

	DIN EN 61508-3 (VDE 0803-3)	DIN
	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.	VDE

ICS 35.240.50

Einsprüche bis 2009-08-31

Vorgesehen als Ersatz für
DIN EN 61508-3
(VDE 0803-3):2002-12
Ersatz für
E DIN IEC 61508-3
(VDE 0803-3):2006-07

Entwurf

**Funktionale Sicherheit sicherheitsbezogener
elektrischer/elektronischer/programmierbarer elektronischer Systeme –
Teil 3: Anforderungen an Software
(IEC 65A/524/CDV:2008);
Deutsche Fassung FprEN 61508-3:2008**

Functional safety of electrical/electronic/programmable electronic safety-related systems –
Part 3: Software requirements
(IEC 65A/524/CDV:2008);
German version FprEN 61508-3:2008

Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables
relatifs à la sécurité –
Partie 3: Prescriptions concernant les logiciels
(CEI 65A/524/CDV:2008);
Version allemande FprEN 61508-3:2008

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2009-06-08 wird der Öffentlichkeit zur Prüfung und Stellungnahme vorgelegt.

Weil die beabsichtigte Norm von der vorliegenden Fassung abweichen kann, ist die Anwendung dieses Entwurfes besonders zu vereinbaren.

Stellungnahmen werden erbeten

- vorzugsweise als Datei per E-Mail an **dke@vde.com** in Form einer Tabelle. Die Vorlage dieser Tabelle kann im Internet unter **www.dke.de/stellungnahme** abgerufen werden
- oder in Papierform an die DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE, Stresemannallee 15, 60596 Frankfurt am Main.

Die Empfänger dieses Norm-Entwurfs werden gebeten, mit ihren Kommentaren jegliche relevante Patentrechte, die sie kennen, mitzuteilen und unterstützende Dokumentationen zur Verfügung zu stellen.

Gesamtumfang 243 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE

Beginn der Gültigkeit

Diese Norm gilt ab ...

Nationales Vorwort

Die Deutsche Fassung des europäischen Dokuments FprEN 61508-3:2008 „Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme – Teil 3: Anforderungen an Software“ (Entwurf in der Umfrage) ist unverändert in diesen Norm-Entwurf übernommen worden.

Die Internationale Elektrotechnische Kommission (IEC) und das Europäische Komitee für Elektrotechnische Normung (CENELEC) haben vereinbart, dass ein auf IEC-Ebene erarbeiteter Entwurf für eine Internationale Norm zeitgleich (parallel) bei IEC und CENELEC zur Umfrage (CDV-Stadium) und Abstimmung als FDIS (en: Final Draft International Standard) bzw. Schluss-Entwurf für eine Europäische Norm gestellt wird, um eine Beschleunigung und Straffung der Normungsarbeit zu erreichen. Dem entsprechend ist das internationale Dokument IEC 65A/524/CDV:2008 „Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 3: Software requirements“ unverändert in den Entwurf FprEN 61508-3:2008 übernommen worden.

Da die Deutsche Fassung noch nicht endgültig mit der Englischen und Französischen Fassung abgeglichen ist, ist die englische Originalfassung des IEC-CDV entsprechend der diesbezüglich durch die IEC erteilten Erlaubnis beigelegt. Die Nutzungsbedingungen für den deutschen Text des Norm-Entwurfes gelten gleichermaßen auch für den englischen IEC-Text.

Das internationale Dokument wurde vom SC 65A „System aspects“ der Internationalen Elektrotechnischen Kommission (IEC) erarbeitet.

Bei der Abstimmung zu dem Europäischen Schluss-Entwurf bei CENELEC und dem Internationalen Schluss-Entwurf bei IEC [Final Draft International Standard (FDIS)] sind jeweils nur „JA/NEIN“-Entscheidungen möglich, wobei „NEIN“-Entscheidungen fundiert begründet werden müssen. Dokumente, die bei CENELEC als Europäische Norm angenommen und ratifiziert werden, sind unverändert als Deutsche Normen zu übernehmen.

Für diesen Norm-Entwurf 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.

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.

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
–	ISO/IEC Guide 51:1999	–	–
–	IEC Guide 104:1997	–	–
FprEN61508-1:2008	65A/522/CDV:2008	E DIN EN 61805-1 (VDE 0803-1) in Vorbereitung	(VDE 0803-1)
FprEN61508-2:2008	65A/523/CDV:2008	E DIN EN 61805-2 (VDE 0803-2) in Vorbereitung	(VDE 0803-2)
FprEN61508-4:2008	65A/525/CDV:2008	E DIN EN 61805-4 (VDE 0803-4) in Vorbereitung	(VDE 0803-4)

Nationaler Anhang NB (informativ)

Literaturhinweise

E DIN EN 61805-1 (VDE 0803-1), *Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme – Teil 1: Allgemeine Anforderungen*

E DIN EN 61805-2 (VDE 0803-2), *Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme – Teil 2: Anforderungen an sicherheitsbezogene elektrische/elektronische/programmierbare elektronische Systeme*

E DIN EN 61805-4 (VDE 0803-4), *Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme – Teil 4: Begriffe und Abkürzungen*

— **Entwurf** —

E DIN EN 61508-3 (VDE 0803-3):2009-06

– Leerseite –

Deutsche Fassung

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

Inhalt

	Seite
Einleitung	4
2 Normative Verweisungen.....	9
3 Begriffe und Abkürzungen	9
4 Übereinstimmung mit dieser Norm	9
5 Dokumentation.....	9
6 Zusätzliche Anforderungen an die Software zum Management der funktionalen Sicherheit.....	9
6.1 Ziele	9
6.2 Anforderungen	9
7 Anforderungen an den Software-Sicherheitslebenszyklus	10
7.1 Allgemeines	10
7.2 Spezifikation der Software-Sicherheitsanforderungen	19
7.3 Validierungsplan für die Softwareaspekte der Systemsicherheit	22
7.4 Softwareentwurf und Entwicklung	23
7.5 Integration der programmierbaren Elektronik (Hardware und Software)	35
7.6 Software-Betriebs- und Modifikationsverfahren	36
7.7 Softwareaspekte bezüglich der Validierung der Systemsicherheit	37
7.8 Softwaremodifikation	38
7.9 Software-Verifikation.....	40
8 Beurteilung der funktionalen Sicherheit.....	44
Anhang A (normativ) Leitfaden für die Auswahl der Verfahren und Maßnahmen	46
A.1 Allgemein	46
Anhang B (informativ) Detailtabellen	56
B.1 Allgemein	56
Anhang C (informativ) Eigenschaften der systematischen Eignung der Software	61
C.1 Einleitung	61
C.2 Eigenschaften der systematischen Integrität.....	63
C.3 Eigenschaften der systematischen Integrität. Ausführliche Tabellen.....	98
Anhang D (normativ) Sicherheitshandbuch für konforme Objekte: zusätzliche Anforderungen an Softwareelemente.....	112
Anhang E (informativ) Beziehungen zwischen der IEC 61508- 2 und IEC 61508-3	115
E.1 Wichtigkeit der IEC 61508-2 für Software-Hersteller.....	115

— Entwurf —

E DIN EN 61508-3 (VDE 0803-3):2009-06
FprEN 61508-3:2008

	Seite
Anhang F (informativ) Verfahren zum Erreichen der Nicht-Beeinflussung zwischen Softwareelementen auf einem Einzelcomputer	117
F.1 Ziel / Einleitung	117
F.2 Definitionsbereich des Verhaltens	117
F.3 Analyse ursächlicher Faktoren	117
F.4 Erreichung räumlicher Unabhängigkeit	118
F.5 Erreichung zeitlicher Unabhängigkeit	118
F.6 Anforderungen an Hilfssoftware	119
F.7 Unabhängigkeit von Softwaremodulen: Aspekte zu Programmiersprachen	119
Anhang G (informativ) Leitlinien zur Anpassung des Lebenszyklusses im Zusammenhang mit datengesteuerten Systemen	122
G.1 Datengesteuert: Systemteil und Anwendungsteil	122
G.2 Eingeschränkter Freiheitsgrad der Konfiguration, eingeschränkte Anwendungskonfigurierbarkeit	123
G.3 Eingeschränkter Freiheitsgrad der Konfiguration, vollständige Anwendungskonfigurierbarkeit	124
G.4 Eingeschränkter Freiheitsgrad der Programmierung, eingeschränkte Anwendungskonfigurierbarkeit	124
G.5 Eingeschränkter Freiheitsgrad der Programmierung, vollständige Anwendungskonfigurierbarkeit	125
G.6 Vollständige Funktionalität der Programmierung/Konfiguration, eingeschränkte Anwendungskonfigurierbarkeit und	125
G.7 Vollständige Funktionalität der Programmierung/Konfiguration, vollständige Anwendungskonfigurierbarkeit	125
Literaturhinweise	126
Bilder	
Bild 1 – Gesamtrahmen dieser Norm	8
Bild 2 – E/E/PE-System-Sicherheitslebenszyklus (in der Realisierungsphase)	12
Bild 3 – Software-Sicherheitslebenszyklus (in der Realisierungsphase)	12
Bild 4 – Beziehung zwischen der IEC 61508-2 und IEC 61508-3 und deren Anwendungsbereiche	13
Bild 5 – Systematische Eignung der Software und der Entwicklungslebenszyklus (das V-Model)	13
Bild G.1 – Freiheitsgrad der Komplexität datengesteuerter Systeme	123
Tabellen	
Tabelle 1 – Software-Sicherheitslebenszyklus: Überblick	14
Tabelle A.1 – Spezifikation der Software-Sicherheitsanforderung	47
Tabelle A.2 – Softwareentwurf und Softwareentwicklung: Entwurf der Software-Architektur	48
Tabelle A.3 – Softwareentwurf und Softwareentwicklung: Hilfswerkzeuge und Programmiersprachen	49
Tabelle A.4 – Softwareentwurf und Softwareentwicklung: detaillierter Entwurf	50
Tabelle A.6 – Integration der programmierbaren Elektronik (Hardware und Software)	51
Tabelle A.7 – Softwareaspekte bezüglich der Validierung der Systemsicherheit	52
Tabelle A.8 – Modifikation	53
Tabelle A.9 – Software-Verifikation	54

