierbarkeit	125
G.3 Eingeschränkter Freiheitsgrad der Konfiguration, vollständige Anwendungskonfigurierbarkeit	126
G.4 Eingeschränkter Freiheitsgrad der Programmierung, eingeschränkte Anwendungskonfigurierbarkeit	126
G.5 Eingeschränkter Freiheitsgrad der Programmierung, vollständige Anwendungskonfigurierbarkeit	126
G.6 Vollständige Funktionalität der Programmierung/Konfiguration, eingeschränkte Anwendungskonfigurierbarkeit	127
G.7 Vollständige Funktionalität der Programmierung/Konfiguration, vollständige Anwendungskonfigurierbarkeit	127
Literaturhinweise	128
Anhang ZA (normativ) Normative Verweisungen auf internationale Publikationen mit ihren entsprechenden europäischen Publikationen.....	129
Bilder	
Bild 1 – Gesamtrahmen der Normenreihe IEC 61508	11
Bild 2 – Gesamtsicherheitslebenszyklus.....	12
Bild 3 – Sicherheitslebenszyklus des E/E/PE-Systems (in der Realisierungsphase).....	16
Bild 4 – Software-Sicherheitslebenszyklus (in der Realisierungsphase)	16
Bild 5 – Beziehung zwischen IEC 61508-2 und IEC 61508-3 und ihre Anwendungsbereiche.....	17
Bild 6 – Systematische Eignung der Software und Entwicklungslebenszyklus (V-Modell)	17
Bild G.1 – Freiheitsgrad gegen Komplexität bei datengesteuerten Systemen	125
Tabellen	
Tabelle 1 – Software-Sicherheitslebenszyklus – Überblick	18
Tabelle A.1 – Spezifikation der Anforderungen an die Sicherheit der Software.....	51
Tabelle A.2 – Softwareentwurf und Softwareentwicklung – Entwurf der Softwarearchitektur	51
Tabelle A.3 – Softwareentwurf und Softwareentwicklung – Werkzeuge und Programmiersprachen	53
Tabelle A.4 – Softwareentwurf und Softwareentwicklung – Detaillierter Entwurf	54
Tabelle A.5 – Softwareentwurf und Softwareentwicklung – Test der Softwaremodule und Integration	54
Tabelle A.6 – Integration der programmierbaren Elektronik (Hardware und Software).....	55
Tabelle A.7 – Softwareaspekte zur Validierung der Sicherheit des Systems	56

	Seite
Tabelle A.8 – Modifikation	56
Tabelle A.9 – Softwareverifikation	57
Tabelle A.10 – Beurteilung der funktionalen Sicherheit.....	58
Tabelle B.1 – Entwurfs- und Programmierrichtlinien	59
Tabelle B.2 – Dynamische Analyse und Test.....	59
Tabelle B.3 – Funktionstest und Black-Box-Test.....	60
Tabelle B.4 – Ausfall-/Versagensanalyse	61
Tabelle B.5 – Modellierung	61
Tabelle B.6 – Leistungstest	62
Tabelle B.7 – Semiformale Methoden	62
Tabelle B.8 – Statische Analyse	63
Tabelle B.9 – Modularer Ansatz	63
Tabelle C.1 – Eigenschaften für die systematische Sicherheitsintegrität – Spezifikation der Anforderungen an die Sicherheit der Software	69
Tabelle C.2 – Eigenschaften für die systematische Sicherheitsintegrität – Softwareentwurf und - entwicklung – Entwurf der Softwarearchitektur	73
Tabelle C.3 – Eigenschaften für die systematische Sicherheitsintegrität – Softwareentwurf und - entwicklung – Unterstützende Werkzeuge und Programmiersprache	87
Tabelle C.4 – Eigenschaften für die systematische Sicherheitsintegrität – Softwareentwurf und - entwicklung – Detaillierter Entwurf (einschließlich Software-Systementwurf, Entwurf der Softwaremodule und Kodierung)	88
Tabelle C.5 – Eigenschaften für die systematische Sicherheitsintegrität – Softwareentwurf und - entwicklung – Test der Softwaremodule und Integration	91
Tabelle C.6 – Eigenschaften für die systematische Sicherheitsintegrität – Integration der programmierbaren Elektronik (Hardware und Software).....	93
Tabelle C.7 – Eigenschaften für die systematische Sicherheitsintegrität – Softwareaspekte bezüglich der Validierung der Sicherheit des Systems	94
Tabelle C.8 – Eigenschaften für die systematische Sicherheitsintegrität – Softwaremodifikation	95
Tabelle C.9 – Eigenschaften für die systematische Sicherheitsintegrität – Softwareverifikation	97
Tabelle C.10 – Eigenschaften für die systematische Sicherheitsintegrität – Beurteilung der funktionalen Sicherheit	99
Tabelle C.11 – Ausführliche Eigenschaften – Entwurfs- und Programmierrichtlinien	101
Tabelle C.12 – Ausführliche Eigenschaften – Dynamische Analyse und Test.....	104
Tabelle C.13 – Ausführliche Eigenschaften – Funktionstest und Black-Box-Test.....	106
Tabelle C.14 – Ausführliche Eigenschaften – Ausfall-/Versagensanalyse.....	107
Tabelle C.15 – Ausführliche Eigenschaften – Modellierung	108
Tabelle C.16 – Ausführliche Eigenschaften – Leistungstest	108
Tabelle C.17 – Ausführliche Eigenschaften – Semiformale Methoden	109
Tabelle C.18 – Eigenschaften der systematischen Sicherheitsintegrität – Statische Analyse.....	111
Tabelle C.19 – Ausführliche Eigenschaften – Modularer Ansatz	112
Tabelle E.1 – Kategorien der IEC 61508-2-Anforderungen.....	117

	Seite
Tabelle E.2 – Anforderungen der 61508-2 an Software und ihre typische Bedeutung für bestimmte Typen von Software	117
Tabelle F.1 – Modulkopplung – Definition von Begriffen	121
Tabelle F.2 – Arten von Modulkopplungen	122

Einleitung

Systeme, die aus elektrischen und/oder elektronischen Elementen bestehen, werden seit vielen Jahren verwendet, um Sicherheitsfunktionen in den meisten Anwendungsbereichen auszuführen. Auf Rechnern basierende Systeme (allgemein ausgedrückt programmierbare elektronische Systeme) werden in allen Anwendungsbereichen benutzt, um Nichtsicherheitsfunktionen und zunehmend auch um Sicherheitsfunktionen auszuführen. Falls Rechnersystemtechnologie wirksam und sicherheitsgerichtet eingesetzt wird, ist es wichtig, dass die für die Entscheidungsfindung Verantwortlichen ausreichende Hilfestellung bezüglich der Sicherheitsaspekte erhalten, nach denen diese Entscheidungen getroffen werden.

Diese Internationale Norm beschreibt einen allgemeinen Lösungsweg für alle Tätigkeiten während des Sicherheitslebenszyklus für Systeme, die aus elektrischen und/oder elektronischen und/oder programmierbaren elektronischen (E/E/PE) Elementen bestehen und die eingesetzt werden, um Sicherheitsfunktionen auszuführen. Dieser allgemeine Lösungsweg wurde gewählt, um ein sinnvolles und konsistentes technisches Verfahren für alle elektrischen Sicherheitssysteme zu entwickeln. Ein Hauptziel ist es, die Entwicklung von produkt- und anwendungsspezifischen internationalen Normen, die auf der Normenreihe IEC 61508 basieren, zu erleichtern.

ANMERKUNG 1 In den Literaturhinweisen sind Beispiele von produkt- und anwendungsspezifischen internationalen Normen, die auf der Normenreihe IEC 61508 basieren, enthalten (siehe Hinweise [1], [2] und [3]).

In den meisten Situationen wird Sicherheit durch eine Anzahl von Systemen erreicht, die auf vielerlei Technologien (zum Beispiel Mechanik, Hydraulik, Pneumatik, Elektrik, Elektronik, programmierbare Elektronik) basieren. Jede Sicherheitsstrategie muss deshalb nicht nur alle Elemente innerhalb eines Einzelsystems (zum Beispiel Sensoren, Steuereinheiten und Aktoren) betrachten, sondern auch all die sicherheitsbezogenen Systeme, welche die Gesamtheit von sicherheitsbezogenen Systemen bilden. Daher kann diese Internationale Norm, obwohl sie sich mit sicherheitsbezogenen E/E/PE-Systemen beschäftigt, auch einen Rahmen bereitstellen, innerhalb dessen sicherheitsbezogene Systeme basierend auf anderen Technologien betrachtet werden können.

Es ist beachtet worden, dass eine große Vielfalt von Anwendungen in vielfältigen Anwendungsbereichen vorliegt, die sicherheitsbezogene E/E/PE-Systeme verwenden, und diese einen weiten Bereich in Bezug auf Komplexität, Gefährdungs- und Risikopotentiale abdecken. In jeder speziellen Anwendung sind die erforderlichen Sicherheitsmaßnahmen von vielen anwendungsspezifischen Faktoren abhängig. Dadurch, dass diese Internationale Norm allgemein gehalten ist, wird die Formulierung solcher Maßnahmen in zukünftigen produkt- und anwendungsspezifischen internationalen Normen und in Überarbeitungen der bereits bestehenden Normen ermöglicht.

Diese Internationale Norm:

- betrachtet alle relevanten Phasen des Gesamt-Sicherheitslebenszyklus, des Sicherheitslebenszyklus des E/E/PE-Systems und des Software-Sicherheitslebenszyklus (zum Beispiel vom anfänglichen Konzept über Entwurf, Implementierung, Betrieb und Instandhaltung bis zur Außerbetriebnahme), wenn E/E/PE-Systeme benutzt werden, um Sicherheitsfunktionen auszuführen;
- wurde unter Berücksichtigung einer sich schnell entwickelnden Technologie entworfen. Der Betrachtungsrahmen ist ausreichend robust und ausführlich genug, um auch für zukünftige Entwicklungen verwendbar zu sein;
- ermöglicht die Erstellung produkt- und anwendungsspezifischer Internationaler Normen, die sich mit sicherheitsbezogenen E/E/PE-Systemen befassen. Die Entwicklung produkt- und anwendungsspezifischer Internationaler Normen sollte innerhalb des Rahmens dieser Norm zu einem hohen Grad an Übereinstimmung (zum Beispiel von zugrunde liegenden Prinzipien, Terminologie usw.) führen, sowohl innerhalb der Anwendungsbereiche als auch über die Anwendungsbereiche hinweg. Dies hat sowohl sicherheitstechnische als auch wirtschaftliche Vorteile;
- liefert eine Methode für die Entwicklung der Spezifikation der Anforderungen an die Sicherheit, die notwendig ist, um die erforderliche funktionale Sicherheit für sicherheitsbezogene E/E/PE-Systeme zu erreichen;
- verwendet einen auf dem Risiko basierenden Lösungsansatz, durch den die Anforderungen an die Sicherheitsintegrität bestimmt werden können;

- führt Sicherheits-Integritätslevel für die Festlegung der Zielvorgabe der Sicherheitsintegrität der Sicherheitsfunktionen ein, die von den sicherheitsbezogenen E/E/PE-Systemen zu implementieren sind;

ANMERKUNG 2 Diese Norm legt weder die Anforderungen an den Sicherheits-Integritätslevel für irgendeine Sicherheitsfunktion fest, noch bestimmt sie, wie der Sicherheits-Integritätslevel festgelegt wird. Stattdessen stellt sie einen risikobasierenden konzeptionellen Rahmen und Beispielverfahren bereit.

- legt Ausfallgrenzwerte für die von den sicherheitsbezogenen E/E/PE-Systemen auszuführenden Sicherheitsfunktionen fest, die mit den Sicherheits-Integritätsleveln verbunden sind;
- legt eine untere Grenze für die Ausfallgrenzwerte für eine Sicherheitsfunktion fest, die von einem einzelnen sicherheitsbezogenen E/E/PE-System ausgeführt wird. Für sicherheitsbezogene E/E/PE-Systeme, die:
 - in der Betriebsart mit einer niedrigen Anforderungsrate betrieben werden, ist die untere Grenze bei einer mittleren Wahrscheinlichkeit eines gefahrbringenden Ausfalls bei Anforderung von 10^{-5} festgelegt;
 - in der Betriebsart mit einer hohen Anforderungsrate oder mit kontinuierlicher Anforderung betrieben werden, ist die untere Grenze bei einer mittleren Häufigkeit eines gefahrbringenden Ausfalls von $10^{-9} [h^{-1}]$ festgelegt;

ANMERKUNG 3 Ein einzelnes sicherheitsbezogenes E/E/PE-System bedeutet nicht notwendigerweise eine einkanale Architektur.

ANMERKUNG 4 Es kann für einfache Systeme möglich sein, Entwürfe von sicherheitsbezogenen Systemen mit niedrigeren Zielwerten für die Sicherheitsintegrität zu erreichen, aber diese Grenzen werden als das betrachtet, was für relativ komplexe Systeme (zum Beispiel sicherheitsbezogene programmierbare elektronische Systeme) gegenwärtig erreicht werden kann.

- legt Anforderungen für die Vermeidung und Beherrschung von systematischen Fehlern fest, die auf Erfahrungen und Urteilsvermögen beruhen, die durch praktische Erfahrung in der Industrie gewonnen wurden. Wenn auch die Wahrscheinlichkeit des Auftretens systematischer Ausfälle im Allgemeinen nicht quantifiziert werden kann, erlaubt die Norm jedoch für eine festgelegte Sicherheitsfunktion den Anspruch zu erheben, dass der mit der Sicherheitsfunktion verbundene Ausfallgrenzwert als erreicht betrachtet werden kann, wenn alle Anforderungen dieser Norm erfüllt worden sind;
- führt die systematische Eignung ein, die für ein Element im Hinblick auf das Vertrauen gilt, dass die systematische Sicherheitsintegrität die Anforderungen des festgelegten Sicherheits-Integritätslevels erfüllt;
- lässt einen weiten Bereich von Prinzipien, Verfahren und Maßnahmen zu, um funktionale Sicherheit für sicherheitsbezogene E/E/PE-Systeme zu erreichen, verwendet aber nicht ausdrücklich das Fail-Safe-Konzept. „Fail-Safe“-Konzepte und „inhärent sichere“ Prinzipien können jedoch anwendbar sein, und der Einsatz solcher Konzepte ist zulässig, vorausgesetzt, dass die Anforderungen der zutreffenden Abschnitte in der Norm erfüllt werden.

Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme – Teil 3: Anforderungen an Software

1 Anwendungsbereich

1.1 Dieser Teil der Normenreihe IEC 61508

- a) ist dazu vorgesehen, nur nach einem gründlichen Verständnis der IEC 61508-1 und IEC 61508-2 angewendet zu werden;
- b) trifft auf jede Software zu, die einen Teil eines sicherheitsbezogenen Systems bildet oder verwendet wird, um ein sicherheitsbezogenes System im Anwendungsbereich der IEC 61508-1 und IEC 61508-2 zu entwickeln. Diese Software wird dann sicherheitsbezogene Software genannt (einschließlich Betriebssysteme, Systemsoftware, Software in Kommunikationsnetzwerken, Funktionen der Mensch-Maschinen-Schnittstelle, und Firmware sowie Anwendungssoftware);
- c) stellt bestimmte Anforderungen an Werkzeuge, die verwendet werden, um ein sicherheitsbezogenes System innerhalb des Anwendungsbereichs der IEC 61508-1 und IEC 61508-2 zu entwickeln und zu konfigurieren;
- d) fordert, dass die Sicherheitsfunktionen der Software und die systematische Eignung spezifiziert werden;

ANMERKUNG 1 Sollte dies bereits als Teil der Spezifikation des sicherheitsbezogenen E/E/PE-Systems (siehe 7.2 der IEC 61508 -2) gemacht worden sein, so muss es an dieser Stelle nicht noch einmal wiederholt werden.

ANMERKUNG 2 Das Spezifizieren der Sicherheitsfunktionen der Software und der systematischen Eignung ist ein iterativer Prozess; siehe [Bilder 3](#) und [6](#).

ANMERKUNG 3 Zur Struktur der Dokumentation siehe Abschnitt 5 und Anhang A der IEC 61508-1. Die Struktur der Dokumentation darf die Prozesse des Unternehmens berücksichtigen und ebenso die Arbeitserfahrung für das spezielle Anwendungsgebiet.

ANMERKUNG 4 Für die Definition des Ausdrucks „systematische Eignung“ siehe 3.5.9 der IEC 61508-4.

- e) legt Anforderungen und Tätigkeiten fest, die in den Phasen des Lebenszyklus während des Entwurfs und der Entwicklung der sicherheitsbezogenen Software (das Modell des Software-Sicherheitslebenszyklus) angewendet werden müssen. Diese Anforderungen beinhalten den Einsatz von Maßnahmen und Verfahren, die zur Vermeidung und Beherrschung von Fehlern und Versagensfällen der Software der erforderlichen systematischen Eignung entsprechend abgestuft sind;
- f) liefert Anforderungen an die Informationen zu Softwareaspekten der Validierung der Sicherheit des Systems, die an die Organisation, die die E/E/PE-Systemintegration durchführt, weitergeleitet werden;
- g) liefert Anforderungen zur Vorbereitung von Informationen und Prozeduren, die die Software betreffen und die vom Anwender für den Betrieb und die Instandhaltung des sicherheitsbezogenen Systems benötigt werden;
- h) liefert Anforderungen, die von der Organisation, die Modifikationen an der sicherheitsbezogenen Software durchführt, zu erfüllen sind;
- i) liefert im Zusammenwirken mit IEC 61508-1 und IEC 61508-2 Anforderungen an Werkzeuge, wie Werkzeuge zur Entwicklung und zum Entwurf, Übersetzer, Werkzeuge für Test und Debugging und Werkzeuge für das Konfigurationsmanagement;

ANMERKUNG 4 Das [Bild 5](#) zeigt die Beziehung zwischen der IEC 61508-2 und IEC 61508-3.

- j) Gilt nicht für Medizingeräte in Übereinstimmung mit der Normenreihe IEC 60601.

1.2 IEC 61508-1, IEC 61508-2, IEC 61508-3 und IEC 61508-4 sind Sicherheits-Grundnormen, dieser Status ist aber im Zusammenhang mit einfachen sicherheitsbezogenen E/E/PE-Systemen nicht anwendbar (siehe IEC 61508-4, 3.4.3). Als Sicherheitsgrundnormen sind sie zur Verwendung durch technische Komitees bei der Erstellung von Normen entsprechend den in IEC Guide 104 und ISO/IEC Guide 51 enthaltenen Grundsätzen vorgesehen. IEC 61508-1, IEC 61508-2, IEC 61508-3 und IEC 61508-4 sind ebenfalls zur Verwendung als eigenständige Veröffentlichungen vorgesehen. Die horizontale Sicherheitsfunktion dieser Internationalen Norm gilt nicht für Medizingeräte in Übereinstimmung mit der Normenreihe IEC 60601.

1.3 Es steht in der Verantwortung eines Technischen Komitees, zur Vorbereitung und Erstellung eigener Veröffentlichungen soweit möglich die Sicherheits-Grundnormen anzuwenden. In diesem Zusammenhang gilt, dass die Anforderungen, Prüfverfahren oder Prüfbedingungen dieser Sicherheitsgrundnorm nicht anzuwenden sind, außer in den Veröffentlichungen der Technischen Komitees wird speziell darauf verwiesen oder sie werden eingebunden.

1.4 [Bild 1](#) zeigt den Gesamtrahmen für die Normenreihe IEC 61508 und zeigt die Rolle, welche die IEC 61508-3 beim Erreichen der funktionalen Sicherheit für sicherheitsbezogene E/E/PE-Systeme spielt.

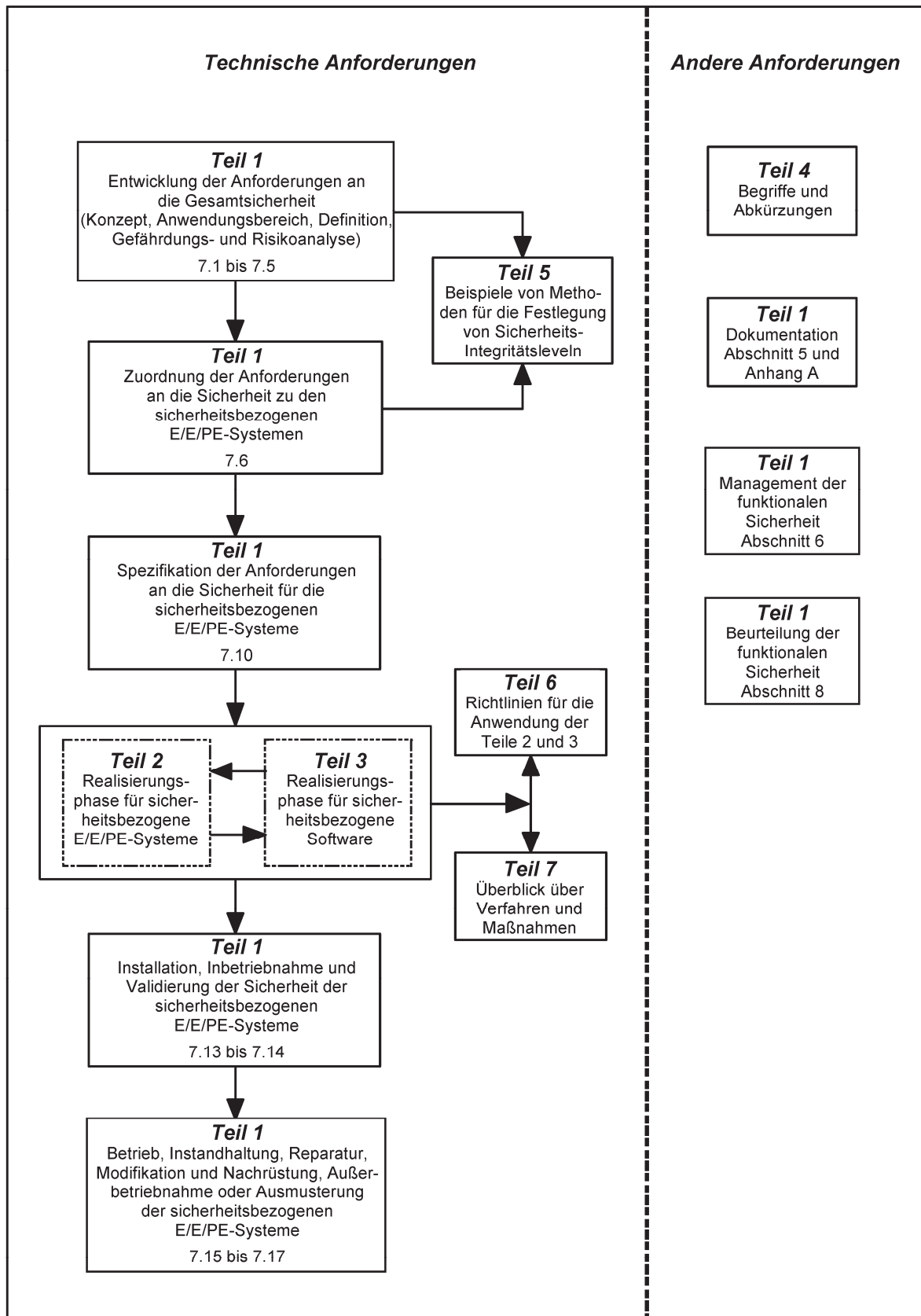


Bild 1 – Gesamtrahmen der Normenreihe IEC 61508

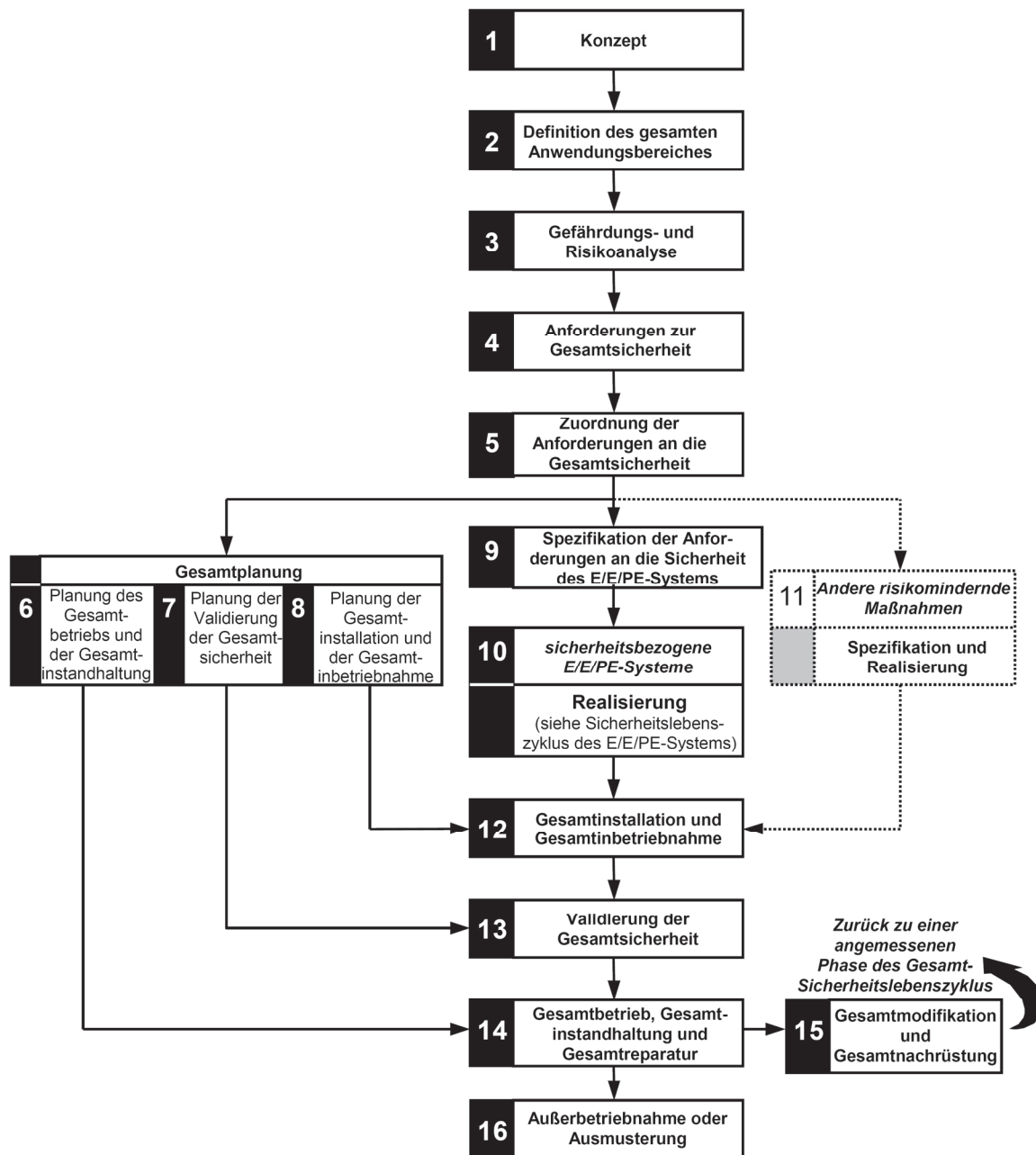


Bild 2 – Gesamtsicherheitslebenszyklus

2 Normative Verweisungen

Die folgenden zitierten Dokumente sind für die Anwendung dieses Dokuments erforderlich. Bei datierten Verweisungen gilt nur die in Bezug genommene Ausgabe. Bei undatierten Verweisungen gilt die letzte Ausgabe des in Bezug genommenen Dokuments (einschließlich aller Änderungen).

IEC 61508-1:2010, *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 1: General requirements*

IEC 61508-2:2010, *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 2: Requirements for electrical/electronic/programmable electronic safety-related systems*

IEC 61508-4:2010, *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 4: Definitions and abbreviations*

IEC Guide 104:1997, *The preparation of safety publications and the use of basic safety publications and group safety publications*

ISO/IEC Guide 51:1999, *Safety aspects – Guidelines for their inclusion in standards*

3 Begriffe und Abkürzungen

Für die Anwendung dieses Dokuments gelten die Begriffe und Abkürzungen nach IEC 61508-4.

4 Übereinstimmung mit dieser Norm

Die Anforderungen an die Übereinstimmung mit dieser Norm befinden sich in IEC 61508-1, Abschnitt 4.

5 Dokumentation

Die Ziele und Anforderungen zur Dokumentation befinden sich in IEC 61508-1, Abschnitt 5.

6 Zusätzliche Anforderungen an das Management der sicherheitsbezogenen Software

6.1 Ziele

Die Ziele sind in IEC 61508-1, 6.1, ausführlich beschrieben.

6.2 Anforderungen

6.2.1 Die Anforderungen sind in IEC 61508-1, 6.2, ausführlich beschrieben, mit folgenden zusätzlichen Anforderungen.

6.2.2 Die Planung der funktionalen Sicherheit muss die Strategie definieren für die Beschaffung, Entwicklung, Integration, Verifikation, Validierung und Modifikation der Software im Rahmen der Anforderungen des Sicherheits-Integritätslevels der Sicherheitsfunktionen, die durch das sicherheitsbezogene E/E/PE-System implementiert werden.

ANMERKUNG Die Philosophie dieser Vorgehensweise ist die Verwendung der Planung der funktionalen Sicherheit als Möglichkeit, diese Norm unter Berücksichtigung der notwendigen Sicherheitsintegrität für jede durch das sicherheitsbezogene E/E/PE-System implementierte Sicherheitsfunktion anzupassen.

6.2.3 Das Software-Konfigurationsmanagement muss:

- a) Verwaltungs- und technische Kontrollen während des Software-Sicherheitslebenszyklus verwenden, um Softwaremodifikationen zu verwalten und somit sicherzustellen, dass die spezifizierten Anforderungen an die sicherheitsbezogene Software weiterhin erfüllt werden;
- b) garantieren, dass alle notwendigen Tätigkeiten durchgeführt wurden, um darzulegen, dass die erforderliche systematische Eignung der Software erreicht wurde;
- c) genau und mit eindeutiger Kennzeichnung alle Konfigurationsmerkmale behandeln, die notwendig sind, um die erforderliche Sicherheitsintegrität des sicherheitsbezogenen E/E/PE-Systems zu erreichen. Konfigurationsmerkmale beinhalten mindestens Folgendes: die Sicherheitsanalyse und Sicherheitsanforderungen, Spezifikation der Software und Entwurfsdokumente, Softwaremodule im Source-Code, Testpläne und Testergebnisse, Verifikationsdokumente, bereits existierende Softwareelemente und Softwarepakete, die in das sicherheitsbezogene E/E/PE-System integriert werden, alle Werkzeuge und Entwicklungsumgebungen, die verwendet werden, um die Software des sicherheitsbezogenen E/E/PE-Systems zu erstellen oder zu testen, oder jedwede Arbeit an ihr durchzuführen;
- d) Verfahren zur Änderungskontrolle anwenden:
 - um nicht autorisierte Modifikationen zu verhindern und Modifikationsanforderungen zu dokumentieren;
 - um die Einflüsse einer beabsichtigten Modifikation analysieren zu können und eine Anfrage genehmigen oder zurückweisen zu können;
 - um die Einzelheiten und die Bevollmächtigung aller genehmigten Modifikationen zu dokumentieren;
 - um die Mindestanforderungen an die Konfiguration zu bestimmten Punkten in der Softwareentwicklung festzulegen und die (teilweisen) Integrationstests der Mindestanforderung zu dokumentieren;
 - um die Zusammenstellung und den Aufbau aller Mindestanforderungen an die Software (mit den Modifikationen früherer Mindestanforderungen) garantieren zu können.