	Seite
Tabelle A.10 – Beurteilung der funktionalen Sicherheit.....	55
Tabelle B.2 – Dynamische Analyse und Test.....	56
Tabelle B.3 – Funktionstest und Black-Box-Test.....	57
Tabelle B.4 – Ausfall-/Versagensanalyse	57
Tabelle B.5 – Modellierung	58
Tabelle B.6 – Leistungstest.....	58
Tabelle B.7 – Semiformale Methoden	59
Tabelle B.8 – Statische Analyse	59
Tabelle B.9 – Modularer Ansatz.....	60
Tabelle C-A.1 – Eigenschaften der systematischen Integrität. Spezifikation der Software- Sicherheitsanforderungen	63
Tabelle C-A.2 – Eigenschaften für die systematische Integrität. Software-Entwurf und -Entwicklung: Entwurf der Software-Architektur.....	67
Tabelle C-A.3 – Eigenschaften für die systematische Integrität. Software Entwurf und Entwicklung: unterstützende Werkzeuge und Programmiersprache.....	84
Tabelle C-A.4 – Eigenschaften für die systematische Integrität. Software Entwurf und Entwicklung: Detaillierter Entwurf	85
Tabelle C-A.5 – Eigenschaften für die systematische Integrität. Software Entwurf und Entwicklung: Test der Softwaremodule und Integration	88
Tabelle C-A.6 – Eigenschaften für die systematische Integrität. Integration der programmierbaren Elektronik (Hardware und Software).....	90
Tabelle C-A.7 – Eigenschaften für die systematische Integrität. Softwareaspekte bezüglich der Validierung der Systemsicherheit	91
Tabelle C-A.8 – Eigenschaften für die systematische Integrität. Softwaremodifikation	92
Tabelle C-A.9 – Eigenschaften für die systematische Integrität. Software-Verifikation.....	94
Tabelle C-A.10 – Eigenschaften für die systematische Integrität. Beurteilung der funktionalen Sicherheit.....	96
Tabelle C-B.1 – Ausführliche Eigenschaften. Entwurfs- und Programmierrichtlinien.....	98
Tabelle C-B.2 – Ausführliche Eigenschaften. Dynamische Analyse und Test.	101
Tabelle C-B.3 – Ausführliche Eigenschaften. Funktionstest und Black-Box-Test	102
Tabelle C-B.4 – Ausführliche Eigenschaften. Ausfall-/Versagensanalyse	103
Tabelle C-B.5 – Ausführliche Eigenschaften. Modellierung.	104
Tabelle C-B.6 – Ausführliche Eigenschaften. Leistungstest.....	105
Tabelle C-B.7 – Ausführliche Eigenschaften. Semiformale Methoden.....	106
Tabelle C-B.8 – Eigenschaften der systematischen Integrität. Statische Analyse.	109
Tabelle C-B.9 – Ausführliche Eigenschaften. Modularer Ansatz.....	111
Tabelle E1 –Kategorien der IEC 61508-2 Anforderungen.....	115

Einleitung

Systeme, die aus elektrischen und/oder elektronischen Elementen bestehen, werden seit vielen Jahren verwendet, um Sicherheitsfunktionen in vielen 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 programmierbar elektronischen Elementen bestehen (E/E/PE) 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 anwendungsspezifischen Normen zu erleichtern.

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, die einzelne Teile einer Gesamtheit sind. Da sich diese Internationale Norm mit sicherheitsbezogenen E/E/PE-Systemen beschäftigt, kann sie einen Rahmen bereitstellen, innerhalb dessen sicherheitsbezogene Systeme, basierend auf anderen Technologien, betrachtet werden können.

Es ist berücksichtigt worden, dass eine große Vielfalt von Anwendungen, sicherheitsbezogener E/E/PE-Systeme in vielfältigen Anwendungsbereichen verwendet und diese einen weiten Bereich in Bezug auf Komplexität, Gefährdungs- und Risikopotentiale abdeckt. In jeder speziellen Anwendung hängen die erforderlichen Sicherheitsmaßnahmen von vielen anwendungsspezifischen Faktoren ab. Dadurch, dass diese Internationale Norm allgemein gehalten ist, wird die Formulierung solcher Maßnahmen in zukünftigen und in Revisionen bereits existierender, anwendungsspezifischer Internationaler 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 anwendungsspezifischer Internationaler Normen, die sich mit sicherheitsbezogenen E/E/PE-Systemen befassen. Die Entwicklung anwendungsspezifischer Normen sollte innerhalb des Rahmens dieser Internationalen 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 Sicherheitsanforderungen, 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 zum Sicherheits-Integritätslevel bestimmt werden können;
- führt Sicherheits-Integritätslevels für die Spezifikation der Zielvorgabe der Sicherheitsintegrität der Sicherheitsfunktionen ein, die in dem sicherheitsbezogenen E/E/PE-System implementiert werden;

ANMERKUNG 1 Diese Norm legt weder die Anforderungen zur Sicherheitsintegrität für irgendeine Sicherheitsfunktion fest, noch bestimmt sie, wie der Sicherheits-Integritätslevel festgelegt wird. Stattdessen stellt sie einen risikobasierenden konzeptionellen Rahmen und Beispielfahren 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 bei kontinuierlicher Anforderung betrieben werden, ist die untere Grenze bei einer mittleren Häufigkeit gefahrbringender Ausfälle von $10^{-9} [h^{-1}]$ festgelegt;

ANMERKUNG 2 Ein einzelnes sicherheitsbezogenes E/E/PE-System bedeutet nicht auch notwendigerweise eine einkanalige Architektur.

ANMERKUNG 3 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 Erfahrung und Urteilsvermögen beruhen, das durch praktische Erfahrung in der Industrie gewonnen wurde. 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;
- lässt einen weiten Bereich von Prinzipien, Verfahren und Maßnahmen zu, um funktionale Sicherheit für sicherheitsbezogene E/E/PE-Systeme zu erreichen, verwendet aber nicht das Fail-Safe-Konzept, das genutzt werden kann, wenn das Ausfallverhalten eindeutig definiert und das Niveau der Komplexität verhältnismäßig niedrig ist. Das Fail-Safe-Konzept wurde wegen des weiten Bereiches der Komplexität von sicherheitsbezogenen E/E/PE-Systemen, die im Rahmen der Norm behandelt werden, als ungeeignet betrachtet.

1 Anwendungsbereich

1.1 Dieser Teil der 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 sicherheitsbezogenes 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 bereit, die anwendbar sind, um Werkzeuge zu unterstützen, 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 2 und 6.

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

- 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 in der Software, die zur Vermeidung und Beherrschung von Fehlern und Versagensfällen der erforderlichen systematischen Eignung entsprechend abgestuft sind;
- f) liefert Anforderungen an die Informationen zu Softwareaspekten der Sicherheits-Validierung, 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 der IEC 61508-1 und IEC 61508-2, Anforderungen an Hilfswerkzeuge, wie Werkzeuge zur Entwicklung und zum Entwurf, Übersetzer, Werkzeuge für Test und Debugging und Werkzeuge für das Konfigurationsmanagement;

ANMERKUNG 4 Das Bild 4 zeigt die Beziehung zwischen der IEC 61508-2 und IEC 61508-3.

- j) ist nicht anwendbar für medizinische Einrichtungen in Übereinstimmung mit der IEC 60601 Serie.

1.2 Die Teile 1, 2, 3 und 4 dieser Norm sind Sicherheits-Grundnormen, dieser Status ist aber im Zusammenhang mit einfachen sicherheitsbezogenen E/E/PE-Systemen nicht anwendbar (siehe 3.4.4 von Teil 4). Als Sicherheits-Grundnormen sind sie zur Verwendung durch technische Komitees bei der Erstellung von Normen nach IEC Guide 104 und ISO/IEC Guide 51 vorgesehen. Die Teile 1, 2, 3, und 4 sind ebenfalls zur Verwendung als eigenständige Veröffentlichungen vorgesehen. Die horizontale Sicherheitsfunktion dieser internationalen Norm ist nicht anwendbar auf medizinische Einrichtungen nach der IEC 60601 Serie.

ANMERKUNG Es steht in der Verantwortlichkeit eines Technischen Komitees, zur Vorbereitung und Erstellung eigener Festlegungen soweit wie möglich die Sicherheits-Grundnormen anzuwenden. In diesem Zusammenhang gilt, dass die Anforderungen, Prüfverfahren oder Prüfbedingungen dieser Sicherheits-Grundnorm nur dann anwendbar sind, wenn in den Festlegungen der Technischen Komitees darauf verwiesen wird oder diese eingebunden werden.

1.3 Bild 1 zeigt den gesamten Rahmen für die Teile 1 bis 7 der IEC 61508 und zeigt die Rolle, die die IEC 61508-1 zum Erreichen der funktionalen Sicherheit der sicherheitsbezogenen E/E/PE-Systeme spielt

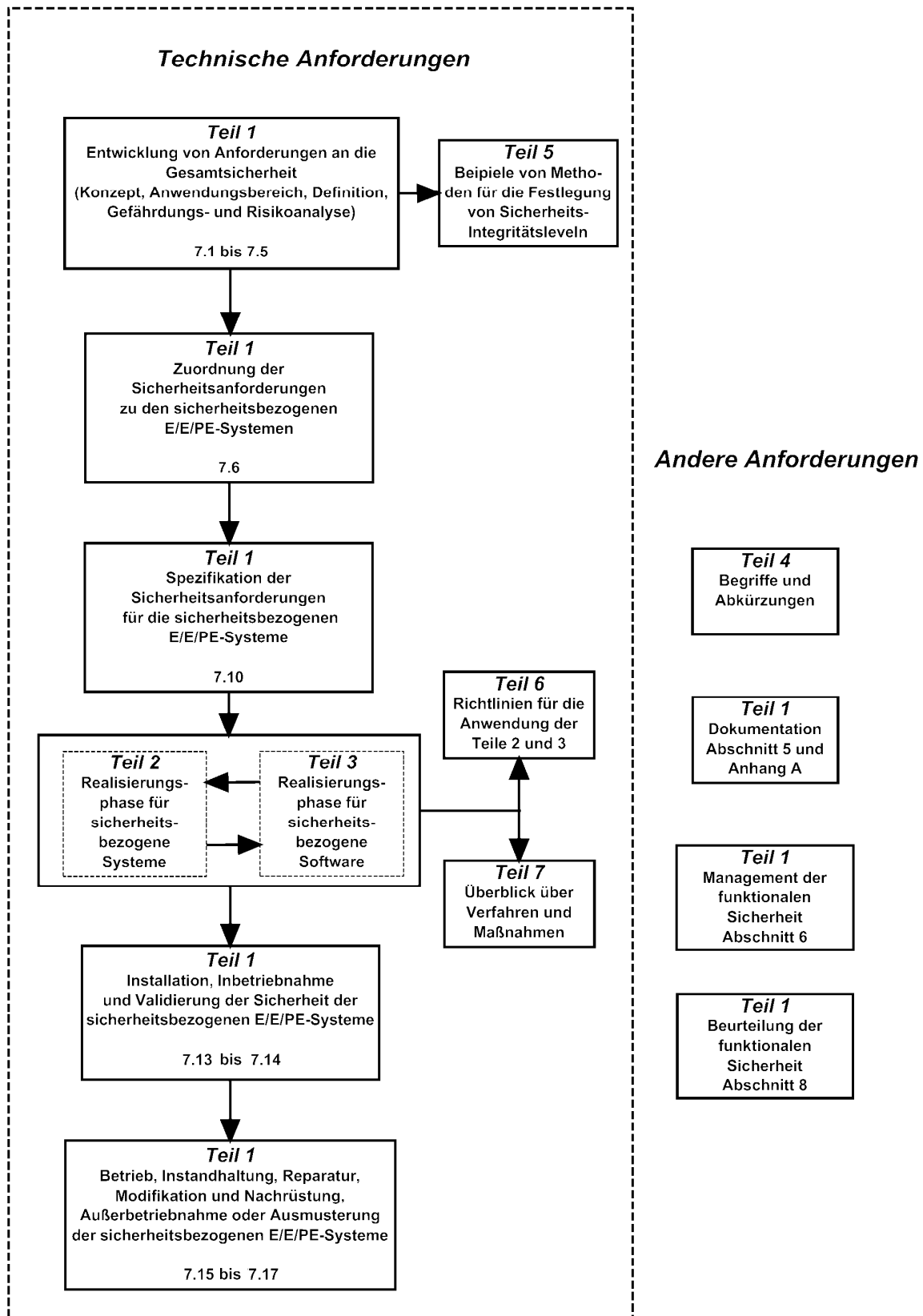


Bild 1 – Gesamtrahmen dieser Norm

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).