ANMERKUNG 1 Zur Anleitung für die Verwendung von Verwaltungs- und technischen Kontrollen und deren Durchsetzung ist die Entscheidung und Autorität des Managements notwendig.

ANMERKUNG 2 Als ein Extrem kann eine Einflussanalyse eine informelle Beurteilung beinhalten. Als anderes Extrem kann eine Einflussanalyse eine strenge formale Analyse der potentiell nachteiligen Einflüsse aller geplanten Modifikationen enthalten, die nur ungenügend verstanden oder angewendet wurden. Siehe IEC 61508-7 zur Anleitung für eine Einflussanalyse.

- e) sicherstellen, dass angemessene Methoden implementiert werden, um gültige Softwareelemente und Daten ordnungsgemäß in das Laufzeitsystem zu laden;

ANMERKUNG 3 Dies kann die Berücksichtigung spezieller lokaler Zielsysteme sowie allgemeiner Systeme einschließen. Nicht-Anwender-Software kann eine sichere Lademethode, z. B. für Firmware, benötigen.

- f) die folgenden Informationen dokumentieren, um ein nachfolgendes Audit der funktionalen Sicherheit zu ermöglichen: den Status der Konfiguration, Version, Begründung (unter Berücksichtigung der Einflussanalyse) und Genehmigung der Modifikationen und aller Einzelheiten der Änderung;
- g) die Freigabe der sicherheitsbezogenen Software formal dokumentieren. Kopien der Software und alle zugehörige Dokumentation und Versionen der verwendeten Daten müssen aufbewahrt werden, um Pflege und Modifikation während der Betriebslebensdauer der freigegebenen Softwareversion zu ermöglichen.

ANMERKUNG 4 Für weitere Information zum Konfigurationsmanagement siehe IEC 61508-7.

7 Anforderungen des Software-Sicherheitslebenszyklus

7.1 Allgemeines

7.1.1 Ziel

Das Ziel der Anforderungen dieses Unterabschnitts ist es, die Entwicklung der Software in definierte Phasen und Tätigkeiten zu strukturieren (siehe [Tabelle 1](#) und [Bilder 3 bis 6](#)).

7.1.2 Anforderungen

7.1.2.1 Für die Entwicklung der Software muss während der Sicherheitsplanung ein Sicherheitslebenszyklus in Übereinstimmung mit IEC 61508-1, Abschnitt 6, ausgewählt und spezifiziert werden.

7.1.2.2 Es darf jedes Modell eines Softwarelebenszyklus verwendet werden, vorausgesetzt, alle Ziele und Anforderungen dieses Abschnitts werden erfüllt.

7.1.2.3 Jede Phase des Software-Sicherheitslebenszyklus muss in elementare Tätigkeiten aufgeteilt werden, wobei Anwendungsbereich, Eingaben und Ausgaben für jede Phase festgelegt sind.

ANMERKUNG Siehe [Bilder 3, 4](#) und [Tabelle 1](#).

7.1.2.4 Falls der Software-Sicherheitslebenszyklus die Anforderungen von [Tabelle 1](#) erfüllt, ist es annehmbar, das V-Modell (siehe [Bild 6](#)) unter Berücksichtigung der Sicherheitsintegrität und Komplexität des Projekts anzupassen.

ANMERKUNG 1 Ein Modell eines Software-Sicherheitslebenszyklus, das die Anforderungen dieses Abschnitts erfüllt, kann in geeigneter Weise auf die Belange des Projektes oder der Organisation angepasst werden. Die gesamte Liste der Phasen des Lebenszyklus in [Tabelle 1](#) ist geeignet für große, neuentwickelte Systeme. Für kleine Systeme kann es zum Beispiel angemessen sein, die Phasen Entwurf des Softwaresystems und Entwurf der Architektur zusammenzufassen.

ANMERKUNG 2 Siehe [Anhang G](#) zu den Eigenschaften von datengesteuerten Systemen (z. B. Programmiersprachen mit vollem/eingeschränktem Sprachumfang und Umfang der Datenkonfiguration), die relevant sein können zur individuellen Anpassung des Software-Sicherheitslebenszyklus.

7.1.2.5 Jede individuelle Anpassung des Software-Sicherheitslebenszyklus muss auf Basis der funktionalen Sicherheit begründet werden.

7.1.2.6 Verfahren zur Gewährleistung der Qualität und Sicherheit müssen in die Tätigkeiten des Sicherheitslebenszyklus integriert werden.

7.1.2.7 Für jede Phase des Lebenszyklus müssen angemessene Verfahren und Maßnahmen verwendet werden. Die [Anhänge A](#) und [B](#) geben Hilfestellung für die Auswahl von Verfahren und Maßnahmen und Verweise auf die IEC 61508-6 und IEC 61508-7. Die IEC 61508-6 und IEC 61506-7 geben Empfehlungen für spezielle Verfahren, um die für die systematische Sicherheitsintegrität erforderlichen Eigenschaften zu erreichen. Die Auswahl von Verfahren aus diesen Empfehlungen gibt für sich allein keine Gewähr, dass die erforderliche Sicherheitsintegrität erreicht wird.

ANMERKUNG Der Erfolg zur Erreichung der systematischen Sicherheitsintegrität hängt ab von der Auswahl der Techniken mit Augenmerk auf folgende Faktoren:

- Beschaffenheit und die ergänzenden Eigenschaften der gewählten Methoden, Sprachen und Werkzeuge für den gesamten Entwicklungszyklus;
- ob die Entwickler Methoden, Sprachen und Werkzeuge verwenden, die sie vollständig verstehen;
- ob die Methoden, Sprachen und Werkzeuge an die speziellen Probleme, die während der Entwicklung vorgefunden werden, gut angepasst sind.

7.1.2.8 Die Ergebnisse der Aktivitäten im Software-Sicherheitslebenszyklus müssen dokumentiert werden (siehe [Abschnitt 5](#)).

ANMERKUNG IEC 61508-1, Abschnitt 5, berücksichtigt die dokumentierten Ausgaben der Phasen des Sicherheitslebenszyklus. Bei der Entwicklung sicherheitsbezogener E/E/PE-Systeme kann das Ergebnis einiger Phasen des Lebenszyklus ein eigenständiges Dokument sein, während die dokumentierten Ausgaben verschiedener Phasen in einem Dokument zusammengefasst werden können. Die wesentliche Anforderung ist, dass die Ausgaben einer Phase des Sicherheitslebenszyklus für den angestrebten Zweck ausreichend sind.

7.1.2.9 Wenn sich in irgendeiner Phase des Software-Sicherheitslebenszyklus eine Modifikation auf eine frühere Phase des Lebenszyklus bezieht, muss eine Einflussanalyse bestimmen (1), welche Softwaremodule betroffen sind und (2) welche früheren Aktivitäten des Lebenszyklus wiederholt werden müssen.

ANMERKUNG Als ein Extrem kann eine Einflussanalyse eine informelle Beurteilung beinhalten. Als anderes Extrem kann eine Einflussanalyse eine strenge formale Analyse der potentiell nachteiligen Einflüsse aller geplanten Modifikationen enthalten, die nur ungenügend verstanden oder angewendet wurden. Siehe IEC 61508-7 zur Anleitung für eine Einflussanalyse.

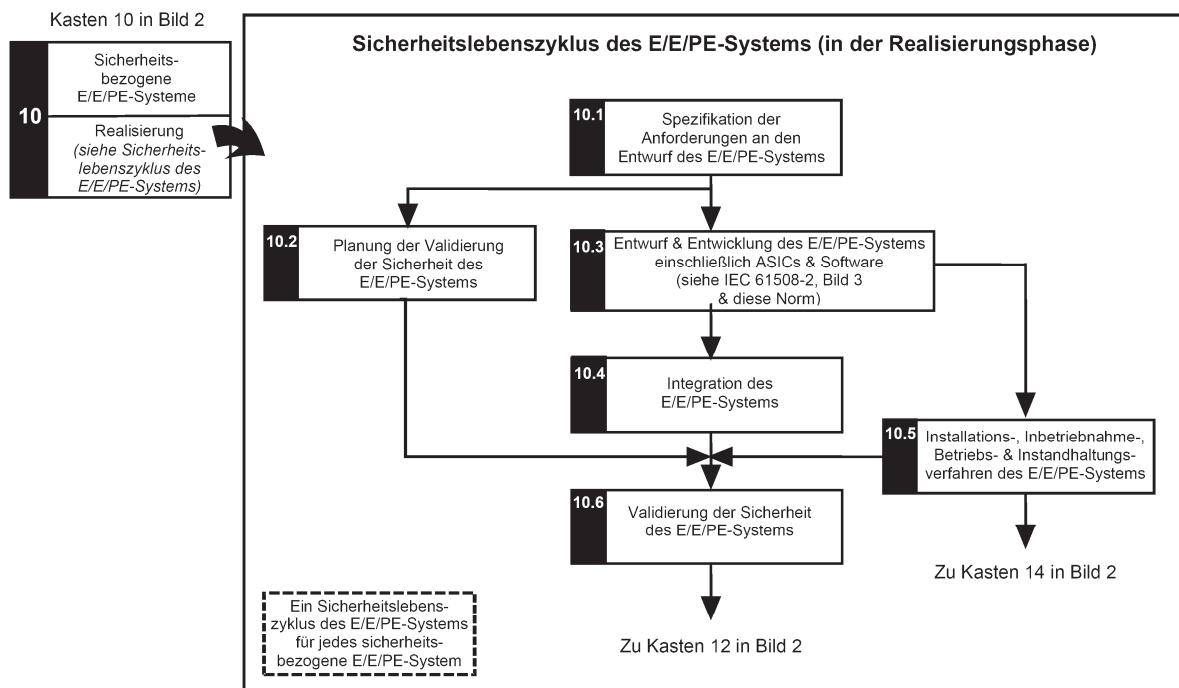


Bild 3 – Sicherheitslebenszyklus des E/E/PE-Systems (in der Realisierungsphase)

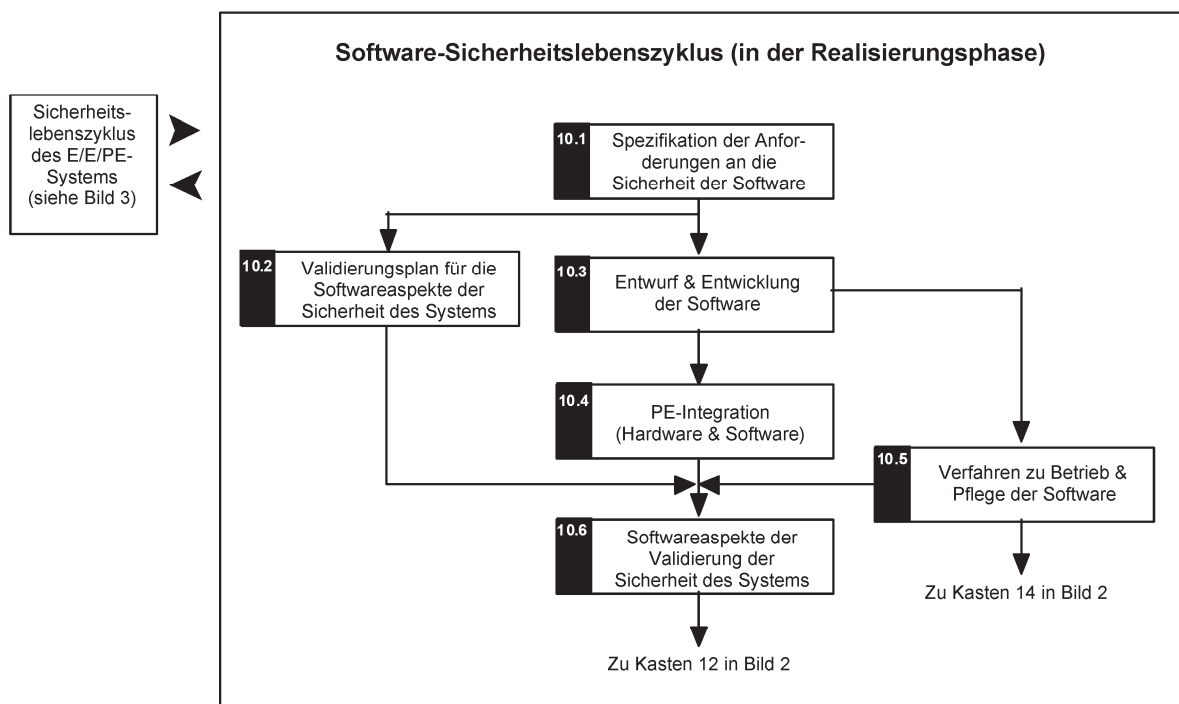


Bild 4 – Software-Sicherheitslebenszyklus (in der Realisierungsphase)

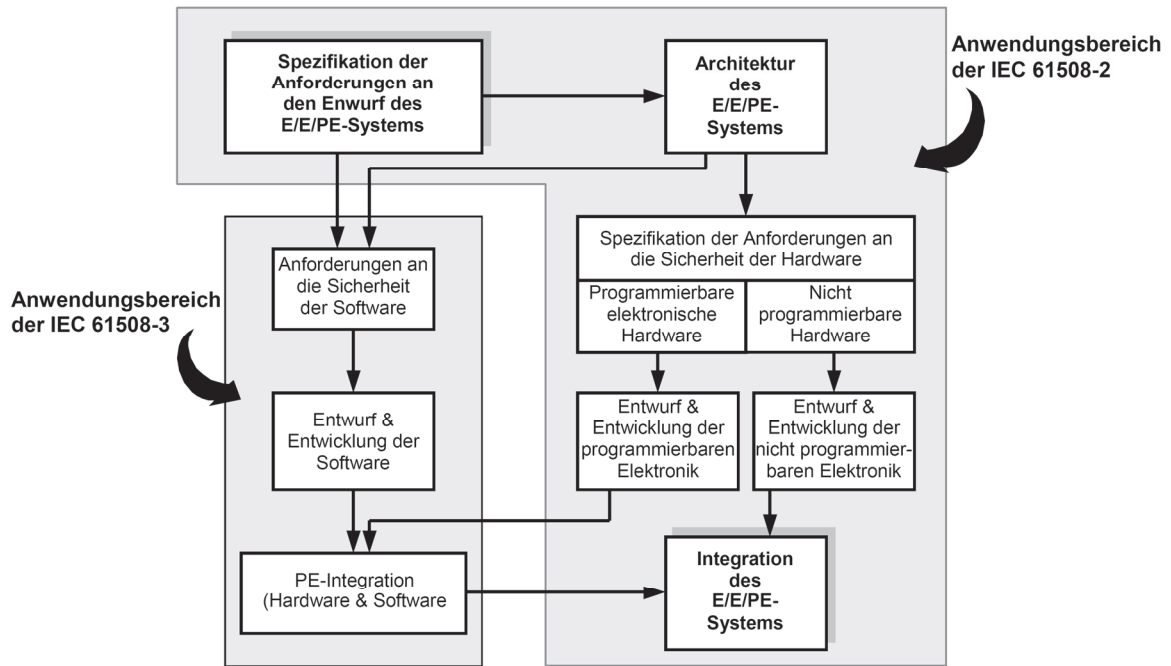


Bild 5 – Beziehung zwischen IEC 61508-2 und IEC 61508-3 und ihre Anwendungsbereiche

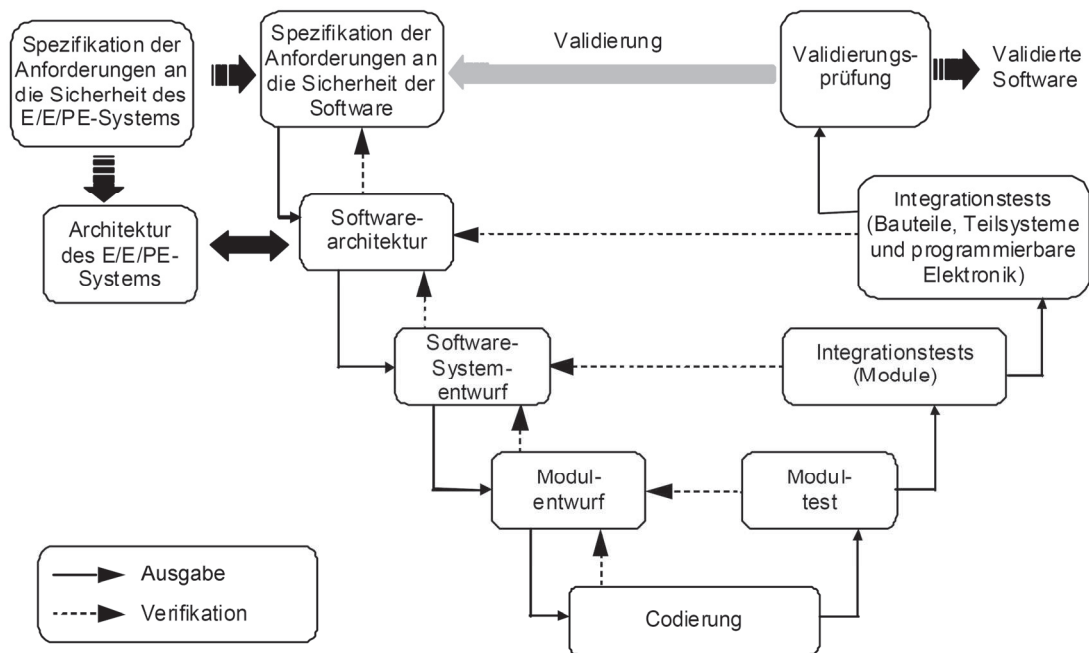


Bild 6 – Systematische Eignung der Software und Entwicklungslebenszyklus (V-Modell)

Tabelle 1 – Software-Sicherheitslebenszyklus – Überblick

Phase des Sicherheitslebenszyklus		Ziele	Anwendungsbereich	Unterabschnitt für Anforderungen	Eingaben (benötigte Information)	Ergebnisse (Ausgaben, erzeugte Informationen)
Kastennummer in Bild 4	Titel					
10.1	Spezifikation der Anforderungen an die Sicherheit der Software	Spezifizieren der Anforderungen an die sicherheitsbezogene Software mit Begriffen der Anforderungen an die Funktionen der Software bezüglich der Sicherheit und der Anforderungen an die systematische Eignung der Software Spezifizieren der Anforderungen an die Funktionen der Software bezüglich der Sicherheit für jedes sicherheitsbezogene E/E/PE-System, das notwendig ist, um die erforderlichen Sicherheitsfunktionen zu implementieren Spezifizieren der Anforderungen an die systematische Eignung der Software für jedes sicherheitsbezogene E/E/PE-System, das notwendig ist, um den spezifizierten Sicherheits-Integritätslevel jeder zugeordneten Sicherheitsfunktion des sicherheitsbezogenen E/E/PE-Systems zu erreichen	PE-System; Software-system	7.2.2	Spezifikation der E/E/PE-Sicherheitsanforderungen wie während der Zuordnung entwickelt (siehe IEC 61508-1) Spezifikation der Anforderungen an die Sicherheit des E/E/PE-Systems (von IEC 61508-2)	Spezifikation der Anforderungen an die Sicherheit der Software
10.2	Validierungsplan für die Softwareaspekte der Systemsicherheit	Entwickeln eines Plans zur Validierung der Aspekte der Systemsicherheit	PE-System; Software-system	7.3.2	Spezifikation der Anforderungen an die Sicherheit der Software	Validierungsplan für die Softwareaspekte der Systemsicherheit
10.3	Softwareentwurf und Entwicklung	Architektur: Schaffen einer Softwarearchitektur, die die spezifizierten Anforderungen für die sicherheitsbezogene Software bezüglich des erforderlichen Sicherheits-Integritätslevels erfüllt; Bewerten der Anforderungen an die Software, die durch die Architektur der Hardware des sicherheitsbezogenen E/E/PE-Systems gestellt werden, einschließlich der Bedeutung der Wechselwirkungen der E/E/PE Hardware mit der Software für die Sicherheit der EUC-Einrichtung	PE-System; Software-system	7.4.3	Spezifikation der Anforderungen an die Sicherheit der Software; E/E/PE-Systementwurf der Hardwarearchitektur (aus IEC 61508-2)	Softwarearchitektur-Entwurf; Spezifikation des Integrationstests der Softwarearchitektur; Spezifikation des Software-/PE-Integrationstests (auch gefordert durch IEC 61508-2)

Tabelle 1 (fortgesetzt)

Phase des Sicherheitslebenszyklus		Ziele	Anwendungsbereich	Unterabschnitt für Anforderungen	Eingaben (benötigte Information)	Ergebnisse (Ausgaben, erzeugte Informationen)
Kastennummer in Bild 4	Titel					
10.3	Softwareentwurf und Entwicklung	Werkzeuge und Programmiersprachen: Auswahl geeigneter Werkzeuge einschließlich Sprachen und Compilern, Schnittstellen zum Laufzeitsystem, Anwenderschnittstellen, Datenformate und Darstellungen für den erforderlichen Sicherheits-Integritätslevel über den gesamten Sicherheitslebenszyklus der Software, die die Verifikation, Validierung, Beurteilung und Modifikationen unterstützt	PE-System; Softwaresystem; Werkzeuge; Programmiersprache	7.4.4	Spezifikation der Anforderungen an die Sicherheit der Software; Software-Architektur-Entwurf	Entwicklungswerkzeuge und Programmierrichtlinien; Auswahl der Entwicklungswerkzeuge
10.3	Softwareentwurf und Entwicklung	Detaillierter Entwurf und Entwicklung (Software-Systementwurf): Entwerfen und Implementieren von Software, die die spezifizierten Anforderungen an die sicherheitsbezogene Software bezüglich des erforderlichen Sicherheits-Integritätslevels erfüllt, die analysierbar und überprüfbar ist und sich sicher modifizieren lässt	Hauptkomponenten und Teilsysteme des Entwurfs der Softwarearchitektur	7.4.5	Software-Architektur Entwurf; Werkzeuge und Programmierrichtlinien	Spezifikation des Software-Systementwurfs; Spezifikation des System-Integrationstests der Software
10.3	Softwareentwurf und Entwicklung	Detaillierter Entwurf und Entwicklung (Entwurf spezieller Softwaremodule): Entwerfen und Implementieren von Software, die die spezifizierten Anforderungen an die sicherheitsbezogene Software bezüglich des erforderlichen Sicherheits-Integritätslevels erfüllt, die analysierbar und überprüfbar ist und sich sicher modifizieren lässt.	Software-Systementwurf	7.4.5	Spezifikation des Software-Systementwurfs; Werkzeuge und Programmierrichtlinien	Spezifikation des Entwurfs der Softwaremodule; Testspezifikation der Softwaremodule
10.3	Softwareentwurf und Entwicklung	Detaillierte Codeimplementierung: Entwerfen und Implementieren von Software, die die spezifizierten Anforderungen an die sicherheitsbezogene Software bezüglich des erforderlichen Sicherheits-Integritätslevel erfüllt, die analysierbar und überprüfbar ist und sich sicher modifizieren lässt	Spezielle Softwaremodule	7.4.6	Spezifikation des Entwurfs der Softwaremodule; Werkzeuge und Programmierrichtlinien	Ausdruck des Source-Codes; Bericht der Code-Überprüfung

Tabelle 1 (fortgesetzt)

Phase des Sicherheitslebenszyklus		Ziele	Anwendungsbereich	Unterabschnitt für Anforderungen	Eingaben (benötigte Information)	Ergebnisse (Ausgaben, erzeugte Informationen)
Kastennummer in Bild 4	Titel					
10.3	Softwareentwurf und Entwicklung	Test der Softwaremodule: Verifizieren, dass die Anforderungen an die sicherheitsbezogene Software (in Begriffen der erforderlichen Funktionen der Software bezüglich der Sicherheit und der systematischen Eignung der Software) erreicht wurde Zeigen, dass jedes Softwaremodul seine bestimmungsgemäße Funktion und keine anderen ausführt Soweit angemessen, sicherstellen, dass die Konfiguration des PE-Systems mit Hilfe von Daten die spezifizierten Anforderungen zur systematischen Eignung der Software erfüllt	Softwaremodule	7.4.7	Spezifikation des Tests der Softwaremodule; Ausdruck des Source-Codes; Bericht der Codeüberprüfung	Ergebnisse des Tests der Softwaremodule; Verifizierte und getestete Softwaremodule
10.3	Softwareentwurf und Entwicklung	Softwareintegrationstest: Verifizieren, dass die Anforderungen an die sicherheitsbezogene Software (in Begriffen der erforderlichen Funktionen der Software bezüglich der Sicherheit und der systematischen Eignung der Software) erreicht wurde Zeigen, dass alle Softwaremodule, -elemente und Teilsysteme ordnungsgemäß zusammenarbeiten und ihre bestimmungsgemäßen Funktionen und keine anderen ausführen Soweit angemessen, sicherstellen, dass die Konfiguration des PE-Systems mit Hilfe von Daten die spezifizierten Anforderungen zur systematischen Eignung der Software erfüllt	Softwarearchitektur; Softwaresystem	7.4.8	Spezifikation der System-Integrationstests der Software	Ergebnisse der Softwareintegrationstests; Verifiziertes und getestetes Softwaresystem

Tabelle 1 (fortgesetzt)

Phase des Sicherheitslebenszyklus		Ziele	Anwendungsbereich	Unterabschnitt für Anforderungen	Eingaben (benötigte Information)	Ergebnisse (Ausgaben, erzeugte Informationen)
Kastennummer in Bild 4	Titel					
10.4	Integration der programmierbaren Elektronik (Hardware und Software)	Integration der Software in die Zielhardware der programmierbaren Elektronik; Zusammenführen der Software und Hardware in der sicherheitsbezogenen programmierbaren Elektronik, um ihre Verträglichkeit sicherzustellen und die Anforderungen des angestrebten Sicherheits-Integritätslevels zu erreichen	Hardware der programmierbaren Elektronik; Integrierte Software	7.5.2	Spezifikation des Integrationstests der Softwarearchitektur; Spezifikation des Integrationstests der Software und PE (auch durch IEC 61508-2 gefordert); Integrierte programmierbare Elektronik	Ergebnisse des Tests der Integration der Softwarearchitektur; Ergebnisse des Tests der Integration der programmierbaren Elektronik; Verifizierte und getestete Integrierte programmierbare Elektronik
10.5	Software-Betriebs- und Modifikationsverfahren	Bereitstellen der Informationen und Verfahren betreffs der notwendigen Software, um sicherzustellen, dass die funktionale Sicherheit des sicherheitsbezogenen E/E/PE-Systems für den Betrieb und bei Modifikationen erhalten bleibt	Wie oben	7.6.2	Alles wie oben, wenn zutreffend	Verfahren für Betrieb und Modifikationen der Software
10.6	Softwareaspekte bezüglich der Validierung der Sicherheit des Systems	Sicherstellen, dass das integrierte System den spezifizierten Anforderungen für die sicherheitsbezogene Software, mit dem vorgesehenen Sicherheits-Integritätslevel, entspricht	Wie oben	7.7.2	Validierungsplan für die Softwareaspekte der Sicherheit des Systems	Ergebnisse der Validierung der Software bezüglich der Sicherheit; Validierte Software
–	Softwaremodifikation	Modifikationen, Verbesserungen oder Erweiterungen an der validierten Software anzuleiten bei Sicherstellung, dass die erforderliche systematische Eignung der Software erhalten bleibt	Wie oben	7.8.2	Verfahren zur Modifikation der Software; Anforderung zur Modifikation der Software	Ergebnisse der Einflussanalyse der Softwaremodifikation; Logbuch der Softwaremodifikation
–	Softwareverifikation	Testen und Bewerten der Ausgaben einer gegebenen Phase des Software-Sicherheitslebenszyklus, um Korrektheit und Konsistenz in Bezug zu den Ausgaben und Normen, die als Eingaben zu dieser Phase geliefert wurden, sicherzustellen	Abhängig von der Phase	7.9.2	Geeigneter Plan der Verifikation (abhängig von der Phase)	Geeigneter Bericht der Verifikation (abhängig von der Phase)

Tabelle 1 (fortgesetzt)

Phase des Sicherheitslebenszyklus		Ziele	Anwendungsbereich	Unterabschnitt für Anforderungen	Eingaben (benötigte Information)	Ergebnisse (Ausgaben, erzeugte Informationen)
Kastennummer in Bild 4	Titel					
–	Beurteilung der funktionalen Sicherheit der Software	Untersuchen und zu einem Urteil gelangen über die Softwareaspekte der funktionalen Sicherheit, die durch das sicherheitsbezogene E/E/PE-System erreicht wird	Alle obigen Phasen	8	Plan für die Beurteilung der funktionalen Sicherheit der Software	Bericht über die Beurteilung der funktionalen Sicherheit der Software

7.2 Spezifikation der Anforderungen an die Sicherheit der Software

ANMERKUNG Diese Phase entspricht Kasten 10.1 in Bild 4.

7.2.1 Ziele

7.2.1.1 Das erste Ziel der Anforderungen dieses Unterabschnittes ist es, die Anforderungen an sicherheitsbezogene Software mit Begriffen der Anforderungen an die Funktionen der Software bezüglich der Sicherheit und der Anforderungen an die systematische Eignung der Software zu spezifizieren.

7.2.1.2 Das zweite Ziel der Anforderungen dieses Unterabschnittes ist es, die Anforderungen an die Funktionen der Software bezüglich der Sicherheit für jedes sicherheitsbezogene E/E/PE-System zu spezifizieren, das notwendig ist, um die erforderlichen Sicherheitsfunktionen auszuführen.

7.2.1.3 Das dritte Ziel der Anforderungen dieses Unterabschnittes ist es, Anforderungen an die systematische Eignung der Software für jedes sicherheitsbezogene E/E/PE-System zu spezifizieren, das notwendig ist, um den Sicherheits-Integritätslevel zu erreichen, der für jede dem sicherheitsbezogenen E/E/PE-System zugeordnete Sicherheitsfunktion spezifiziert worden ist.

7.2.2 Anforderungen

ANMERKUNG 1 Diese Anforderungen werden in den meisten Fällen durch eine Kombination von allgemeiner Embedded-Software und anwendungsspezifischer Software erfüllt. Es ist eine Kombination von beiden, die die erforderlichen Eigenschaften bereitstellt, die die folgenden Unterabschnitte erfüllen. Die exakte Trennung zwischen allgemeiner und anwendungsspezifischer Software hängt von der gewählten Softwarearchitektur ab (siehe 7.4.3).