ISO/IEC Guide 51:1999, Safetyaspects – Guidelines for their inclusion in standards

IEC Guide 104:1997, Guide to the drafting of safety standards, and the role of Committees with safety pilot functions and safety group functions

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

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

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

ANMERKUNG Für die Anleitung zur Anwendung der normativen Teile 1 bis 4 der IEC 61508, siehe die Bibliographiehinweise [2], [3] und [4].

3 Begriffe und Abkürzungen

Für die Anwendung dieser Norm gelten die in der IEC 61508-4 aufgeführten Begriffe und Abkürzungen.

4 Übereinstimmung mit dieser Norm

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

5 Dokumentation

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

6 Zusätzliche Anforderungen an die Software zum Management der funktionalen Sicherheit

6.1 Ziele

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

6.2 Anforderungen

6.2.1 Die Anforderungen sind in der IEC 61508-1. 6.1 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 sicherheitsbezogenen E/E/PE-System implementierte, Sicherheitsfunktion anzupassen.

6.2.3 Das Konfigurationsmanagement der Software 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 Quelltext, 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 zu erstellen oder zu testen, oder jedwede Arbeit an der Software des sicherheitsbezogenen E/E/PE-Systems durchzuführen;
- d) Verfahren anwenden, um nicht autorisierte Modifikationen zu verhindern, Modifikationsanforderungen zu dokumentieren, Einflüsse einer beabsichtigten Modifikation analysieren zu können und eine Anfrage genehmigen oder zurückweisen zu können, die Einzelheiten der Modifikation und die Bevollmächtigung für die genehmigten Modifikation zu dokumentieren, Mindestanforderungen an die Konfiguration zu bestimmten Punkten in der Softwareentwicklung festzulegen und die teilweisen Integrationstests zu dokumentieren, um die Mindestanforderung zu bestätigen, 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 formale Beurteilung der potentiellen Einflüsse aller möglichen bedeutenden Effekte enthalten, falls ungenügend durchdacht oder ausgeführt. Siehe IEC 61508-7 zu Anleitung, Möglichkeiten und Checklisteneinträgen zur Einflussanalyse.

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

ANMERKUNG 3 Dies kann Überlegungen für spezielle lokale Zielsysteme, sowie allgemeine Systeme einschließen. Abweichend von einer Anwendung, benötigt Software eine sichere Lademethode, z.B. durch Firmware.

- f) die folgenden Informationen dokumentieren, um ein nachfolgendes Audit zu ermöglichen: den Status der Konfiguration, Version, Begründung und Genehmigung der Modifikationen und deren Einzelheiten;
- g) die Version der sicherheitsbezogenen Software formal dokumentieren. Kopien der Software und alle zugehörige Dokumentation und Versionen der Wartungsdaten müssen aufbewahrt werden, um Pflege und Modifikation während der Betriebslebensdauer der Softwareversion zu ermöglichen.

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

7 Anforderungen an den 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 2 bis 5).

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, 6 ausgewählt und spezifiziert werden.

7.1.2.2 Es kann jedes Modell eines Software-Lebenszyklus verwendet werden, das anbietet alle Ziele und Anforderungen dieses Abschnitts erfüllen zu können.

7.1.2.3 Jede Phase des Software-Sicherheitslebenszyklus muss in elementare Tätigkeiten aufgeteilt werden, mit dem Zweck, Eingaben und Ausgaben für jede Phase festzulegen.

7.1.2.4 Falls der Software-Sicherheitslebenszyklus die Anforderungen von Tabelle 1 erfüllt, ist es möglich, das V-Modell (siehe Bild 5) unter Berücksichtigung der Sicherheitsintegrität und Komplexität des Projekts anzupassen.

ANMERKUNG 1 Ein Modell eines Software-Sicherheitslebenszyklusses 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. den Programmiersprachen mit vollem/eingeschränktem Sprachumfang, und Umfang der Datenkonfiguration) die relevant sein können zur individuellen Anpassung des Software-Sicherheitslebenszyklusses.

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 erforderlichen Eigenschaften der systematischen Integrität zu erreichen. Die Auswahl von Verfahren aus diesen Empfehlungen gibt für sich allein keine Gewähr, dass die erforderliche Sicherheitsintegrität erreicht wird.

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

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

7.1.2.9 Wenn sich in irgendeinem Abschnitt 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 formale Beurteilung der potentiellen Einflüsse aller möglichen bedeutenden Effekte enthalten, falls ungenügend durchdacht oder ausgeführt. Siehe IEC 61508-7 zu Anleitung, Möglichkeiten und Checklistenbeiträgen zur Einflussanalyse.

— Entwurf —

E DIN EN 61508-3 (VDE 0803-3):2009-06
FprEN 61508-3:2008

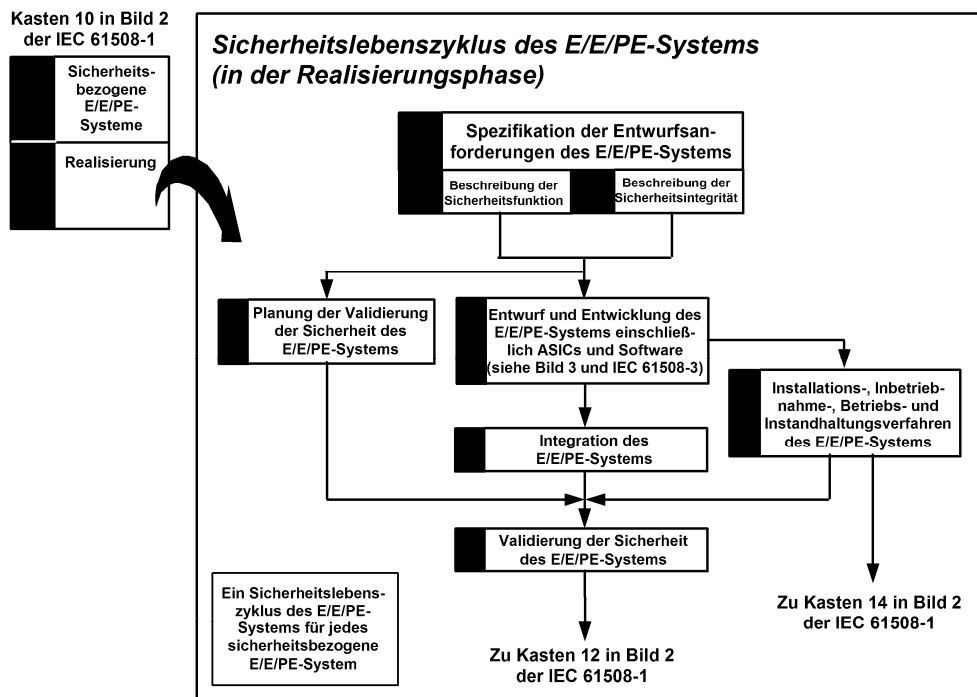


Bild 2 – E/E/PE-System-Sicherheitslebenszyklus (in der Realisierungsphase)

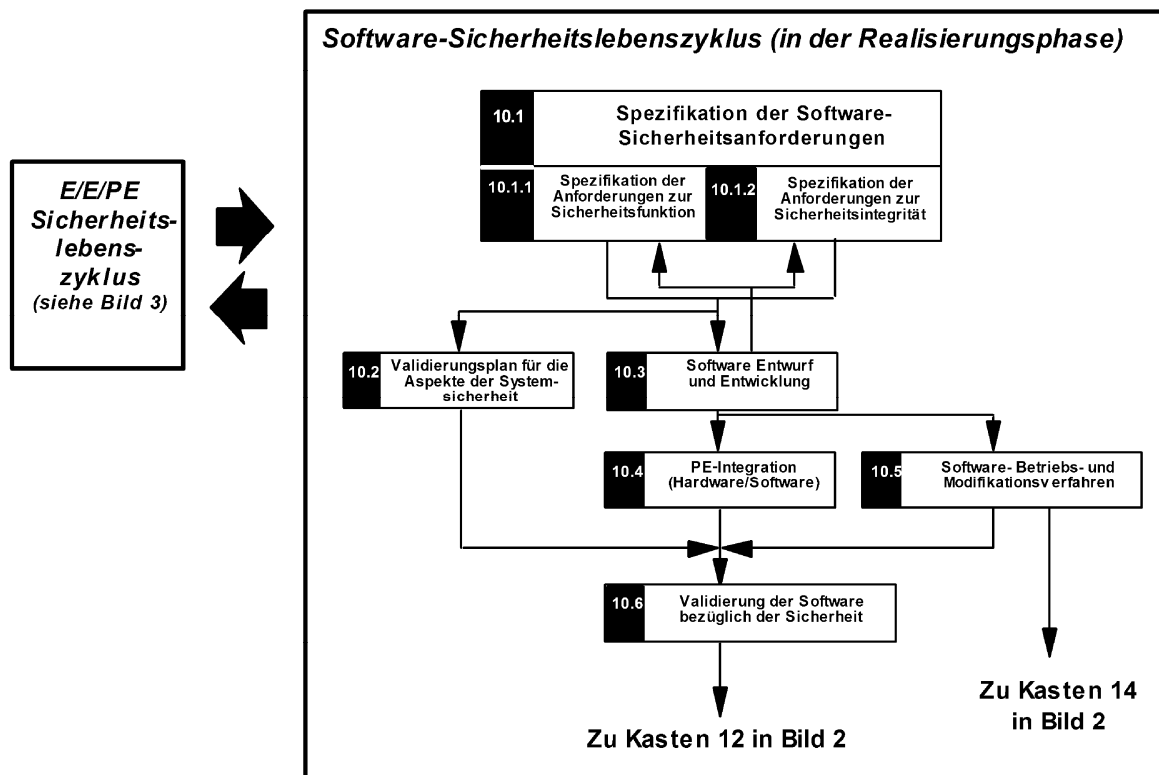


Bild 3 – Software-Sicherheitslebenszyklus (in der Realisierungsphase)