ANMERKUNG 2 Für die Auswahl geeigneter Verfahren und Maßnahmen (siehe Anhang A und B), um die Anforderungen dieses Abschnittes zur Spezifikation der Anforderungen an die Sicherheit der Software zu implementieren, sollten folgende Eigenschaften berücksichtigt werden (siehe Anhang C für eine Anleitung der Interpretation von Eigenschaften und Anhang F der IEC 61508-7 zu informativen Definitionen):

- Vollständigkeit bezüglich der Sicherheitsbedürfnisse, die durch die Software angesprochen werden;
- Korrektheit bezüglich der Sicherheitsbedürfnisse, die durch die Software angesprochen werden;
- Freiheit von Spezifikationsfehlern einschließlich der Freiheit von Mehrdeutigkeiten;
- Verständlichkeit der Sicherheitsanforderungen;
- Freiheit von nachteiligen Beeinflussungen der durch die Software angesprochenen Sicherheitsbedürfnisse durch die Nicht-Sicherheitsfunktionen;
- Fähigkeit, die Basis für eine Verifikation und Validierung zu liefern.

ANMERKUNG 3 Die durch die Software angesprochenen Sicherheitsbedürfnisse sind die Menge von Sicherheitsfunktionen und zugehörigen Anforderungen an die Sicherheitsintegrität, die den Softwarefunktionen durch den Entwurf des E/E/PE zugeordnet werden. (Die vollständige Menge der Sicherheitsbedürfnisse des Systems ist größer und beinhaltet auch Sicherheitsfunktionen, die nicht von Software abhängen). Die Vollständigkeit der Spezifikation der Anforderungen an die Sicherheit der Software hängt wesentlich von der Auswirkung früherer Phasen des Systemlebenszyklus ab.

7.2.2.1 Wenn die Anforderungen an die sicherheitsbezogene Software für das sicherheitsbezogene E/E/PE-System bereits beschrieben worden sind (siehe Abschnitt 7 der IEC 61508-2), dann muss die Spezifikation der Software-Sicherheitsanforderungen nicht mehr wiederholt werden.

7.2.2.2 Die Spezifikation der Anforderungen an die sicherheitsbezogene Software muss sich von den spezifizierten Sicherheitsanforderungen des sicherheitsbezogenen E/E/PE-Systems (siehe IEC 61508-2, Abschnitt 7) und jeder Anforderung der Sicherheitsplanung herleiten (siehe [Abschnitt 6](#)). Diese Information muss dem Softwareentwickler zur Verfügung gestellt werden.

ANMERKUNG 1 Diese Anforderung bedeutet nicht, dass keine Iterationen zwischen dem Entwickler des E/E/PE-Systems und dem Entwickler der Software erfolgen (IEC 61508-2 und IEC 61508-3). Wenn die Anforderungen der sicherheitsbezogenen Software und die Softwarearchitektur präziser werden, kann eine Beeinflussung der Hardwarearchitektur des E/E/PE-Systems stattfinden. Deswegen ist eine enge Zusammenarbeit zwischen dem Hardware- und Softwareentwickler notwendig. Siehe [Bild 5](#).

ANMERKUNG 2 Wenn ein Softwareentwurf bereits existierende, wieder verwendbare Software beinhaltet, kann es sein, dass diese entwickelt worden ist, ohne die aktuelle System-Anforderungsspezifikation zu berücksichtigen. Siehe [7.4.2.12](#) zu Anforderungen an bereits existierende Software, um die Spezifikation der Anforderungen an die Sicherheit der Software zu erfüllen.

7.2.2.3 Die Spezifikation der Anforderungen an die sicherheitsbezogene Software muss hinreichend genau sein, um einen Entwurf und eine Ausführung zu ermöglichen, die die erforderliche Sicherheitsintegrität erreichen (einschließlich jeder Anforderung zur Unabhängigkeit, siehe 7.4.3 der IEC 61508-2) und um die Beurteilung der funktionalen Sicherheit zu erlauben.

ANMERKUNG Der Detaillierungsgrad der Spezifikation kann mit der Komplexität der Anwendung variieren. Eine angemessene Spezifikation für funktionales Verhalten kann Anforderungen an Genauigkeit, Zeitverhalten und Leistungsfähigkeit, Kapazität, Robustheit, Überlastungstoleranz und andere charakteristische Eigenschaften der speziellen Anwendung beinhalten.

7.2.2.4 Um Unabhängigkeit zu erreichen muss eine angemessene Analyse von Versagen aufgrund gemeinsamer Ursache durchgeführt werden. Wo mögliche Ausfallmechanismen festgestellt werden, müssen wirksame Schutzmaßnahmen getroffen werden.

ANMERKUNG Siehe [Anhang F](#) für Verfahren zur Erreichung der Unabhängigkeit von Software.

7.2.2.5 Der Softwareentwickler muss die Information in 7.2.2.2 bewerten, um sicherzustellen, dass die Anforderungen ausreichend spezifiziert sind. Insbesondere muss der Softwareentwickler folgendes berücksichtigen:

- a) die Sicherheitsfunktionen;
- b) die Konfiguration oder Architektur des Systems;
- c) die Anforderungen an die Sicherheitsintegrität der Hardware (programmierbare Elektronik, Sensoren, Aktoren);
- d) die Anforderungen der systematischen Eignung der Software;
- e) die Leistungsfähigkeit und Reaktionszeit;
- f) die Einrichtungen und Bedienerschnittstellen, einschließlich der vernünftigerweise vorhersehbaren Fehlanwendung.

ANMERKUNG Die Verträglichkeit zu allen bereits existierenden Anwendungen sollte berücksichtigt werden.

7.2.2.6 Falls nicht schon ausreichend in den Sicherheitsanforderungen des sicherheitsbezogenen E/E/PE-Systems definiert, müssen alle relevanten Betriebsarten der EUC, des E/E/PE-Systems und jedes Betriebsmittels oder Systems, das an das E/E/PE-System angeschlossen wird, in den speziellen Anforderungen der sicherheitsbezogenen Software beschrieben werden.

7.2.2.7 Die Spezifikation der Anforderungen an die Sicherheit der Software muss jede sicherheitsbezogene oder relevante Einschränkung zwischen der Hardware und der Software spezifizieren und beschreiben.

7.2.2.8 In dem Maße, wie es durch den Entwurf der E/E/PE Hardwarearchitektur erforderlich ist, und unter Berücksichtigung einer möglicherweise steigenden Komplexität muss die Spezifikation der Anforderungen an die Sicherheit der Software Folgendes berücksichtigen:

- a) die Selbstüberwachung der Software (für Beispiele siehe IEC 61508-7);
- b) die Überwachung der Hardware der programmierbaren Elektronik, der Sensoren und der Aktoren;
- c) die periodischen Tests der Sicherheitsfunktionen des Systems während des Betriebs;
- d) die Testbarkeit der Sicherheitsfunktionen während des Betriebs der EUC;
- e) die Softwarefunktionen zur Durchführung von Wiederholungsprüfungen und aller Diagnosetests, um die Anforderungen an die Sicherheitsintegrität des sicherheitsbezogenen E/E/PE-Systems zu erfüllen.

ANMERKUNG Eine steigende Komplexität als Ergebnis der obigen Betrachtungen kann erforderlich machen, die Architektur noch einmal zu überdenken.

7.2.2.9 Wenn das sicherheitsbezogene E/E/PE-System Nicht-Sicherheitsfunktion auszuführen hat, müssen die spezifizierten Anforderungen an die sicherheitsbezogene Software diese Nicht-Sicherheitsfunktion eindeutig herausstellen.

ANMERKUNG Für Anforderungen zur Nichtbeeinflussung zwischen Sicherheitsfunktionen und Nicht-Sicherheitsfunktion siehe [7.4.2.8](#) und [7.4.2.9](#).

7.2.2.10 Die Spezifikation der Anforderungen an die Sicherheit der Software muss die geforderten sicherheitstechnischen Eigenschaften des Produktes beschreiben, nicht aber die des Projektes. Dies wird durch die Sicherheitsplanung abgedeckt (siehe Abschnitt 6 der 61508-1). Mit Bezug zu [7.2.2.1](#) bis [7.2.2.9](#) muss Folgendes angemessen beschrieben werden:

- a) die Anforderungen an die folgenden Funktionen der Softwaresicherheit:
 - 1) Funktionen, die es der EUC ermöglichen, einen sicheren Zustand zu erreichen oder aufrechtzuerhalten;
 - 2) Funktionen zur Erkennung, Anzeige und Behandlung von Fehlern in der Hardware der programmierbaren Elektronik;
 - 3) Funktionen zur Erkennung, Anzeige und Behandlung von Fehlern der Sensoren und Aktoren;
 - 4) Funktionen zur Erkennung, Anzeige und Behandlung von Fehlern in der Software selbst (Selbstüberwachung der Software);
 - 5) Funktionen bezüglich der periodischen Online-Tests der Sicherheitsfunktionen (z. B. in der vorgesehenen betrieblichen Umgebung);
 - 6) Funktionen bezüglich der periodischen Offline-Tests der Sicherheitsfunktionen (z. B. in einer Umgebung, wo man den Sicherheitsfunktionen der EUC nicht vertrauen kann);
 - 7) Funktionen, die erlauben, Modifikationen am PE-System sicher durchzuführen;
 - 8) Schnittstellen zu nicht sicherheitsbezogenen Funktionen;
 - 9) Leistungsfähigkeit und Reaktionszeit;
 - 10) Schnittstellen zwischen der Software und dem PE-System;

ANMERKUNG 1 Diese beinhalten beides, Online- und Offline-Programmiereinrichtungen.

- 11) Sicherheitsbezogene Kommunikation (siehe IEC 61508-2, 7.4.11);
- b) die Anforderungen an die systematische Eignung der Software:

- 1) die Sicherheits-Integritätslevel für jede der Funktionen unter a) oben;

ANMERKUNG 2 Siehe IEC 61508-5, Anhang A, für Informationen, die sich mit der Zuordnung der Sicherheitsintegrität zu Softwareelementen befassen.

- 2) Anforderungen an die Unabhängigkeit zwischen Funktionen.

7.2.2.11 Wenn Software-Sicherheitsanforderungen durch Konfigurationsdaten beschrieben oder implementiert werden, müssen die Daten:

- a) konsistent zu den Sicherheitsanforderungen des Systems sein;
- b) ausgedrückt werden mit Ausdrücken des erlaubten Bereichs und autorisierten Kombinationen seiner Betriebsparameter;
- c) definiert werden in einer Weise, die kompatibel ist mit der zugrunde liegenden Software (z. B. Reihenfolge der Ausführung, Laufzeit, Datenstrukturen usw.)

ANMERKUNG 1 Diese Anforderung an die Anwendungsdaten ist besonders relevant bei datengesteuerten Anwendungen. Diese sind wie folgt gekennzeichnet: Der Source-Code ist bereits vorhanden und das Hauptziel der Entwicklung ist, Sicherheit zu geben, dass die Konfigurationsdaten das Verhalten, das von der Anwendung gefordert ist, korrekt festlegen. Es können komplexe Abhängigkeiten zwischen Dateneinheiten bestehen und die Gültigkeit der Daten kann sich über die Zeit verändern.

ANMERKUNG 2 Siehe [Anhang G](#) zur Anleitung für datengesteuerte Systeme.

7.2.2.12 Wo Daten die Schnittstelle zwischen Software und externen Systemen bestimmen, müssen folgende Leistungsmerkmale, zusätzlich zu IEC 61508-2, 7.4.11, betrachtet werden:

- a) die Notwendigkeit zur begrifflichen Konsistenz in der Definition der Daten;
- b) ungültige oder außerhalb des zulässigen Bereichs liegende Werte oder solche zur falschen Zeit;
- c) die Reaktionszeit und der Durchsatz, einschließlich der Maximalbelastung;
- d) die Ausführungszeit im ungünstigsten/günstigsten Fall und Verklemmungen;
- e) ein Überlauf und Unterschreiten der Kapazität des Datenspeichers.

7.2.2.13 Betriebsparameter müssen geschützt werden gegen:

- a) ungültige oder außerhalb des zulässigen Bereichs liegende Werte oder solche zur falschen Zeit;
- b) unbefugte Änderungen;
- c) Verfälschung.

ANMERKUNG 1 Der Schutz gegen unbefugte Änderungen sollte berücksichtigt werden durch sowohl Softwaremechanismen als auch Mechanismen ohne Software. Man beachte jedoch, dass der Schutz gegen unbefugte Änderungen nachteilige Auswirkungen auf die Sicherheit haben kann, wenn z. B. Änderungen schnell oder unter Stressbedingungen benötigt werden.

ANMERKUNG 2 Obwohl eine Person ein Teil eines sicherheitsbezogenen Systems sein kann (siehe Abschnitt 1 der IEC 61508-1), werden in dieser Norm Anforderungen an die menschliche Zuverlässigkeit in Bezug zum Entwurf von sicherheitsbezogenen E/E/PE-Systemen nicht im Detail betrachtet. Die folgenden menschlichen Erwägungen, wenn angemessen, sollten angesprochen werden:

- Das Informationssystem für den Bediener sollte eine bildhafte Darstellung und eine Ausdrucksweise verwenden, die ihm vertraut ist. Sie sollte klar, verständlich und frei von unnötigen Details und/oder Aspekten sein;
- Die dem Bediener angezeigte Information zum EUC sollte eng der physikalischen Anordnung der EUC folgen;
- Wenn für den Bediener mehrere Anzeigeninhalte möglich und/oder wenn die Auswirkungen des Bedieneingriffs nicht sofort ersichtlich sind, sollte die angezeigte Information für jeden Zustand der Anzeige oder des Ablaufs automatisch beinhalten, welcher Zustand des Ablaufs erreicht ist, welche Betätigungen möglich sind und welche möglichen Auswirkungen gewählt werden können.

7.3 Validierungsplan für die Softwareaspekte der Systemsicherheit

ANMERKUNG 1 Diese Phase entspricht dem Kasten 10.2 in Bild 4.

ANMERKUNG 2 Software kann üblicherweise nicht separat ohne ihre zugehörige Hardware und Systemumgebung validiert werden.

7.3.1 Ziel

Das Ziel der Anforderungen dieses Unterabschnittes ist es, einen Plan zur Validierung der sicherheitsbezogenen Software unter den Aspekten der Systemsicherheit zu entwickeln.

7.3.2 Anforderungen

7.3.2.1 Es muss eine Planung erstellt werden, die sowohl die Verfahrensschritte als auch die technischen spezifiziert, die darlegen, dass die Software ihre Sicherheitsanforderungen erfüllt.

7.3.2.2 Der Validierungsplan für die Softwareaspekte der Systemsicherheit muss Folgendes betrachten:

- a) die Einzelheiten, wann die Validierung stattfinden muss;
- b) die Einzelheiten der Personen, die die Validierung durchführen müssen;
- c) eine Kennzeichnung der wichtigen Betriebsarten der EUC mit:
 - 1) Vorbereitung zum Betrieb mit Einrichten und Justieren;
 - 2) Start-, Einlern-, Automatik-, Hand-, Halbautomatik-, Dauerbetrieb;
 - 3) Rücksetzen, Abschaltung, Instandhaltung;
 - 4) vernünftigerweise vorhersehbar anormale Zustände und vernünftigerweise vorhersehbare Fehlbedienung.
- d) eine Kennzeichnung der sicherheitsbezogenen Software, die vor Beginn der Inbetriebnahme für jede Betriebsart der EUC zu validieren ist;
- e) die Vorgehensweise für die Validierung (z. B. analytische Methoden, statistische Tests usw.);
- f) entsprechend Punkt e) Maßnahmen (Verfahren) und Vorgehensweisen, die eingesetzt werden müssen, um zu bestätigen, dass jede Sicherheitsfunktion mit den spezifizierten Anforderungen an die Funktionen bezüglich der Sicherheit und der spezifizierten Anforderungen an die systematische Eignung der Software übereinstimmt;
- g) die erforderlichen Umgebungsbedingungen, bei denen die Validierungen durchzuführen sind (z. B. kann dies bei Tests kalibrierte Werkzeuge und Einrichtungen beinhalten);
- h) die Kriterien für bestanden und nicht bestanden;
- i) die Verfahrensweisen und Prozeduren, um die Ergebnisse, besonders das Versagen, zu bewerten.

ANMERKUNG Diese Anforderungen basieren auf den allgemeinen Anforderungen in 7.8 der IEC 61508-1.

7.3.2.3 Die Validierung muss eine Begründung der verwendeten Vorgehensweise geben. Die technische Vorgehensweise für die Validierung der sicherheitsbezogenen Software muss folgende Informationen beinhalten:

- a) die Auswahl von manuellen oder automatisierten Verfahren oder beidem;
- b) die Auswahl von statischen oder dynamischen Verfahren oder beidem;
- b) die Auswahl von analytischen oder statistischen Verfahren oder beidem;
- c) die Auswahl der Akzeptanzkriterien, basierend auf objektiven Faktoren oder Beurteilungen durch Experten, oder beidem.

7.3.2.4 Als Teil des Verfahrens zur Validierung sicherheitsbezogener Softwareaspekte, müssen Bereich und Inhalt des Validierungsplans für die Softwareaspekte der Systemsicherheit mit der beurteilenden Person oder der Gruppe, die die beurteilende Person repräsentiert, abgestimmt werden, sofern dies durch den Si-

cherheits-Integritätslevel gefordert ist (siehe IEC 61508-1, Abschnitt 8). Dieses Verfahren muss auch eine Aussage zur Anwesenheit der beurteilenden Person während des Tests machen.

7.3.2.5 Die Kriterien für die Softwarevalidierung bestanden und nicht bestanden müssen beinhalten:

- a) die erforderlichen Eingangssignale mit ihrer Reihenfolge und ihren Werten;
- b) die erwarteten Ausgangssignale mit ihrer Reihenfolge und ihren Werten und
- c) andere Annahmekriterien, wie z. B. Speicherbelegung, Zeit- und Wertetoleranzen.

7.4 Softwareentwurf und Entwicklung

ANMERKUNG Diese Phase entspricht dem Kasten 10.3 in Bild 4.

7.4.1 Ziele

7.4.1.1 Das erste Ziel der Anforderungen dieses Unterabschnittes ist es, eine Softwarearchitektur zu schaffen, die die spezifizierten Anforderungen an die sicherheitsbezogene Software im Hinblick auf den geforderten Sicherheits-Integritätslevel erfüllt.

7.4.1.2 Das zweite Ziel der Anforderungen dieses Unterabschnittes ist es, die Anforderungen, die durch die Hardwarearchitektur des sicherheitsbezogenen E/E/PE-Systems an die Software gestellt werden, zu bewerten, einschließlich der Bedeutung der Wechselwirkungen zwischen der E/E/PE-Hardware und Software für die Sicherheit der EUC-Einrichtung.

7.4.1.3 Das dritte Ziel der Anforderungen dieses Unterabschnittes ist es, einen geeigneten Satz von Werkzeugen mit Sprachen und Compilern, Schnittstellen zum Laufzeitsystem, Anwenderschnittstellen, Datenformaten und Darstellungen für den geforderten Sicherheits-Integritätslevel auszuwählen, der über den gesamten Sicherheitslebenszyklus der Software Verifikation, Validierung, Beurteilung und Modifikation unterstützt.

7.4.1.4 Das vierte Ziel der Anforderungen dieses Unterabschnittes ist es, Software zu entwerfen und zu implementieren, die die spezifizierten Anforderungen an die sicherheitsbezogene Software bezüglich des geforderten Sicherheits-Integritätslevels erfüllt, analysierbar und verifizierbar ist und sich sicher modifizieren lässt.

7.4.1.5 Das fünfte Ziel der Anforderungen dieses Unterabschnittes ist es, zu verifizieren, dass die Anforderungen an die sicherheitsbezogene Software (ausgedrückt als geforderte Sicherheitsfunktionen und systematische Eignung) erreicht worden sind.

7.4.1.6 Das sechste Ziel der Anforderungen dieses Unterabschnittes ist es, sicherzustellen, dass, soweit angemessen, die Konfiguration des PE-Systems durch Daten die spezifizierten Anforderungen der systematischen Eignung der Software erfüllt.

7.4.2 Allgemeine Anforderungen

7.4.2.1 Abhängig von der Art der Softwareentwicklung kann die Verantwortlichkeit für die Übereinstimmung mit 7.4 beim Lieferanten einer sicherheitsbezogenen programmierbaren Einrichtung (z. B. Lieferanten einer SPS) allein, beim Anwender dieser Einrichtung (z. B. Entwickler der Anwendungssoftware) allein oder bei beiden liegen. Die Zuordnung der Verantwortlichkeit muss während der Sicherheitsplanung (siehe Abschnitt 6) erfolgen.

ANMERKUNG Siehe 7.4.3 zu den Aspekten der System- und Softwarearchitektur zur Zuordnung der Verantwortlichkeiten.

7.4.2.2 In Übereinstimmung mit dem geforderten Sicherheits-Integritätslevel und den speziellen technischen Anforderungen der Sicherheitsfunktion muss die gewählte Entwurfsmethode Eigenschaften beinhalten, die Folgendes erleichtern:

- a) die Abstraktion, Modularität und andere Eigenschaften, die die Komplexität begrenzen;

- b) die Beschreibung von:
 - 1) Funktionalität;
 - 2) Informationsfluss zwischen den Elementen;
 - 3) Reihenfolge und zeitlichen Zusammenhängen;
 - 4) zeitlichen Einschränkungen;
 - 5) Gleichzeitigkeit und synchronisiertem Zugriff auf gemeinsame Ressourcen;
 - 6) Datenstrukturen und deren Eigenschaften;
 - 7) Voraussetzungen für den Entwurf und ihre Abhängigkeiten;
 - 8) Behandlung von Ausnahmen;
 - 9) Voraussetzungen für den Entwurf (Vorbedingungen, Nachbedingungen, Invarianten);
 - 10) Kommentaren;
- c) verschiedenen Sichtweisen des Entwurfs einschließlich solcher von Struktur und Verhalten;
- d) das Verständnis durch die Entwickler und andere, die den Entwurf verstehen müssen;
- e) die Verifikation und Validierung.

7.4.2.3 Die Testbarkeit und Möglichkeit sicherer Modifikationen muss während des Entwurfs betrachtet werden, damit die Implementierung dieser Eigenschaften in das endgültige sicherheitsbezogene System erleichtert wird.

ANMERKUNG Beispiele sind Instandhaltungsverfahren für Maschinen und Prozessanlagen.

7.4.2.4 Die verwendete Entwurfsmethode muss Eigenschaften besitzen, die die Softwaremodifikationen erleichtern. Diese Eigenschaften beinhalten Modularität, Geheimnisprinzip (en: Information Hiding) und Kapselung (en: Encapsulation).

7.4.2.5 Die Darstellung des Entwurfs muss in einer eindeutig definierten Aufzeichnungsart erfolgen oder auf eindeutig definierte Eigenschaften beschränkt sein.

7.4.2.6 Soweit es praktisch möglich ist, muss der Entwurf den sicherheitsbezogenen Teil der Software einfach halten.

7.4.2.7 Der Softwareentwurf muss entsprechend dem erforderlichen Sicherheits-Integritätslevel die Überwachung des Kontroll- und Datenflusses beinhalten. Bei Versagenserkennung müssen geeignete Maßnahmen ergriffen werden.

7.4.2.8 Wenn die Software sowohl Sicherheitsfunktionen als auch Nichtsicherheitsfunktion umsetzt, muss die gesamte Software als sicherheitsbezogen behandelt werden, sofern nicht angemessene Maßnahmen sicherstellen, dass Versagen von Nichtsicherheitsfunktionen Sicherheitsfunktionen nicht nachteilig beeinflussen können.

7.4.2.9 Wenn die Software Sicherheitsfunktionen verschiedener Sicherheits-Integritätslevel umsetzt, muss die gesamte Software so behandelt werden, als ob sie zum höchsten Sicherheits-Integritätslevel gehört, sofern nicht ausreichende Unabhängigkeit zwischen den Sicherheitsfunktionen der verschiedenen Sicherheits-Integritätslevel im Entwurf gezeigt werden kann. Es muss dargelegt werden, dass entweder (1) die Unabhängigkeit sowohl im räumlichen und zeitlichen Bereich erreicht wird oder (2) jede Verletzung der Unabhängigkeit beherrscht wird. Die Rechtfertigung für diese Unabhängigkeit muss dokumentiert werden.

ANMERKUNG Siehe [Anhang F](#) für Verfahren zur Erreichung der Unabhängigkeit von Software.

7.4.2.10 Wenn die systematische Eignung eines Softwareelementes niedriger ist als der Sicherheits-Integritätslevel der Sicherheitsfunktion, die das Softwareelement unterstützt, so muss es in einer Kombination mit weiteren Elementen verwendet werden, so dass die systematische Eignung dieser Kombination mindestens dem Sicherheits-Integritätslevel der Sicherheitsfunktion gleichkommt.

7.4.2.11 Wo eine Sicherheitsfunktion unter Verwendung einer Kombination von Softwareelementen mit bekannter systematischer Eignung implementiert wird, müssen die Anforderungen an die systematische Eignung aus 7.4.3 der IEC 61508-2 für diese Kombination angewendet werden.

ANMERKUNG Zu unterscheiden ist zwischen (1) der *durchgehenden Sicherheitsfunktion*, die durch ein oder mehrere Elemente bereitgestellt wird und (2) der *Element-Sicherheitsfunktion*, die von jedem einzelnen Element bereitgestellt wird. Wo eine Zusammenstellung von zwei Elementen eine höhere systematische Eignung erreichen soll, muss jedes der Elemente in der Lage sein, das gefährliche Ereignis zu verhindern oder zu mildern. Für die Elemente besteht aber nicht die Anforderung, identische Element-Sicherheitsfunktionen haben. Weiter ist nicht gefordert, dass jedes der Elemente allein in der Lage ist, die von der Kombination geforderte Sicherheitsfunktion bereitzustellen.

BEISPIEL Die elektronische Geschwindigkeitsregelung einer Maschine mit der *durchgehenden Sicherheitsfunktion* „Vermeidung eines unerwarteten Anlaufs“ wird durch zwei Prozessoren realisiert. Die *Element-Sicherheitsfunktion* der primären Steuerung ist das ideale Verhalten der Geschwindigkeitsregelung. Die *Element-Sicherheitsfunktion* der sekundären Steuerung ist eine diversitäre Überwachung (siehe IEC 61508-7, C.3.4) mit einer Stillsetzung im Notfall. Die Kombination der beiden Prozessoren gibt ein höheres Vertrauen, dass die durchgehende Sicherheitsfunktion „Vermeidung eines unerwarteten Anlaufs“ erreicht wird.

7.4.2.12 Wenn ein bereits existierendes Softwareelement wiederverwendet wird, um eine Sicherheitsfunktion ganz oder teilweise zu realisieren, muss das Element beide nachfolgenden Anforderungen a) und b) für die systematische Sicherheitsintegrität erfüllen:

a) Einhaltung der Anforderungen eines der folgenden Konformitätspfade:

- Pfad 1_S: Konforme Entwicklung. Übereinstimmung mit den Anforderungen dieser Norm zur Vermeidung und Beherrschung systematischer Fehler in der Software;
- Pfad 2_S: Betriebsbewährtheit. Nachweise erbringen, dass das Element betriebsbewährt ist. Siehe 7.4.10 der IEC 61508-2;
- Pfad 3_S: Beurteilung einer nicht-konformen Entwicklung. Übereinstimmung mit 7.4.2.13.

ANMERKUNG 1 Die Pfade 1_S, 2_S und 3_S sind die wichtigen Konformitätspfade aus 7.4.2.2 c) der IEC 61508-2 mit besonderem Bezug zu Softwareelementen. Sie werden hier nur zur besseren Handhabbarkeit wiedergegeben, denn Verweisungen zurück zur IEC 61508-2 sollen gering gehalten werden.

ANMERKUNG 2 Siehe IEC 61508-4, 3.2.8. Die bereits existierende Software kann ein im Handel erhältliches Produkt sein oder sie wurde durch irgendeine Organisation für ein früheres Produkt oder System entwickelt. Die bereits existierende Software kann nach den Anforderungen dieser Norm entwickelt worden sein oder auch nicht.

ANMERKUNG 3 Anforderungen an bereits existierende Elemente gelten für eine Laufzeitbibliothek oder einen Interpreter.

b) Bereitstellung eines Sicherheitshandbuchs (siehe IEC 61508-2, Anhang D, und [Anhang D](#) dieser Norm), welches das Element so ausreichend genau und vollständig beschreibt, dass es die Begutachtung der Integrität einer bestimmten Sicherheitsfunktion ermöglicht, die vollständig oder teilweise vom bereits existierenden Softwareelement abhängt.

ANMERKUNG 4 Das Sicherheitshandbuch kann aus der Dokumentation, die vom Lieferanten des Elements beigestellt wird, und seinen Aufzeichnungen vom Herstellungsprozess abgeleitet werden oder es kann durch zusätzliche Qualifizierungstätigkeiten, die vom Entwickler des sicherheitsbezogenen Systems oder von Dritten durchgeführt werden, erstellt oder ergänzt werden. In einigen Fällen kann „Reverse Engineering“ notwendig sein, um eine angemessene Spezifikation oder eine Entwurfsdokumentation zu erzeugen, die den Anforderungen dieses Abschnittes entspricht, sofern nicht allgemein geltende gesetzliche Bestimmungen Vorrang haben (z. B. Urheberrecht oder Schutz geistigen Eigentums).

ANMERKUNG 5 Die Rechtfertigung eines Elements kann während der Planung der Sicherheit entwickelt werden (siehe [Abschnitt 6](#)).

7.4.2.13 Um den Pfad 3_S zu erfüllen, muss ein bereits existierendes Softwareelement alle folgenden Anforderungen a) bis i) erfüllen:

a) Die Spezifikation der Anforderungen an die Sicherheit der Software des Elements in dessen neuer Anwendung muss in der gleichen Genauigkeit dokumentiert werden, wie es durch diese Norm für jedes

sicherheitsbezogene Element mit der gleichen systematischen Eignung gefordert wird. Die Spezifikation der Anforderungen an die Sicherheit der Software muss sowohl das funktionale als auch das Sicherheitsverhalten, soweit es für das Element in dessen neuer Anwendung zutreffend ist und so, wie es in 7.2 spezifiziert ist, umfassen. Siehe [Tabelle A.1](#).