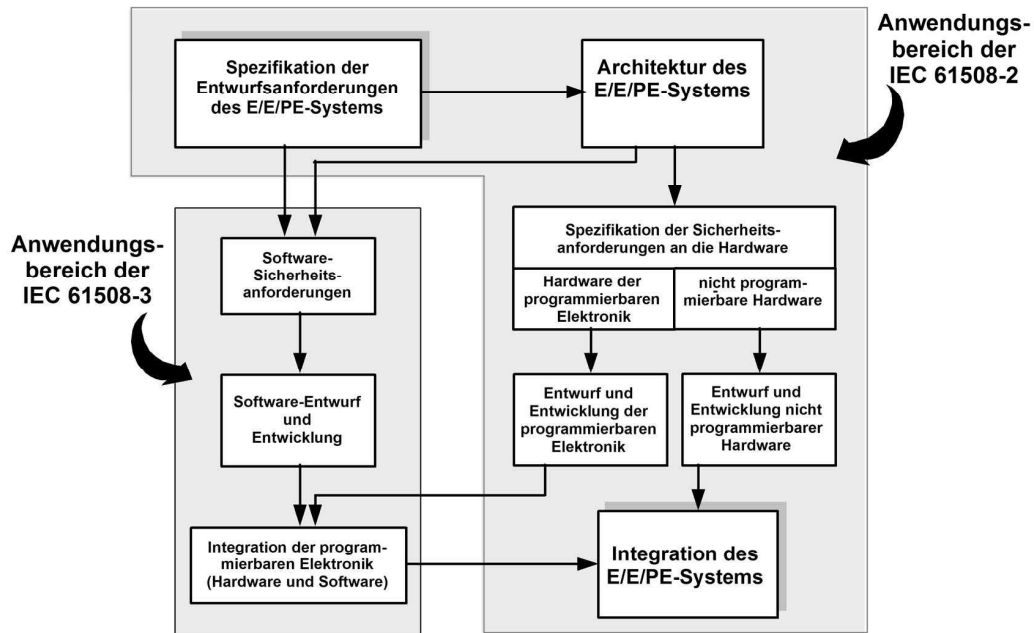


Bild 4 – Beziehung zwischen der IEC 61508-2 und IEC 61508-3 und deren Anwendungsbereiche

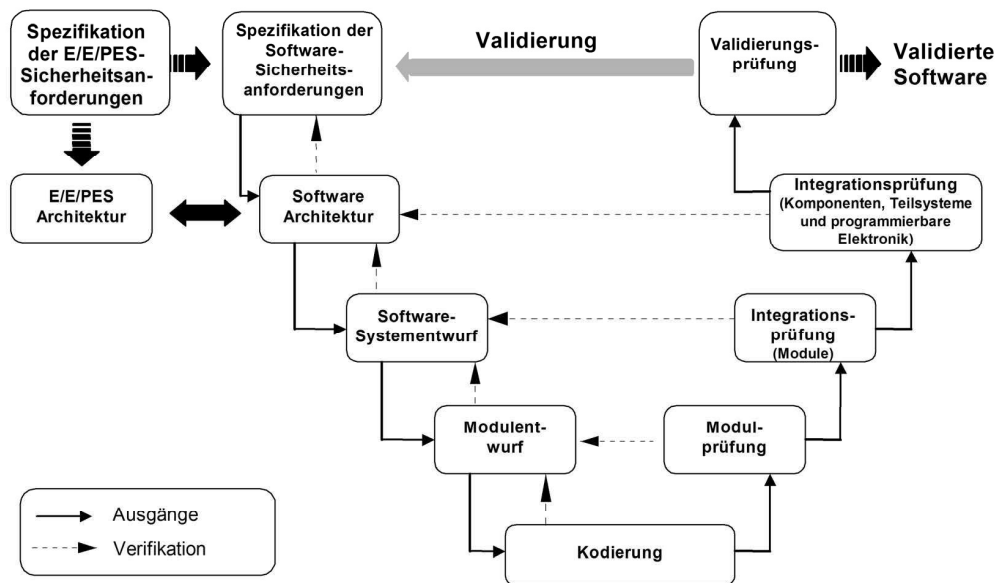


Bild 5 – Systematische Eignung der Software und der Entwicklungslebenszyklus (das V-Model)

Tabelle 1 – Software-Sicherheitslebenszyklus: Überblick

Phase des Sicherheitslebenszyklus		Ziele	Anwendungsbereich	Unterabschnitt für Anforderungen	Eingaben (benötigte Information)	Ausgaben (erzeugte Information)
Kastennummer in Bild 3	Titel					
10.1	Spezifikation der Software-Sicherheitsanforderungen	Spezifizieren der Anforderungen an die sicherheitsbezogene Software mit Begriffen der Anforderungen an die Funktionen der Software-Sicherheit und den Anforderungen an die systematische Eignung der Software; Systematische Eignung; Spezifizieren der Anforderungen an die Funktionen der Software-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) E/E/PE Spezifikation der System-Sicherheitsanforderungen (von IEC 61508-2).	Spezifikation der Software-Sicherheitsanforderungen.
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 Software-Sicherheitsanforderungen	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 beeinflusst werden, einschließlich der Bedeutung der Wechselwirkungen der E/E/PE Hardware mit der Software für die Sicherheit der EUC.	PE System; Software-System.	7.4.3	Spezifikation der Software-Sicherheitsanforderungen; E/E/PE-Systementwurf der Hardware-Architektur (aus IEC 61508-2).	Software-Architektur-Entwurf; Spezifikation des Integrationstests der Software-Architektur; 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)	Ausgaben (erzeugte Information)
Kastennummer in Bild 3	Titel					
10.3	Softwareentwurf und Entwicklung	Hilfswerkzeuge und Programmiersprachen: Auswahl geeigneter Werkzeuge einschließlich Sprachen und Compiler, Schnittstellen zum Laufzeitsystem, Anwenderschnittstellen, Datenformate und Darstellungen, für die erforderliche Sicherheitsintegrität des gesamten Sicherheitslebenszykluses der Software, die die Verifikation, Validierung, Beurteilung und Modifikationen, unterstützen.	PE System; Software-System; Hilfswerkzeuge; Programmierersprache.	7.4.4	Spezifikation der Software-Sicherheitsanforderungen; 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 der sicherheitsbezogenen Software bezüglich des erforderlichen Sicherheits-Integritätslevels erfüllt, die analysierbar und überprüfbar ist und sich sicher modifizieren lässt.	Hauptkomponenten und Teilsysteme des Entwurfs der Software-Architektur.	7.4.5	Software-Architektur Entwurf; Hilfswerkzeuge 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 der sicherheitsbezogenen 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; Hilfswerkzeuge 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 der sicherheitsbezogenen 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; Hilfswerkzeuge und Programmierrichtlinien.	Ausdruck des Quelltextes; Bericht der Code-Überprüfung.

Tabelle 1 (fortgesetzt)

Phase des Sicherheitslebenszyklus		Ziele	Anwendungsbereich	Unterabschnitt für Anforderungen	Eingaben (benötigte Information)	Ausgaben (erzeugte Information)
Kastennummer in Bild 3	Titel					
10.3	Softwareentwurf und Entwicklung	Test der Softwaremodule: Verifizieren, dass die Anforderungen an die sicherheitsbezogene Software (mit 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 durch Daten, die spezifizierten Anforderungen zur systematischen Eignung der Software, erfüllt.	Softwaremodule.	7.4.7	Spezifikation des Tests der Softwaremodule; Ausdruck des Quelltextes; 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 (mit Begriffen der erforderlichen Funktionen der Software bezüglich der Sicherheit und der systematischen Eignung der Software) erreicht wurde. Zeigen, dass alle Softwaremodule, Komponenten und Teilsysteme ihre bestimmungsgemäßen Funktionen und keine anderen ausführen. Soweit angemessen, sicherstellen, dass die Konfiguration des PE Systems durch Daten, die spezifizierten Anforderungen zur systematischen Eignung der Software, erfüllt.	Software-Architektur; Software-System.	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)	Ausgaben (erzeugte Information)
Kastennummer in Bild 3	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	Validierung der Software bezüglich der Sicherheit	Sicherstellen, dass das integrierte System, mit den spezifizierten Anforderungen für die sicherheitsbezogene Software, dem vorgesehenen Sicherheits-Integritätslevel entspricht.	Wie oben	7.7.2	Validierungsplan für die Softwareaspekte der System-sicherheit	Ergebnisse der Validierung der Software bezüglich der Sicherheit; Validierte Software.
–	Software-Modifikation	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; Anfrage zur Modifikation der Software	Ergebnisse der Einflussanalyse der Softwaremodifikation; Logbuch der Softwaremodifikation.
–	Software-Verifikation	Testen und Bewerten der Ausgaben einer gegebenen Phase des Software- Sicherheitslebenszyklus, um Korrektheit und Konsistenz in Bezug zwischen den Ausgängen und Normen als Eingaben dieser Phase 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)	Ausgaben (erzeugte Information)
Kastennummer in Bild 3	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 Software-Sicherheitsanforderungen

ANMERKUNG Diese Phase entspricht Kasten 10.1 in Bild 3.

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, um die folgenden Unterabschnitte zu 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 Software-Sicherheitsanforderungen zu implementieren, müssen folgende Eigenschaften (zur Anleitung siehe Anhang C) berücksichtigt werden:

- Vollständigkeit bezüglich der Sicherheitsbedürfnisse die durch die Software angesprochen werden;
- Korrektheit bezüglich der Sicherheitsbedürfnisse die durch die Software angesprochen werden;
- Freiheit von wesentlichen Spezifikationsfehlern einschließlich der Freiheit von Zweideutigkeiten;
- Verständlichkeit der Sicherheitsanforderungen;
- Freiheit von nachteiligen Beeinflussungen durch die Nichtsicherheitsfunktionen für die Sicherheitsbedürfnisse die durch die Software angesprochen werden;
- Möglichkeit, 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 der Sicherheitsintegrität die den Softwarefunktionen durch den Entwurf des E/E/PE-Systems zugeordnet werden. (Der vollständige Satz der Sicherheitsbedürfnisse des Systems ist größer und beinhaltet auch Sicherheitsfunktionen die nicht von der Software abhängen). Die Vollständigkeit der Spezifikation der Software-Sicherheitsanforderungen hängt wesentlich von der Wirksamkeit früherer Phasen des Systemlebenszyklusses ab.