- b) Die Rechtfertigung zur Verwendung eines Softwareelements muss Beweise liefern, dass die wünschenswerten Sicherheitseigenschaften, die in den verwiesenen Unterabschnitten (d. h. [7.2.2](#), [7.4.3](#), [7.4.4](#), [7.4.5](#), [7.4.6](#), [7.4.7](#), [7.5.2](#), [7.7.2](#), [7.8.2](#), [7.9.2](#) und [Abschnitt 8](#)) festgelegt wurden, berücksichtigt worden sind, unter Begutachtung der Anleitung in [Anhang C](#).
- c) Der Entwurf des Elements muss mit einer derartigen Genauigkeit dokumentiert werden, dass ein ausreichender Nachweis der Übereinstimmung mit der Anforderungsspezifikation und den Anforderungen der systematischen Eignung gegeben ist. Siehe [7.4.3](#), [7.4.5](#) und [7.4.6](#) und die [Tabellen A.2](#) und [A.4](#) im [Anhang A](#).
- d) Der in [7.4.2.13 a\)](#) und [7.4.2.13 b\)](#) geforderte Nachweis muss die Integration der Software in die Hardware beinhalten. Siehe [7.5](#) und die [Tabelle A.6](#) im [Anhang A](#).
- e) Es muss einen Nachweis geben, dass das Element einer Verifikation und Validierung unter Verwendung eines systematischen Ansatzes mit dokumentierten Tests und Überprüfung aller Teile des Entwurfs des Elementes und dessen Codes unterzogen wurde. Siehe [7.4.7](#), [7.4.8](#), [7.5](#), [7.7](#) und [7.9](#) und die [Tabellen A.5 bis A.7](#) und [A.9](#) in [Anhang A](#) wie auch die zugehörigen Tabellen in [Anhang B](#).

ANMERKUNG 1 Eine positive Betriebserfahrung darf verwendet werden, um Black-Box- und probabilistische Testanforderungen zu erfüllen (siehe die [Tabellen A.7](#) und [B.3](#)).

- f) Wo das Softwareelement Funktionen, die im sicherheitsbezogenen System nicht notwendig sind, bereitstellt, muss der Nachweis erbracht werden, dass die nicht erwünschten Funktionen das E/E/PE-System nicht daran hindern werden, seine Sicherheitsanforderungen zu erfüllen.

ANMERKUNG 2 Möglichkeiten, diese Anforderung zu erfüllen, beinhalten:

- Entfernung der Funktionen aus dem Aufbau;
 - Deaktivierung der Funktionen;
 - geeignete Systemarchitektur (z. B. Aufteilung, Schutzmechanismen, Diversität, Prüfung der Glaubwürdigkeit von Ausgaben);
 - umfangreiche Tests.
- g) Es muss einen Nachweis geben, dass alle glaubwürdigen Versagensarten des Softwareelements identifiziert wurden und dass alle angemessenen Schadensbegrenzungsmaßnahmen implementiert worden sind.

ANMERKUNG 3 Angemessene Schadensbegrenzungsmaßnahmen beinhalten:

- geeignete Systemarchitektur (z. B. Aufteilung, Schutzmechanismen, Diversität, Glaubwürdigkeit der Prüfung von Ausgaben);
 - Behandlung von Ausnahmen.
- h) Die Planung zur Verwendung des Elements muss die Konfiguration des Softwareelements, der Software und der Hardware-Laufzeit-Umgebung und, wenn notwendig, die Konfiguration des Systems zum Kompilieren/Linken festlegen.
 - i) Die Begründung für die Verwendung des Elements darf nur für die Anwendungen gelten, für die die Annahmen erfüllt sind, die im Sicherheitshandbuch für konforme Objekte für dieses Element gemacht worden sind (siehe IEC 61508-2, [Anhang D](#), und [Anhang D](#)).

7.4.2.14 Dieser Unterabschnitt 7.4.2 muss, soweit geeignet, auf Daten und Sprachen, die Daten erzeugen, angewendet werden.

ANMERKUNG Für eine Anleitung zu datengesteuerten Systemen siehe [Anhang G](#).

- a) Wo ein PE-System aus bereits existierenden datenkonfigurierten Funktionalitäten besteht, um die speziellen Anforderungen der Anwendung zu erfüllen, muss der Entwurf der Anwendungssoftware dem Grad der Konfigurierbarkeit der Anwendung, bereits ausgelieferten Funktionalitäten und der Komplexität des sicherheitsbezogenen PE-Systems entsprechen.

- b) Wo die sicherheitsbezogene Funktionalität eines PE-Systems wesentlich oder hauptsächlich durch Konfigurationsdaten bestimmt wird, müssen angemessene Verfahren und Maßnahmen verwendet werden, um die Einbringung von Fehlern während des Entwurfs, der Produktion, dem Laden und der Modifikation der Konfigurationsdaten zu vermeiden und um sicherzustellen, dass die Konfigurationsdaten die Logik der Anwendung ordnungsgemäß darstellen.
- c) Die Spezifikation von Datenstrukturen muss sein:
 - 1) konsistent zu den funktionalen Anforderungen des Systems einschließlich der Anwendungsdaten;
 - 2) vollständig;
 - 3) widerspruchsfrei;
 - 4) so, dass die Datenstrukturen geschützt gegen Veränderung und Verfälschung sind.
- d) Besteht ein PE-System aus bereits existierender Funktionalität, die durch Daten konfiguriert wird, um speziellen Anwendungsanforderungen zu genügen, muss der Konfigurationsvorgang selbst entsprechend dokumentiert werden.

7.4.3 Anforderungen an den Entwurf der Softwarearchitektur

ANMERKUNG 1 Die Softwarearchitektur legt die wesentlichen Elemente und Teilsysteme der Software fest, wie sie verbunden sind und wie die erforderlichen Eigenschaften, insbesondere die Softwareintegrität, erreicht werden. Sie definiert auch das Gesamtverhalten der Software und wie Softwareelemente verbunden sind und deren Wechselwirkungen. Beispiele wesentlicher Softwareelemente sind Betriebssysteme, Datenbanken, EUC-Teilsysteme für Ein- und Ausgaben in Anlagen, Teilsysteme für die Kommunikation, Anwenderprogramm(e), Programmier- und Diagnosewerkzeuge usw.

ANMERKUNG 2 In bestimmten industriellen Bereichen wird die Softwarearchitektur Funktionsbeschreibung oder Spezifikation des funktionalen Entwurfs genannt (obwohl diese Dokumente auch die Hardware beinhalten können).

ANMERKUNG 3 In einigen Zusammenhängen von Anwenderprogrammen, insbesondere für SPSen (siehe Anhang E der IEC 61508-6), wird die Softwarearchitektur durch den Lieferanten als serienmäßige Eigenschaft des Produkts geliefert. Der Lieferant wird unter dieser Norm dazu angehalten, dem Anwender seiner Produkte die Übereinstimmung mit den Anforderungen in 7.4 zu versichern. Der Anwender passt die SPS unter Verwendung der standardisierten Programmiermöglichkeiten, z. B. Kontaktplan, auf die Anwendung an. Die Anforderungen von 7.4.3 bis 7.4.8 gelten weiter. Die Anforderung, die Softwarearchitektur zu beschreiben und zu dokumentieren, kann als die Information betrachtet werden, die der Anwender verwendet, um die SPS (oder Ähnliches) für die Anwendung auszuwählen.

ANMERKUNG 4 Vom Standpunkt der Sicherheit aus ist die Phase der Softwarearchitektur dort, wo die grundlegende Sicherheitsstrategie für die Software entwickelt wird.

ANMERKUNG 5 Obwohl die Normenreihe IEC 61508 numerische Ausfallgrenzwerte für Sicherheitsfunktionen festlegt, die durch ein sicherheitsbezogenes E/E/PE-System ausgeführt werden, ist die systematische Sicherheitsintegrität üblicherweise nicht quantifiziert (siehe IEC 61508-4, 3.5.6), und die Sicherheitsintegrität der Software (siehe 3.5.5 der IEC 61508-4) wird definiert als eine systematische Fähigkeit auf einer Vertrauensskala von 1-4 (siehe IEC 61508-4, 3.5.9). Diese Norm erkennt an, dass ein Softwareversagen sicher oder unsicher sein kann, in Abhängigkeit von der speziellen Verwendung der Software. Die System-/Softwarearchitektur ist so auszuführen, dass unsichere Ausfälle eines Elements durch Einschränkungen seitens der Architektur begrenzt werden und dass die Entwicklungsmethoden diese berücksichtigen sollten. Diese Norm wendet Entwicklungs- und Validierungsverfahren mit jeweils einer Strenge an, die qualitativ zur geforderten systematischen Eignung passt.

ANMERKUNG 6 Zur Auswahl angemessener Verfahren und Maßnahmen (siehe [Anhänge A und B](#)), um die Anforderungen dieses Abschnitts zu erfüllen (siehe [Anhang C](#) für eine Anleitung zur Interpretation von Eigenschaften und IEC 61508-7, Anhang F, für informative Definitionen) sollten folgende Eigenschaften des Entwurfs der Softwarearchitektur berücksichtigt werden:

- Vollständigkeit in Bezug auf die Spezifikation der Anforderungen an die Sicherheit der Software;
- Korrektheit in Bezug auf die Spezifikation der Anforderungen an die Sicherheit der Software;
- Freiheit von Entwurfsfehlern;
- Einfachheit und Verständlichkeit;
- Vorhersagbarkeit des Verhaltens;

- Nachprüfbarer und testbarer Entwurf;
- Fehlertoleranz;
- Schutz gegen Ausfälle infolge gemeinsamer Ursache durch externe Ereignisse.

7.4.3.1 In Abhängigkeit von der Art der Softwareentwicklung kann die Verantwortlichkeit für die Übereinstimmung mit 7.4.4 bei mehreren Beteiligten liegen. Die Zuordnung der Verantwortlichkeit muss während der Sicherheitsplanung dokumentiert werden (siehe IEC 61508-1, Abschnitt 6).

7.4.3.2 Der Entwurf der Softwarearchitektur muss ausführlich durch den Softwarelieferanten und/oder Softwareentwickler festgelegt werden. Der Entwurf der Softwarearchitektur muss:

- a) einen zusammenhängenden Satz von Verfahren und Maßnahmen (siehe [7.1.2.7](#)) auswählen und rechtfertigen, die notwendig sind, während der Phasen des Software-Sicherheitslebenszyklus die Spezifikation der Anforderungen an die Sicherheit der Software für den erforderlichen Sicherheits-Integritätslevel zu erfüllen. Diese Verfahren und Maßnahmen beinhalten Verfahren des Softwareentwurfs für sowohl Fehlertoleranz (in Übereinstimmung mit der Hardware) und Fehlervermeidung einschließlich (wo angemessen) Redundanz und Diversität;
- b) auf der Aufteilung in Elemente/Teilsysteme beruhen, wobei für jedes folgende Information bereitgestellt werden muss:
 - 1) ob die Elemente/Teilsysteme bereits verifiziert sind und wenn ja, die Bedingungen der Verifikation;
 - 2) ob diese Teilsysteme/Elemente sicherheitsbezogen sind oder nicht;
 - 3) die systematische Eignung der Software für das Teilsystem/Element.
- c) alle Wechselwirkungen der Software/Hardware festlegen und ausführlich deren Bedeutung bewerten;

ANMERKUNG Wo das Zusammenspiel der Software/Hardware bereits in der Systemarchitektur berücksichtigt wurde, reicht es aus, auf diese Systemarchitektur zu verweisen.

- d) eine Aufzeichnungsart zur Darstellung der Architektur verwenden, die eindeutig definiert oder auf eindeutig definierte Eigenschaften eingeschränkt ist;
- e) Eigenschaften für den Entwurf auswählen, die für die Aufrechterhaltung der Sicherheits-Integrität aller Daten verwendet werden. Diese Daten dürfen Eingangs- und Ausgangsdaten der Anlage, Kommunikationsdaten, Daten der Bedienerschnittstelle, Daten für die Instandhaltung und interne Daten der Datenbank umfassen;
- f) angemessene Integrationstests der Softwarearchitektur beschreiben, um sicherzustellen, dass die Softwarearchitektur die Spezifikation der Anforderungen an die Sicherheit der Software gemäß dem Sicherheits-Integritätslevel erfüllt.

7.4.3.3 Verlangte Änderungen an der Spezifikation der Anforderungen an die Sicherheit des E/E/PE-Systems (siehe [7.2.2](#)) nach Anwendung von 7.4.3.2 müssen mit dem Entwickler des E/E/PE abgestimmt und dokumentiert werden.

ANMERKUNG Iterationen zwischen der Hardware- und Softwarearchitektur sind unvermeidlich (siehe [Bild 5](#)), und es ist notwendig, Punkte wie z. B. die Testspezifikation zur Integration der Hardware und Software der programmierbaren Elektronik (siehe [7.5](#)) mit dem Hardwareentwickler zu besprechen.

7.4.4 Anforderungen an Werkzeuge einschließlich Programmiersprachen

ANMERKUNG Zur Auswahl angemessener Verfahren und Maßnahmen (siehe [Anhänge A und B](#)), um die Anforderungen dieses Abschnitts zu erfüllen, sollten folgende Eigenschaften (siehe [Anhang C](#) für eine Anleitung zur Interpretation von Eigenschaften und IEC 61508-7, Anhang F, für informative Definitionen) der Werkzeuge berücksichtigt werden:

- der Umfang, in welchem die Werkzeuge die Erstellung der Software mit den notwendigen Softwareeigenschaften, unterstützen;
- die Übersichtlichkeit der Bedienung und Funktionalität des Werkzeugs;
- die Korrektheit und Wiederholbarkeit der Ergebnisse.

7.4.4.1 Ein Online-Software-Werkzeug muss als ein Softwareelement des sicherheitsbezogenen Systems angesehen werden.

ANMERKUNG Siehe IEC 61508-4, 3.2.10 und 3.2.11, für Beispiele zu Online- und Offline-Werkzeugen.

7.4.4.2 Offline-Softwarewerkzeuge müssen ausgewählt werden als ein zusammenhängender Teil der Aktivitäten der Softwareentwicklung.

ANMERKUNG 1 Zu den Anforderungen an die Entwicklung des Softwarelebenszyklus siehe 7.1.2.

ANMERKUNG 2 Es sollten Offline-Werkzeuge verwendet werden, die die Entwicklung von Software angemessen unterstützen mit dem Ziel, die Integrität der Software durch Verringerung der Wahrscheinlichkeit, Fehler einzubringen oder solche nicht zu bemerken, zu erhöhen. Beispiele für Werkzeuge, die zu bestimmten Phasen des Software-Entwicklungszyklus gehören, sind:

- a) Transformations- oder Übersetzungs-Werkzeuge, die eine Software oder die Darstellung eines Entwurfs (z. B. Text oder Diagramm) von einer Abstraktionsebene in eine andere Ebene konvertieren: Werkzeuge zur Verbesserung des Entwurfs, Compiler, Assembler, Linker, Binder, Ladewerkzeuge und Code generierende Werkzeuge;
- b) Verifikations- und Validierungswerkzeuge wie statische Codeanalysatoren, Testabdeckungskontrollen, Assistenten zum Nachweis von Theoremen und Simulatoren;
- c) Diagnosewerkzeuge, um die Software unter Betriebsbedingungen zu überwachen und zu erhalten;
- d) Werkzeuge zur Infrastruktur, wie Systeme zur Unterstützung der Entwicklung;
- e) Konfigurationswerkzeuge, wie zum Beispiel Werkzeuge zur Versionsverwaltung;
- f) Werkzeuge für die Anwendungsdaten, die diese erzeugen oder aufrechterhalten und die notwendig sind, um Parameter zu definieren und Systemfunktionen zu instanzieren. Solche Daten beinhalten Funktionsparameter, Anzeigebereiche, Alarm- und Schaltschwellen, Ausgabezustände, die im Fall eines Ausfalls eingenommen werden, räumliche Aufteilung.

ANMERKUNG 3 Offline-Werkzeuge sollten zur Integration ausgewählt werden. In diesem Zusammenhang sind Werkzeuge integriert, wenn sie so zusammenarbeiten, dass die Ausgaben eines Werkzeugs Inhalte und Formate für die automatische Eingabe in ein nachgeordnetes Werkzeug liefern. Dies verringert die Möglichkeit menschlicher Fehler bei der Nachbearbeitung von Zwischenergebnissen.