7.2.2.1 Wenn die Anforderungen an die sicherheitsbezogene Software für das sicherheitsbezogene E/E/PE-System bereits beschrieben worden sind (siehe IEC 61508-2, 7), 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, 7) und jeder Anforderung der Sicherheitsplanung herleiten (siehe 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 Hardware-Architektur des E/E/PE-Systems stattfinden. Deswegen ist eine enge Zusammenarbeit zwischen dem Hardware- und Softwareentwickler notwendig. Siehe Bild 4.

ANMERKUNG 2 Wenn ein Softwareentwurf bereits existierende, wieder verwendbare Software beinhaltet, dann kann es sein, dass diese Software 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 Software-Sicherheitsanforderungen zu erfüllen.

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

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 Ausfällen aufgrund gemeinsamer Ursache durchgeführt werden. Wo mögliche Ausfallmechanismen festgestellt werden, müssen wirksame Schutzmaßnahmen getroffen werden.

ANMERKUNG Siehe Anhang F („Verfahren zum Erreichen der Nicht-Beeinflussung zwischen Softwareelementen auf einem Einzelcomputer“) zu Verfahren zur Erreichung eines Aspektes von 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 zur Sicherheitsintegrität der Hardware (programmierbare Elektronik, Sensoren, Aktoren);
- d) die Anforderungen der systematischen Eignung der Software;
- e) die Leistungsfähigkeit und Reaktionszeit;
- f) die Einrichtungen und Schnittstellen zum Anwender.

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

7.2.2.6 Falls nicht schon ausreichend in den speziellen Sicherheitsanforderungen des sicherheitsbezogenen E/E/PE-Systems definiert, müssen alle relevanten Betriebsarten der EUC, des E/E/PE-Systems und jeder Ausrüstung oder eines 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 Software-Sicherheitsanforderungen muss jede sicherheitsbezogene oder relevante Beziehung 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 Hardware-Architektur erforderlich ist und unter Berücksichtigung einer möglicherweise steigenden Komplexität, muss die Spezifikation der Software-Sicherheitsanforderung 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) während des Betriebs der EUC, die Testbarkeit der Sicherheitsfunktionen ermöglichen;
- e) die Softwarefunktionen zur Durchführung von Selbsttests 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 es nötig machen die Architektur noch einmal zu überdenken.

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

ANMERKUNG Für Anforderungen zur Nichtbeeinflussung zwischen Sicherheitsfunktionen und Nichtsicherheitsfunktion siehe 7.4.2.8 und 7.4.2.9.

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

- a) die Anforderungen an die folgenden Software-Sicherheitsfunktionen:
 - 1) Funktionen, die es der EUC ermöglichen, einen sicheren Zustand zu erreichen oder diesen aufrechtzuerhalten;
 - 2) Funktionen bezüglich der Detektion, Anzeige und Handhabung von Fehlern in der Hardware der programmierbaren Elektronik;
 - 3) Funktionen bezüglich der Detektion, Anzeige und Handhabung von Fehlern der Sensoren und Aktoren;
 - 4) Funktionen bezüglich der Detektion, Anzeige und Handhabung von Fehlern in der Software selbst (Selbstüberwachung der Software);
 - 5) Funktionen bezüglich der periodischen Online-Tests der Sicherheitsfunktionen;
 - 6) Funktionen bezüglich der periodischen Offline-Tests der Sicherheitsfunktionen;
 - 7) Funktionen, die es erlauben, Modifikationen am PE-System sicher durchführen zu können;
 - 8) Schnittstellen zu nicht sicherheitsbezogenen Funktionen;
 - 9) Leistungsfähigkeit und Reaktionszeit;
 - 10) Schnittstellen zwischen der Software und dem PE-System.

ANMERKUNG 1 Die Schnittstellen beinhalten beides, Online- und Offline-Programmierungsmöglichkeiten.

- 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 obigen Funktionen unter a);

ANMERKUNG 2 Siehe Anhang A der IEC 61508-5 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 den Begriffen 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 Quelltext ist bereits vorhanden und das primäre Ziel der Entwicklung ist es Sicherheit zu geben, dass die Konfigurationsdaten das erforderliche Verhalten der Anwendung korrekt beschreiben. Es können komplexe Abhängigkeiten zwischen Dateneinheiten bestehen und die Gültigkeit der Daten kann über die Zeit wechseln.

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 zur IEC 61508-2, 7.4.11, betrachtet werden:

- a) die Notwendigkeit zur Konsistenz in Bezug auf die Definition der Daten;
- b) ungültige, außerhalb des zulässigen Bereichs liegende oder Werte zur falschen Zeit;
- c) die Reaktionszeit und Durchsatz, einschließlich der Bedingungen zur Maximalbelastung;
- d) die Ausführungszeit und Blockierungen;
- e) einen Überlauf und Unterschreiten der Kapazität des Datenspeichers.

7.2.2.13 Betriebsparameter müssen geschützt werden gegen:

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

ANMERKUNG 1 Der Schutz gegen unbefugte Änderungen sollte berücksichtigt werden, durch sowohl Software-mechanismen 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 erfolgen sollen.

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

- Das Informationssystem für den Bediener sollte eine bildhafte Darstellung und eine Ausdrucksweise verwenden, die dem Bediener 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 ausführbar und/oder wenn die Konsequenzen der Beeinflussungen durch den Bediener nicht sofort ersichtlich sind, sollte die angezeigte Information für jeden Zustand automatisch die Handlungsreihenfolge beinhalten, mit dem durch die Sequenz erreichbaren Zustände, 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 3.

ANMERKUNG 2 Software kann üblicherweise nicht separat ohne seine 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 durchgeführt werden, um die Schritte zu spezifizieren, sowohl die des Verfahrens als auch die technischen, die verwendet werden, um darzulegen, dass die Software ihre Sicherheitsanforderungen erfüllt.

7.3.2.2 Der Validierungsplan für die Softwareaspekte der Systemsicherheit unter den Aspekten 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.
- d) eine Kennzeichnung der sicherheitsbezogenen Software, die vor der Auslieferung 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 nachzuweisen, dass jede Sicherheitsfunktion mit den spezifizierten Anforderungen bezüglich den Sicherheitsfunktionen und den Anforderungen der systematischen Eignung der Software übereinstimmt;
- g) die erforderlichen Umgebungsbedingungen, bei denen die Validierungen durchzuführen sind (z. B. kann dies für 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 der IEC 61508-1, 7.8.

7.3.2.3 Die Validierung muss eine nachvollziehbare 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 Software unter den Aspekten der Systemsicherheit, muss das Ziel und Inhalt des Validierungsplans für die Softwareaspekte der Systemsicherheit, mit dem Gutachter oder mit einer Gruppe die den Gutachter repräsentiert, wenn durch den Sicherheits-Integritätslevel gefordert, überprüft werden (siehe IEC 61508-1, 8). Dieses Verfahren muss auch eine Aussage zur Anwesenheit der Gutachter während des Tests machen.

7.3.2.5 Die Kriterien, für bestanden und nicht bestanden, der Softwarevalidierung 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 Werttoleranzen.

7.4 Softwareentwurf und Entwicklung

ANMERKUNG Diese Phase entspricht dem Kasten 10.3 in Bild 3:

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.

7.4.1.3 Das dritte Ziel der Anforderungen dieses Unterabschnittes ist es, einen geeigneten Satz von Werkzeugen, mit Sprachen und Compilern, Laufzeitsystem-Schnittstellen, Anwenderschnittstellen, Datenformaten und Darstellungen für den geforderten Sicherheits-Integritätslevel auszuwählen, der den gesamten Sicherheitslebenszyklus der Software einschließlich 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 der sicherheitsbezogenen Software bezüglich des geforderten Sicherheits-Integritätslevels erfüllt, analysierbar und verifizierbar ist und die Möglichkeit bietet, sicher modifiziert werden zu können.

7.4.1.5 Das fünfte Ziel der Anforderungen dieses Unterabschnittes ist es, zu verifizieren, dass die Anforderungen an die sicherheitsbezogene Software (betrifft der geforderten Sicherheitsfunktionen und systematischen Eignung) erreicht worden sind.

7.4.1.6 Das sechste Ziel der Anforderungen dieses Unterabschnittes ist es sicherzustellen, soweit wie angemessen, dass 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 bei dem Lieferanten einer sicherheitsbezogenen programmierbaren Einrichtung (z. B. Lieferanten einer SPS) allein, dem Anwender dieser Einrichtung (z. B. Entwickler der Anwendungssoftware) allein oder bei beiden liegen. Die Zuordnung der Verantwortlichkeit muss während der Sicherheitsplanung (siehe Abschnitt 6) festgelegt werden.

ANMERKUNG Siehe 7.4.3 zu den entsprechenden Aspekten der System- und Softwarearchitektur für die Entscheidung der Aufteilung der Verantwortlichkeiten.

7.4.2.2 In Übereinstimmung mit dem geforderten Sicherheits-Integritätslevel 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 der Informationen;
 - 4) zeitlichen Einschränkungen;
 - 5) Gleichzeitigkeit und synchronisierter Zugriff auf gemeinsame Ressourcen;
 - 6) Datenstrukturen und deren Eigenschaften;
 - 7) Voraussetzungen für den Entwurf und ihre Abhängigkeiten;
 - 8) Behandlung von Ausnahmen;
 - 9) Voraussetzung für den Entwurf (Vorbedingungen, Nachbedingungen, Unveränderlichkeiten);
 - 10 Kommentare;
- c) der Fähigkeit verschiedene Sichtweisen des Entwurfs zu ermöglichen einschließlich Ansichten zur 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 für sichere 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 Softwaredmodifikationen erleichtern. Diese Eigenschaften beinhalten Modularität, Information Hiding und Kapselungen (en: Encapsulation).

7.4.2.5 Die Darstellung des Entwurfs muss in einer eindeutig definierten Aufzeichnungsart erfolgen oder auf eingeschränkte 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 Aufdeckung eines Ausfalls müssen geeignete Maßnahmen ergriffen werden.

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

7.4.2.9 Wenn die Software Sicherheitsfunktionen verschiedener Sicherheits-Integritätslevel beinhaltet, 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 erreicht wird durch sowohl im räumlichen und zeitlichen Bereich oder (2) jede Verletzung der Unabhängigkeit beherrscht wird. Die Rechtfertigung für diese Unabhängigkeit muss dokumentiert werden.

ANMERKUNG Siehe Anhang F („Verfahren zum Erreichen der Nicht-Beeinflussung zwischen Softwareelementen auf einem Einzelcomputer“) zu Verfahren zur Erreichung eines Aspektes von 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 das Element 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 Reihe von Softwareelementen mit bekannter systematischer Eignung implementiert wird, müssen die Anforderungen an die systematische Eignung aus der IEC 61508-2, 7.4.3 für die Kombination der Elemente angewendet werden.