ANMERKUNG 4 Offline-Werkzeuge sollten so ausgewählt werden, dass sie kompatibel sind mit den Bedürfnissen der Anwendung des sicherheitsbezogenen Systems und der integrierten Gruppe von Werkzeugen.

ANMERKUNG 5 Die Verfügbarkeit geeigneter Werkzeuge, die notwendige Leistung über die gesamte Lebensdauer des sicherheitsbezogenen E/E/PE-Systems zu liefern (z. B. Werkzeuge zur Unterstützung der Spezifikation, Entwurf, Implementierung, Dokumentation, Änderung), sollte berücksichtigt werden.

ANMERKUNG 6 Die Kompetenz der Anwender der ausgewählten Werkzeuge sollte berücksichtigt werden. Siehe Abschnitt 6 der IEC 61508-1 zu den Anforderungen an die Kompetenz.

7.4.4.3 Die Auswahl der Offline-Werkzeuge muss begründet werden.

7.4.4.4 Jedes Offline-Werkzeug der Klassen T2 und T3 muss eine Spezifikation oder Produktdokumentation haben, welche eindeutig sein Verhalten und Bedienungsanweisungen oder Einschränkungen seiner Verwendung beschreibt. Siehe 7.1.2 zu den Anforderungen an den Software-Entwicklungszyklus und IEC 61508-4, 3.2.11, zu Kategorien der Offline-Softwarewerkzeuge.

ANMERKUNG Diese „Spezifikation oder Produktdokumentation“ ist kein Sicherheitshandbuch für konforme Objekte (siehe Anhang D der 61508-2 und auch dieser Norm) für das Werkzeug selbst. Das Sicherheitshandbuch für konforme Objekte bezieht sich auf ein bereits bestehendes Element, das in das ausführbare sicherheitsbezogene System eingefügt wird. Wo ein bereits bestehendes Element durch ein T3-Werkzeug erzeugt und in ein ausführbares sicherheitsbezogenes System eingefügt worden ist, sollte jede wichtige Information (z. B. kann die Dokumentation für einen optimierenden Compiler darauf hindeuten, dass die Reihenfolge der Auswertung der Funktionsparameter nicht gewährleistet ist) aus der „Spezifikation oder Produktdokumentation“ des Werkzeugs in das Sicherheitshandbuch für konforme Objekte übernom-

men werden, die es ermöglicht, die Integrität der Sicherheitsfunktion, die ganz oder teilweise zum eingefügten Element gehört, beurteilen zu können.

7.4.4.5 Es muss eine Beurteilung für Offline-Werkzeuge der Klassen T2 und T3 durchgeführt werden, um den Grad des dem Werkzeug entgegengebrachten Vertrauens und die potentiellen Ausfallmechanismen, die die ausführbare Software beeinflussen könnten, zu bestimmen. Wo solche Ausfallmechanismen festgestellt werden, müssen geeignete Schadensbegrenzungsmaßnahmen getroffen werden.

ANMERKUNG 1 Die Anwendung von HAZOP (en.: Hazard and Operability) auf Software ist ein Verfahren, um die Auswirkungen potentiellen Versagens von Softwarewerkzeugen zu analysieren.

ANMERKUNG 2 Beispiele von Schadensbegrenzungsmaßnahmen beinhalten: Vermeidung bekannter Fehler, eingeschränkte Verwendung von Werkzeugfunktionalitäten, Kontrolle der Ausgaben, Verwendung diversitärer Werkzeuge für den gleichen Zweck.

7.4.4.6 Für jedes Werkzeug der Klasse T3 muss der Nachweis verfügbar sein, dass das Werkzeug mit seiner Spezifikation oder Dokumentation übereinstimmt. Der Nachweis kann auf einer geeigneten Kombination erfolgreicher Verwendung in ähnlichen Umgebungen und ähnlichen Anwendungen (innerhalb der eigenen Organisation oder anderer Organisationen) beruhen und auf einer Validation des Werkzeugs wie in 7.4.4.7 beschrieben.

ANMERKUNG 1 Eine Versionshistorie kann Vertrauen zur Reife des Werkzeugs geben, und ein Bericht über die Fehler/Unklarheiten sollte berücksichtigt werden, wenn das Werkzeug in der neuen Entwicklungsumgebung verwendet wird.

ANMERKUNG 2 Die für T3 aufgeführten Nachweise können auch für T2-Werkzeuge zum Beurteilen der Korrektheit ihrer Ergebnisse verwendet werden.

7.4.4.7 Die Ergebnisse der Validierung des Werkzeugs müssen dokumentiert werden unter Einbeziehung folgender Ergebnisse:

- a) einer chronologischen Aufzeichnung der Validierungsaktivitäten;
- b) der Version des verwendeten Handbuchs zum Werkzeug;
- c) den validierten Funktionen des Werkzeugs;
- d) der verwendeten Werkzeuge und Einrichtungen;
- e) der Ergebnisse der Validierungsaktivitäten; die dokumentierten Ergebnisse der Validierung müssen entweder darlegen, dass die Software die Validierung bestanden hat, oder die Gründe warum nicht;
- f) Testfälle und deren Ergebnisse für nachfolgende Analysen;
- g) Unterschiede zwischen erwarteten und tatsächlichen Ergebnissen.

7.4.4.8 Wo der Nachweis der Übereinstimmung nach 7.4.4.6 nicht verfügbar ist, müssen effektive Maßnahmen getroffen werden, um Versagensfälle des ausführbaren sicherheitsbezogenen Systems, die dem Werkzeug zuzuschreiben sind, zu beherrschen.

ANMERKUNG Ein Beispiel einer Maßnahme könnte die Generierung eines divers-redundanten Codes sein, der die Detektion und Beherrschung eines Versagens des ausführbaren sicherheitsbezogenen Systems ermöglicht, das durch einen Fehler verursacht wurde, der von einem Übersetzer in das ausführbare sicherheitsbezogene System eingebracht wurde.

7.4.4.9 Die Verträglichkeit der Werkzeuge einer integrierten Gruppe muss bestätigt werden.

ANMERKUNG Werkzeuge sind integriert, wenn sie so zusammenarbeiten, dass die Ausgaben eines Werkzeugs geeignete Inhalte und Formate für die automatische Eingabe in ein nachgeordnetes Werkzeug liefern. Dies verringert die Möglichkeit menschlicher Fehler bei der Nachbearbeitung von Zwischenergebnissen. Siehe IEC 61508-7, B.3.5.

7.4.4.10 In dem Umfang, wie durch den Sicherheits-Integritätslevel gefordert, muss die gewählte Software oder die Repräsentation des Entwurfs (einschließlich einer Programmiersprache):

- a) einen Übersetzer haben, der für die Eignung der Anwendung beurteilt worden ist, wo angemessen, nach einer Internationalen oder nationalen Norm;
- b) nur definierte Sprachfunktionen verwenden;
- c) den Besonderheiten der Anwendung entsprechen;
- d) Eigenschaften enthalten, die die Erkennung von Entwurfs- und Programmierfehlern erleichtern;
- e) Eigenschaften unterstützen, die zur Entwurfsmethode passen.

ANMERKUNG 1 Eine Programmiersprache ist eine Klasse von Darstellungen einer Software oder eines Entwurfs. Ein Übersetzer konvertiert eine Software oder die Darstellung eines Entwurfs (z. B. Text oder ein Diagramm) von einer Abstraktionsebene in eine andere. Beispiele von Übersetzern beinhalten: Werkzeuge zur Verfeinerung des Entwurfs, Compiler, Assembler, Linker, Binder, Ladewerkzeuge und codegenerierende Werkzeuge.

ANMERKUNG 2 Die Beurteilung eines Übersetzers kann für die Anwendung in einem speziellen Projekt oder für eine Klasse von Anwendungen erfolgen. Im letzteren Fall sollten alle notwendigen Informationen zum Werkzeug (das „Spezifikations- oder Produkthandbuch“, siehe 7.4.4.4) unter Berücksichtigung der beabsichtigten und angemessenen Verwendung dem Anwender des Werkzeugs zur Verfügung stehen. Die Beurteilung des Werkzeugs für ein spezielles Projekt kann dann reduziert werden auf die Überprüfung einer allgemeinen Eignung des Werkzeugs für dieses Projekt und Übereinstimmung mit dem „Spezifikations- oder Produkthandbuch“ (d. h. richtige Anwendung des Werkzeugs). Die richtige Anwendung kann zusätzliche Verifikationen innerhalb des Projektes beinhalten.

ANMERKUNG 3 Eine Validierungsreihe (das bedeutet eine Reihe von Testprogrammen, deren ordnungsgemäße Übersetzung im Voraus bekannt ist) kann verwendet werden, die Tauglichkeit eines Übersetzers für die Anwendung nach definierten Kriterien zu bewerten. Diese Kriterien sollten funktionale und nichtfunktionale Anforderungen beinhalten. Für die funktionalen Anforderungen an den Übersetzer können dynamische Tests ein wesentliches Verfahren zur Validierung sein. Wenn möglich sollte eine automatische Testreihe verwendet werden.

7.4.4.11 Wenn 7.4.4.10 nicht vollständig erfüllt werden kann, muss die Eignung für die Anwendung der Sprache und jede zusätzlichen Maßnahme, die irgendeinen bekannten Mangel der Sprache betrifft, begründet werden.

7.4.4.12 Programmiersprachen für die Entwicklung von sicherheitsbezogener Software müssen in Übereinstimmung mit angemessenen Programmierrichtlinien verwendet werden.

ANMERKUNG Siehe IEC 61508-7 als Orientierungshilfe zu Aspekten von Programmierrichtlinien mit Bezug zur Softwaresicherheit.

7.4.4.13 Eine Programmierrichtlinie für eine Programmiersprache muss gute Programmiertechniken beschreiben, unsichere Sprachenmerkmale verbieten (z. B. undefinierte Sprachmerkmale, unstrukturierte Entwürfe usw.), die Codeverständlichkeit fördern, Verifikation und Tests erleichtern und die Verfahren für die Dokumentation des Source-Codes beschreiben. Wo durchführbar, müssen mindestens folgende Informationen im Source-Code enthalten sein:

- a) die juristische Person (z. B. die Firma, Autor(en) usw.);
- b) die Beschreibung;
- c) die Ein- und Ausgaben;
- d) die Vorgeschichte des Konfigurationsmanagements.

7.4.4.14 Wo automatische Codeerzeugung oder ähnliche automatische Übersetzungen Einsatz finden, muss die Eignung des automatischen Übersetzers für die Entwicklung des sicherheitsbezogenen Systems an dem Punkt des Lebenszyklus der Entwicklung beurteilt werden, an dem die Werkzeuge ausgewählt werden.

7.4.4.15 Wo Offline-Werkzeuge der Klassen T2 und T3 etwas in der Basiskonfiguration erzeugen, muss das Konfigurationsmanagement sicherstellen, dass Informationen zu den Werkzeugen in der Basiskonfiguration aufgezeichnet werden. Dies beinhaltet im Einzelnen:

- a) die Bezeichnung des Werkzeugs und seine Version;
- b) die Bezeichnung der Einzelheiten der Basiskonfiguration, für die die Werkzeugversion verwendet worden ist;

- c) die Art, wie das Werkzeug verwendet wurde (einschließlich der Parameter des Werkzeugs, Optionen und verwendete Skripte) für jede Eigenschaft der Basiskonfiguration;

ANMERKUNG Das Ziel dieses Abschnitts ist es, die Wiederherstellung der Basiskonfiguration zu ermöglichen.

7.4.4.16 Das Konfigurationsmanagement muss sicherstellen, dass für Werkzeuge in den Klassen T2 und T3 nur geeignete Versionen eingesetzt werden.

7.4.4.17 Das Konfigurationsmanagement muss sicherstellen, dass nur zueinander und zum sicherheitsbezogenen System verträgliche Werkzeuge eingesetzt werden.

ANMERKUNG Die Hardware des sicherheitsbezogenen Systems kann den Softwarewerkzeugen ebenfalls Randbedingungen zur Verträglichkeit auferlegen, z. B. muss der Emulator eines Prozessors ein exaktes Modell der Prozessorelektronik sein.

7.4.4.18 Jede neue Version eines Offline-Werkzeugs muss qualifiziert werden. Diese Qualifikation kann auf dem Vertrauen zu einer Vorgängerversion basieren, wenn ausreichende Nachweise vorliegen, dass:

- a) die funktionalen (falls vorhanden) Unterschiede die Verträglichkeit mit dem Rest der Gruppe von Werkzeugen nicht berührt und;
- b) die neue Version wahrscheinlich keine wesentlichen neuen, unbekannten Fehler enthält.

ANMERKUNG Vertrauen, dass die neue Version wahrscheinlich keine wesentlichen neuen und unbekannten Fehler enthält, kann auf (1) einer eindeutigen Identifikation der durchgeführten Änderungen, (2) einer Analyse der an der neuen Version durchgeführten Validierungs- und Verifizierungstätigkeiten und (3) allen bestehenden operativen Erfahrungen anderer Anwender, die relevant für die neue Version sind, beruhen.

7.4.4.19 In Abhängigkeit von der Natur der Softwareentwicklung kann die Verantwortlichkeit für die Übereinstimmung mit 7.4.4 auf mehreren Beteiligten ruhen. Die Aufteilung der Verantwortlichkeiten muss während der Sicherheitsplanung dokumentiert werden (siehe IEC 61508-1, Abschnitt 6).

7.4.5 Anforderungen an den detaillierten Entwurf und die Entwicklung – Softwaresystementwurf

ANMERKUNG 1 Als detaillierter Entwurf wird hier der Softwaresystementwurf verstanden: die Aufteilung der Hauptelemente der Architektur in ein System von Softwaremodulen, das Entwerfen von einzelnen Softwaremodulen und die Kodierung. In kleinen Anwendungen können der Softwaresystementwurf und der Architekturentwurf zusammengefasst werden.

ANMERKUNG 2 Die Art des detaillierten Entwurfs und der Entwicklung hängt von der Art der Softwareentwicklung und der Softwarearchitektur ab (siehe 7.4.3). In einigen Arten der Anwendungsprogrammierung, z. B. Kontaktplan und Funktionsbausteine, kann der detaillierte Entwurf mehr als Konfigurieren denn als Programmieren angesehen werden. Es ist jedoch ein guter Stil, die Software strukturiert zu entwerfen, einschließlich der Organisation der Software in einer modularen Struktur, die (soweit möglich) sicherheitsbezogene Teile abteilt, mit Bereichsprüfung und weiteren Eigenschaften, die Schutz gegen fehlerhafte Dateneingaben bieten, unter Verwendung bereits verifizierter Softwaremodule und Anfertigung eines Entwurfs, der die spätere Softwaremodifikationen erleichtert.

ANMERKUNG 3 Zur Auswahl angemessener Verfahren und Maßnahmen (siehe [Anhänge A und B](#)), um die Anforderungen dieses Abschnitts zu erfüllen, sollten folgende Eigenschaften (siehe [Anhang C](#) für eine Anleitung zur Interpretation von Eigenschaften und IEC 61508-7, Anhang F, für informative Definitionen) des Entwurfs und der Entwicklung berücksichtigt werden:

- Vollständigkeit in Bezug auf die Spezifikation der Anforderungen an die Sicherheit der Software;
- Korrektheit in Bezug auf die Spezifikation der Anforderungen an die Sicherheit der Software;
- Freiheit von Entwurfsfehlern;
- Einfachheit und Verständlichkeit;
- Vorhersagbarkeit des Verhaltens;
- nachprüfbarer und testbarer Entwurf;
- Fehlertoleranz/Fehlererkennung;