ANMERKUNG Konsequenterweise ist zu unterscheiden zwischen (1) der durchgehenden Sicherheitsfunktion die durch ein oder mehrere Elemente bereitgestellt wird und (2) der Sicherheitsfunktion die von jedem einzelnen Element bereitgestellt wird. Wo eine Kombination von zwei Elementen eine höhere systematische Eignung erreicht, muss jedes der verbundenen Elemente in der Lage sein das gefährliche Ereignis zu verhindern oder zu mildern. Für die verbundenen Elemente besteht aber nicht die Anforderung, dass jedes Element identische Sicherheitsfunktionen hat. Weiter ist nicht gefordert, dass jedes der verbundenen Elemente unabhängig in der Lage ist die gesamte, geforderte Sicherheitsfunktion der Kombination bereit zu stellen. Zur Verdeutlichung: Die elektronische Geschwindigkeitsregelung einer Maschine mit der durchgehenden Sicherheitsfunktion „Vermeidung eines unerwarteten Anlaufs“, wird durch zwei Prozessoren realisiert. Die Sicherheitsfunktion des Elements der primären Steuerung ist das ideale Anforderungs-/Reaktions-Verhalten der Geschwindigkeitsregelung. Die Sicherheitsfunktion des Elements der sekundären Steuerung ist die Bereitstellung einer Sicherheitsüberwachung mit einer Stillsetzung im Notfall, wenn notwendig. Die Kombination der beiden Prozessoren gibt ein höheres Vertrauen das die durchgehende Sicherheitsfunktion „Vermeidung eines unerwarteten Anlaufs“, erreicht wird.

7.4.2.12 Wenn ein bereits existierendes Softwareelement verwendet wird um eine Sicherheitsfunktion zu realisieren, muss die Eignung des Softwareelements zur Erfüllung der Spezifikation der Software-Sicherheitsanforderungen begründet werden. Diese Begründung muss basieren auf:

- a) der Erfüllung der Anforderungen eines Konformitätspfads mit der IEC 61508-2, 7.4.2.2 c) und
- b) einem Sicherheitshandbuch (siehe IEC 61508-2 Anhang D und IEC 61508-3 Anhang D) welches ausreichend, präzise und vollständig das Element beschreibt, um die Beurteilung der Integrität einer speziellen Sicherheitsfunktion, die vollständig oder teilweise zum Element gehört, zu ermöglichen.

ANMERKUNG 1 Siehe IEC 61508-4, 3.2.8. Die bereits existierende Software kann ein kommerzielles Produkt sein oder 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 2 Der erforderliche Nachweis nach diesem Abschnitt kann durch die vom Lieferanten eigene Dokumentation und Aufzeichnungen des Herstellungsprozesses oder durch zusätzliche qualifizierte Aktivitäten durch den Entwickler des sicherheitsbezogenen Systems oder durch Dritte, erfolgen. 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.

ANMERKUNG 3 Die Rechtfertigung kann während der Softwareplanung entwickelt werden (siehe Abschnitt 6).

ANMERKUNG 4 Anforderungen an bereits existierende Softwareelemente beziehen sich auf Laufzeitbibliotheken oder auf einen Interpreter.

7.4.2.13 Ein bereits existierendes Softwareelement dessen Eignung durch IEC 61508-2, 7.4.2.2 c) Weg 3_s begründet wurde, muss folgende Anforderungen 7.4.2.13.1 bis 7.4.2.13.9, erfüllen.

7.4.2.13.1 Die Spezifikation der Software-Sicherheitsanforderungen für das Element in seiner neuen Anwendung muss in der gleichen Sorgfalt dokumentiert werden, wie es durch die IEC 61508-3 für jedes sicherheitsbezogene Element mit der gleichen systematischen Eignung, gefordert wird. Die Spezifikation der Software-Sicherheitsanforderungen umfasst das funktionale als auch Sicherheitsverhalten soweit zutreffend für das Element in seiner neuen Anwendung und wie in 7.2 spezifiziert. Siehe Tabelle A.1.

7.4.2.13.2 Die Rechtfertigung zur Verwendung eines Softwareelements muss den Nachweis liefern, dass die wünschenswerten, in der IEC 61508-3 spezifizierten Sicherheitseigenschaften der verwiesenen Abschnitte (d. h. 7.2.2, 7.4.3, 7.4.4, 7.4.5, 7.4.6, 4.4.7, 7.5.2, 7.7.2, 7.8.2, 7.9.2, 8) berücksichtigt wurden mit der Anleitung von Anhang C.

7.4.2.13.3 Der Entwurf des Elements muss in einem derartigen Umfang so 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 Anhang A Tabellen A.2 und A.4.

7.4.2.13.4 Der in 7.4.2.13.1 und 7.4.2.13.2 geforderte Nachweis muss die Integration der Software in die Hardware mit beinhalten. Siehe 7.5 und Anhang A Tabelle A.6.

7.4.2.13.5 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 Anhang A Tabellen A.5 bis A.7 und A.9 und die zugehörigen Tabellen in Anhang B.

ANMERKUNG Eine positive Betriebserfahrung kann verwendet werden, um Anforderungen an Black-Box-Tests und probabilistische Tests zu erfüllen. (siehe IEC 61508-3 Tabellen A.7 und B.3).

7.4.2.13.6 Wo das Softwareelement Funktionen, die nicht verwendet werden oder im sicherheitsbezogenen System nicht erforderlich sind, bereitstellt, muss der Nachweis erbracht werden, dass die nicht verwendeten und nicht benötigten Funktionen das E/E/PE-System nicht daran hindern werden seine Sicherheitsanforderungen zu erfüllen.

ANMERKUNG Möglichkeiten, diese Anforderung zu erfüllen, beinhalten:

- Entfernung der Funktionen aus dem Aufbau;
- Deaktivierung der Funktionen;
- geeignete Systemarchitektur (z. B. Aufteilung, Schutzmechanismen, Diversität, Glaubwürdigkeit der Prüfung von Ausgaben);
- umfangreiche Tests.

7.4.2.13.7 Es muss einen Nachweis geben dass alle anzunehmenden Ausfallmechanismen des Softwareelements identifiziert wurden und dass alle angemessenen mindernden Maßnahmen implementiert worden sind.

ANMERKUNG Angemessene, Schadensbegrenzungsmaßnahmen beinhalten:

- geeignete Systemarchitektur (z. B. Aufteilung, Schutzmechanismen, Diversität, Glaubwürdigkeit der Prüfung von Ausgaben);
- Behandlung von Ausnahmen.

7.4.2.13.8 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.

7.4.2.13.9 Die Begründung für die Verwendung des Elements darf nur für die Anwendungen gültig sein, für die Annahmen im Sicherheitshandbuch für konforme Objekte gemacht worden sind (Siehe IEC 61508-2 Anhang D und IEC 61508-3 Anhang D).

7.4.2.14 Dieser Unterabschnitt 7.4 muss, soweit geeignet, auf Daten und Sprachen die Daten erzeugen, angewendet werden.

ANMERKUNG Siehe Anhang G zur Anleitung von datengesteuerten Systemen:

- a) wo ein PE-System aus bereits existierenden datenkonfigurierten Funktionalitäten besteht, um die speziellen Anforderungen der Anwendung zu erfüllen, muss der Entwurf dem Grad der Konfigurierbarkeit der Anwendungssoftware der Anwendung, bereits ausgelieferten Funktionalitäten und der Komplexität des sicherheitsbezogenen PE-Systems, entsprechen;
- b) wo die sicherheitsbezogene Funktionalität eines PE-Systems wesentlich durch Konfigurationsdaten bestimmt wird, müssen angemessene Verfahren und Maßnahmen verwendet werden, um die Fehlereinpflanzung während des Entwurfs, der Produktion, dem Laden und Modifikation der Konfigurationsdaten zu vermeiden. Dies, um sicherzustellen, dass die Konfigurationsdaten die Logik der Anwendung ordnungsgemäß darstellen;
- c) Die Spezifikation von Datenstrukturen muss
 - 1) konsistent zu den funktionalen Anforderungen des Systems einschließlich der Anwendungsdaten;
 - 2) vollständig;
 - 3) widerspruchsfrei;
 - 4) geschützt gegen Veränderung und Verfälschung

sein.

7.4.3 Anforderungen an den Softwarearchitektur Entwurf

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 wird. Sie definiert auch das Gesamtverhalten der Software und wie Softwareelemente verbunden sind und deren Wechselwirkungen. Ein wesentlicher Teil der Softwarearchitektur kann automatisch erzeugt worden sein (in Abhängigkeit der Einschränkungen der verfügbaren Elemente). Der Prozess der Bestimmung der Elemente und Teilsysteme kann darin bestehen, Einschränkungen zu beschreiben. Die Übereinstimmung mit den Einschränkungen kann dann durch automatische Werkzeuge oder manuelle Tätigkeiten erreicht werden. Beispiele wesentlicher Softwareelemente sind Betriebssysteme, Datenbanken, Untersysteme 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 Architektur durch den Lieferanten als serienmäßige Eigenschaft der SPS 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 sind weiterhin anzuwenden. Die Anforderung, die Architektur 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 IEC 61508 numerische Ausfallgrenzwerte für Sicherheitsfunktionen festlegt die durch ein sicherheitsbezogenes E/E/PE-System ausgeführt werden, ist die systematische Integrität üblicherweise nicht quantifizierbar (siehe IEC 61508-4, 3.5.6) und die Sicherheitsintegrität der Software (siehe IEC 61508-4, 3.5.9 und 3.5.10) wird definiert als eine systematische Fähigkeit auf einer Vertrauensskala von 1-4. Die IEC 61508-3 erkennt an, dass es ein sicheres Softwareversagen geben kann, so dass die reine Ausfallrate nicht das signifikante Maß für Sicherheit ist. Die System/Softwarearchitektur ist so auszuführen, dass unsichere Ausfälle eines Elements durch Randbedingungen auf Grund der Architektur begrenzt werden und dass die Entwicklungsmethoden diese Randbedingungen berücksichtigen sollten. Die IEC 61508-3 wendet Entwicklungs- und Validierungsverfahren an mit einer konsequenten Strenge qualitativ passend zur systematischen Eignung.

ANMERKUNG 6 Zur Auswahl angemessener Verfahren und Maßnahmen, (siehe Anhänge A und B). Um die Anforderungen dieses Abschnitts zu implementieren, sollten folgende Eigenschaften an den Softwarearchitektur Entwurf (zur Anleitung siehe Anhang C) berücksichtigt werden:

- Vollständigkeit in Bezug auf die Spezifikation der Software-Sicherheitsanforderungen;
- Korrektheit in Bezug auf die Spezifikation der Software-Sicherheitsanforderungen;
- Freiheit von wesentlichen Entwurfsfehlern;
- Einfachheit und Verständlichkeit;
- Vorhersagbarkeit des Verhaltens;
- Nachprüfbarer und testbarer Entwurf;
- Fehlertoleranz;
- Abwehr 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 7.4.3 beim mehreren Beteiligten liegen. Die Zuordnung der Verantwortlichkeit muss während der Sicherheitsplanung (siehe IEC 61508-1, 6) dokumentiert werden.

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

- a) einen zusammenhängenden Satz von Verfahren und Maßnahmen auswählen und rechtfertigen, die notwendig sind, während der Phasen des Software-Sicherheitslebenszyklus die Spezifikation der Software-Sicherheitsanforderungen 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 diese 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-Ausgangsdaten der Anlage, Kommunikationsdaten, Daten der Bedienerchnittstelle, Daten für die Wartung und interne Daten der Datenbank umfassen;
- f) angemessene Integrationstests der Softwarearchitektur beschreiben, um sicherzustellen, dass die Softwarearchitektur die Spezifikation der Software-Sicherheitsanforderungen gemäß dem Sicherheits-Integritätslevel erfüllt.

7.4.3.3 Jede notwendige Änderung an der Spezifikation der Sicherheitsanforderungen 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, offene 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 Hilfswerkzeuge einschließlich Programmiersprachen

ANMERKUNG Zur Auswahl geeigneter Verfahren und Maßnahmen (siehe Anhänge A und B). Zur Implementierung der Anforderungen dieses Abschnitts sollten die folgenden Eigenschaften berücksichtigt werden (zur Anleitung siehe Anhang C):

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

7.4.4.1 Online-Software-Hilfswerkzeuge müssen wie Software behandelt werden die zum sicherheits-bezogenen System, oder zu einem anderen System, gehören.

ANMERKUNG Für Beispiele zu Online- und Offline-Hilfswerkzeugen siehe IEC 61508-4, 3.2.10 und 3.2.11.

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

ANMERKUNG 1 Zu den Anforderungen an die Entwicklung des Software-Lebenszyklus, siehe 7.1.2.

ANMERKUNG 2 Es sollten angemessene Offline-Hilfswerkzeuge verwendet werden, um die Entwicklung von Software zu unterstützen, mit dem Ziel die Integrität bei der Entwicklung der Software zu erhöhen, durch Verringerung der Wahrscheinlichkeit Fehler einzubringen oder nicht zu bemerken. Beispiele relevanter Werkzeuge im Zyklus der Softwareentwicklungs-Phasen beinhalten:

- 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, Ladeprogramme und Code generierende Werkzeuge;
- b) Verifikations- und Validierungswerkzeuge wie statische Codeanalysatoren, Testabdeckungskontrollen, Assistenten zum Nachweis von Theoremen und Simulatoren;
- c) Dynamische Werkzeuge verwendet um die Software unter Betriebsbedingungen zu überwachen;
- 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 die notwendig sind Parameter zu definieren um Systemfunktionen zu instantieren z.B. Funktionsparameter, Anzeigebereiche, Alarm- und Schaltschwellen, Ausgangszustände die im Fall eines Ausfalls übernommen werden, räumliche Aufteilung.

ANMERKUNG 3 Offline-Hilfswerkzeuge sollten für eine Integration ausgewählt werden. In diesem Zusammenhang sind Werkzeuge integriert, wenn diese so zusammenarbeiten, dass die Ausgaben eines Werkzeugs geeignete Inhalte und Formate als automatische Eingabe für ein nachgeordnetes Werkzeug liefern. Dies minimiert die Möglichkeit menschlichen Versagens bei der Nachbearbeitung von Zwischenergebnissen.

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

ANMERKUNG 5 Die Verfügbarkeit geeigneter Werkzeuge, die die notwendige Leistung über die gesamte Lebensdauer des sicherheitsbezogenen E/E/PE-Systems liefern, sollte berücksichtigt werden.

ANMERKUNG 6 Die Fähigkeiten der Anwender, für die ausgewählten Werkzeuge, sollten berücksichtigt werden. Siehe IEC 61508-6 zu den Anforderungen an Fähigkeiten.

7.4.4.3 Die Auswahl der Offline-Hilfswerkzeuge muss begründet werden.

7.4.4.4 Jedes Offline-Hilfswerkzeug der Klassen T2 und T3 muss ein Spezifikations- oder Produkthandbuch haben welches eindeutig das Verhalten des Werkzeugs und die Einschränkungen seiner Verwendung beschreibt.

ANMERKUNG 1 Zu den Kategorien der Software-Offline-Hilfswerkzeuge siehe IEC 61508-4, 3.2.11.

ANMERKUNG 2 Dieses „Spezifikations- oder Produkthandbuch“ ist kein Sicherheitshandbuch für konforme Objekte (siehe 61508-2 Anhang D und 61508-3 Anhang D) für das Werkzeug selbst. Das Konzept des Sicherheitshandbuchs für konforme Objekte bezieht sich nur 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 aus dem „Spezifikations- oder Produkthandbuch“ des Werkzeugs in das Sicherheitshandbuch für konforme Objekte übernommen 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-Hilfswerkzeuge 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 potentieller Ausfälle 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 Handbuch 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 einer Validation des Werkzeugs wie in 7.4.4.7 beschrieben.

ANMERKUNG 1 Eine Versionshistorie und die Aufzeichnung von Fehlern/Unklarheiten in Verbindung mit den Umgebungsbedingungen der Anwendung, kann Vertrauen zur Reife des Werkzeugs geben.

ANMERKUNG 2 Der für T3 gelistete Nachweis kann auch für T2 Werkzeuge zum Nachweis der Korrektheit ihrer Ergebnisse verwendet werden.

7.4.4.7 Die Ergebnisse der Validierung des Werkzeugs müssen dokumentiert werden durch Abdeckung folgender Ergebnisse:

- a) einer chronologische Aufzeichnung der Validierungsaktivitäten;
- b) der Version des verwendeten Produkthandbuchs zum Werkzeug;
- c) den validierten Funktionen des Werkzeugs;
- d) der verwendeten Werkzeuge und Ausrüstung;
- e) der Ergebnisse der Validierungsaktivitäten; die dokumentierten Ergebnisse der Validierung muss 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 Beleg der Übereinstimmung nach 7.4.4.6 nicht verfügbar ist, müssen effektive Maßnahmen getroffen werden um Ausfälle des ausführbaren sicherheitsbezogenen Systems zu beherrschen, die dem Werkzeug zuzuschreiben sind.

ANMERKUNG Ein Beispiel einer Maßnahme könnte sein die Generierung divers-redundanten Codes der die Detektion und Beherrschung von Ausfällen ermöglicht, die durch einen Übersetzer eingebracht wurden.

7.4.4.9 Die Verträglichkeit der Werkzeuge einer eingefügten Gruppe von Werkzeugen muss bestätigt werden.

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, eine Beurteilung nach internationalen oder nationalen Normen;
- b) nur definierte Sprachfunktionen verwenden;
- c) den Eigenschaften 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 aus einer Klasse von Darstellungen von 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 Ebene. 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. Für einen späteren Fall sollten alle notwendigen Information 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 reduziert werden auf die Kontrolle einer allgemeinen Eignung des Werkzeugs für das Projekt und Übereinstimmung mit dem „Spezifikations- oder Produkthandbuch“ (d. h. korrekte Anwendung des Werkzeugs). Die korrekte Anwendung kann zusätzliche Verifikationen innerhalb des Projektes beinhalten.

ANMERKUNG 3 Eine Folge von Validierungen 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 jeder zusätzlichen Maßnahme die irgendeinen bekannten Mangel der Sprache betrifft, begründet werden.

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

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

7.4.4.13 Eine Programmierrichtlinie für eine Programmiersprache muss gute Programmiertechniken beschreiben, unsichere Sprachenmerkmale verbieten (z. B. undefinierte Sprachenmerkmale, unstrukturierte Entwürfe usw.) die Codeverständlichkeit unterstützen, Verifikation und Tests erleichtern und die Verfahren für die Dokumentation des Quelltextes beschreiben. Wo durchführbar, müssen mindestens folgende Informationen im Quelltext 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 Codegeneratoren oder ähnliche automatische Übersetzungen Einsatz finden, muss die Eignung des automatischen Übersetzers für die Entwicklung des sicherheitsbezogenen Systems an dem Punkt des Lebenszyklusses der Entwicklung beurteilt werden, an dem die Hilfswerkzeuge ausgewählt werden.

7.4.4.15 Wo Offline-Hilfswerkzeuge der Klassen T2 und T3 Eigenschaften in der Basiskonfiguration generieren, muss das Konfigurationsmanagement sicherstellen, dass diese 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 Eigenschaften 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 Rekonstruktion des Grundzustandes zu ermöglichen.

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

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

ANMERKUNG Die Hardware des sicherheitsbezogenen Systems kann den Werkzeugen ebenfalls Randbedingungen zur Kompatibilität auferlegen, z.B. Bedürfnisse zur Emulierung des Prozessors für ein exaktes Modell der Prozessorelektronik.

7.4.4.18 Jede neue Version eines Offline-Hilfswerkzeugs muss begründet werden. Diese Begründung kann auf dem Vertrauen zu einer Vorgängerversion basieren, wenn ausreichende Nachweise vorliegen, das

- 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 und unbekannten Fehler enthält.

ANMERKUNG Vertrauen das die neue Version wahrscheinlich keine wesentlichen neuen und unbekannten Fehler enthält, kann auf (1) einer glaubwürdigen Identifikation der durchgeführten Änderungen und (2) einer Analyse der durchgeführten Verifikations- und Validierungs-Aktivitäten, basieren.

7.4.4.19 In Abhängigkeit von der Natur der Softwareentwicklung, kann die Verantwortlichkeit für die Übereinstimmung mit 7.4.3 auf mehreren Beteiligten ruhen. Die Aufteilung der Verantwortlichkeiten muss während der Sicherheitsplanung dokumentiert werden (siehe IEC 61508-1, 6).

7.4.5 Anforderungen an den detaillierten Entwurf und die Entwicklung: Software-Systementwurf

ANMERKUNG 1 Als detaillierter Entwurf wird hier der Software-Systementwurf verstanden; die Zuordnung der Hauptelemente der Architektur in ein System von Softwaremodulen, das Entwerfen von einzelnen Softwaremodulen und die Codierung. In kleinen Anwendungen kann der Software-Systementwurf 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 Zusammenhängen wie die 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) sicherheitsrelevante Teile abteilt, mit Bereichstest 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 implementieren, sollten folgende Eigenschaften (siehe Anhang C zur Anleitung) zu Entwurf und Entwicklung berücksichtigt werden:

- Vollständigkeit in Bezug auf die Spezifikation der Software-Sicherheitsanforderungen;
- Korrektheit in Bezug auf die Spezifikation der Software-Sicherheitsanforderungen;
- Freiheit von wesentlichen Entwurfsfehlern;
- Einfachheit und Verständlichkeit;
- Vorhersagbarkeit des Verhaltens;
- Nachprüfbarer und testbarer Entwurf;
- Fehlertoleranz/Fehlererkennung;
- Freiheit von Ausfällen infolge gemeinsamer Ursache.

7.4.5.1 In Abhängigkeit von der Natur der Softwareentwicklung, kann die Verantwortlichkeit für die Übereinstimmung mit 7.4.5 auf mehreren Beteiligten ruhen. Die Aufteilung der Verantwortlichkeiten muss während der Sicherheitsplanung dokumentiert werden (siehe IEC 61508-1, 6).

7.4.5.2 Die folgenden Informationen müssen vor dem Beginn des detaillierten Entwurfs zu Verfügung stehen: Die Spezifikation der Anforderungen an das sicherheitsbezogene E/E/PE-System; der Entwurf der Software-Architektur; der Validierungsplan für die Softwareaspekte der Systemsicherheit.

7.4.5.3 Die Software muss erstellt werden, um Modularität, Prüfbarkeit und die Fähigkeit sicherer Modifizierbarkeit zu erreichen.

7.4.5.4 Für jedes wichtige Element/Teilsystem im Entwurf der Softwarearchitektur muss eine weitere Verfeinerung des Entwurfs, beruhend auf einer Teilung in Softwaremodule, erfolgen (d. h. Spezifikation des Software-Systementwurfs). Der Entwurf und die anzuwendenden Verifikationen für jedes Softwaremodul müssen spezifiziert werden.

ANMERKUNG 1 Zu bereits existierenden Softwareelementen siehe 7.4.2.

ANMERKUNG 2 Die Verifikation beinhaltet Test und Analyse.

7.4.5.5 Für das Softwaresystem müssen angemessene Integrationstests spezifiziert werden um sicherzustellen, dass das Softwaresystem die Spezifikation der Software-Sicherheitsanforderungen, in Höhe des erforderlichen Sicherheits-Integritätslevels, erfüllt.

7.4.6 Anforderungen an die Codeimplementierung

ANMERKUNG In dem Maß, wie durch den Sicherheits-Integritätslevel erforderlich, muss der Quelltext folgende Eigenschaften besitzen (siehe Anhänge A und B für spezielle Verfahren und siehe Anhang C für eine Hilfestellung zur Interpretation der Eigenschaften):

- lesbar, verständlich und prüfbar sein;
- die spezifizierten Anforderungen des Entwurfs der Softwaremodule erfüllen (siehe 7.4.5);
- die spezifizierten Anforderungen der Programmierrichtlinien erfüllen (siehe 7.4.4);
- alle wichtigen Anforderungen, die während der Sicherheitsplanung spezifiziert wurden, erfüllen (siehe 6).

7.4.6.1 Jedes Modul des Softwarecodes muss überprüft werden. Wo der Code durch ein automatisches Werkzeug erzeugt wurde, müssen die Anforderungen aus 7.4.4 erfüllt werden. Wo der Quelltext aus wieder verwendeter, bereits existierender, Software besteht, müssen die Anforderungen von 7.4.2 erfüllt werden.

ANMERKUNG Die Code-Überprüfung ist eine Verifikationstätigkeit (siehe 7.9).

7.4.7 Anforderungen an den Test von Softwaremodulen

ANMERKUNG 1 Der Test, dass das Softwaremodul die Prüfspezifikation korrekt erfüllt, ist eine Verifikationstätigkeit (siehe 7.9). Es ist die Kombination von Code-Überprüfung und Test des Softwaremoduls, die die Sicherheit liefert, dass ein Softwaremodul die zugehörige Spezifikation erfüllt, d. h., dass es verifiziert ist.

ANMERKUNG 2 Zur Auswahl geeigneter Verfahren und Maßnahmen (siehe Anhänge A und B). Zur Implementierung der Anforderungen dieses Abschnitts sollten die folgenden Eigenschaften (zur Anleitung siehe Anhang C) und Tests der Softwaremodule, berücksichtigt werden:

- die Vollständigkeit der Tests und Integration in Bezug zur Spezifikation des Softwareentwurfs;
- die Korrektheit der Tests und Integration in Bezug zur Spezifikation des Softwareentwurfs (erfolgreiche Fertigstellung);
- die Wiederholbarkeit;
- eine Genaue Definition der Testkonfiguration.

7.4.7.1 Jedes Softwaremodul das während des Entwurfs des Softwaresystems entwickelt wurde, muss, wie durch die Spezifikation des Tests der Softwaremodule gefordert, getestet werden (siehe 7.4.5).

ANMERKUNG Die Verifikation beinhaltet Test und Analyse.

7.4.7.2 Diese Verifikation muss zeigen, ob jedes Softwaremodul seine bestimmungsgemäße Funktion und keine unerwünschten Funktionen ausführt.

ANMERKUNG 1 Dies beinhaltet weder den Test aller Eingangskombinationen noch den Test aller Ausgangskombinationen. Der Test aller Äquivalenzklassen oder Tests auf Strukturebene kann ausreichend sein. Eine Grenzwertanalyse oder Kontrollflussanalyse kann die Testfälle auf eine akzeptable Anzahl reduzieren. Analysierbare Programme machen es einfacher die Anforderungen zu erfüllen. Siehe IEC 61508-7 Anhang C zu diesen Verfahren.

ANMERKUNG 2 Dort, wo die Entwicklung formale Methoden verwendet, wie formale Beweise oder Assertions, kann der Umfang solcher Tests reduziert werden. Siehe IEC 61508-7 Anhang C zu diesen Verfahren.

ANMERKUNG 3 Obwohl die systematische Integrität normalerweise nicht quantifiziert ist (siehe IEC 61508-4, 3.5.6) ist ein quantitativer statistischer Beweis (z. B. Statistische Tests, erhöhte Zuverlässigkeit) akzeptierbar, wenn alle relevanten Bedingungen für einen gültigen statistischen Beweis erfüllt sind. Siehe z. B. IEC 61508-7 Anhang D.

ANMERKUNG 4 Wenn ein Modul einfach genug ist um einen umfassenden Test durchzuführen, dann ist das der wirksamste Weg, eine Übereinstimmung nachzuweisen.

7.4.7.3 Die Ergebnisse der Tests der Softwaremodule müssen dokumentiert werden.

7.4.7.4 Die Verfahren für Korrekturen bei nicht bestandenem Test müssen spezifiziert werden.

7.4.8 Anforderungen an den Softwareintegrationstest

ANMERKUNG 1 Siehe auch die Tabellen A.5, B.2, B.3 und B.6.

ANMERKUNG 2 Der Test, dass die Software ordnungsgemäß integriert wurde, ist eine Verifikationstätigkeit (siehe 7.9).

7.4.8.1 Der Softwareintegrationstest muss während der Entwurfs- und Entwicklungsphase spezifiziert werden (siehe 7.4.5).

7.4.8.2 Die Spezifikation der System-Integrations-Tests der Software muss folgendes festlegen:

- a) die Aufteilung der Software in handhabbare Integrationseinheiten;
- b) die Testfälle und Testdaten;
- c) die Arten der Tests, die durchzuführen sind;
- d) die Testumgebung, Werkzeuge, die Konfiguration und Programme;

- e) die Testkriterien, nach denen die Vollständigkeit der Tests beurteilt werden wird; und
- f) die Verfahren für Korrekturen bei Versagen im Test.

7.4.8.3 Die Software muss in Übereinstimmung mit den Softwareintegrationstests getestet werden. Diese Tests sind in der Spezifikation des Integrationstests der Software, spezifiziert. Die Tests müssen zeigen, dass alle Softwaremodule und Softwareelemente/-teilsysteme ordnungsgemäß zusammenarbeiten, um ihre bestimmungsgemäße Funktion und keine anderen auszuführen.

ANMERKUNG 1 Dies beinhaltet weder den Test aller Eingangskombinationen, noch den aller Ausgangskombinationen. Der Test aller Äquivalenzklassen oder Tests auf Strukturebene kann ausreichend sein. Eine Grenzwertanalyse oder Kontrollflussanalyse kann die Testfälle auf eine akzeptable Anzahl reduzieren. Analysierbare Programme machen es einfacher die Anforderungen zu erfüllen. Siehe IEC 61508-7 Anhang C zu solchen Verfahren.

ANMERKUNG 2 Dort, wo die Entwicklung formale Methoden verwendet, wie formale Beweise oder Assertions, können solche Tests im Umfang reduziert werden. Siehe IEC 61508-7 Anhang C zu solchen Verfahren.

ANMERKUNG 3 Obwohl die systematische Integrität normalerweise nicht quantifiziert ist (siehe IEC 61508-4, 3.5.6) ist ein quantitativer statistischer Beweis (z. B. Statistische Tests, erhöhte Zuverlässigkeit) akzeptierbar, wenn alle relevanten Bedingungen für einen gültigen statistischen Beweis erfüllt sind z. B. siehe IEC 61508-7 Anhang D.

7.4.8.4 Die Ergebnisse der Softwareintegrationstests müssen dokumentiert werden und es müssen Aussagen getroffen werden, ob die Ziele und Testkriterien erreicht wurden. Sollte ein Versagen festgestellt werden, müssen die Gründe für dieses Versagen dokumentiert werden.

7.4.8.5 Während der Softwareintegration muss jede Modifikation der Software einer Einflussanalyse unterzogen werden, um den entsprechenden Einfluss auf alle Softwaremodule zu ermitteln, sowie die notwendigen Tätigkeiten für die Neuverifikation und den Neuentwurf.

7.5 Integration der programmierbaren Elektronik (Hardware und Software)

ANMERKUNG Diese Phase entspricht dem Kasten 10.4 in Bild 3.

7.5.1 Ziele

7.5.1.1 Das erste Ziel der Anforderungen dieses Unterabschnittes ist die Integration der Software in die Zielhardware der programmierbaren Elektronik.

7.5.1.2 Das zweite Ziel der Anforderungen dieses Unterabschnittes ist es, die Software und die Hardware in der sicherheitsbezogenen programmierbaren Elektronik zusammenzuführen, um ihre Verträglichkeit sicherzustellen und die Anforderungen des angestrebten Sicherheits-Integritätslevels zu erfüllen.

ANMERKUNG 1 Das Testen, dass die Software ordnungsgemäß in die Hardware der programmierbaren Elektronik integriert wurde, ist eine Verifikationstätigkeit (siehe 7.9).

ANMERKUNG 2 In Abhängigkeit von der Art der Anwendung können diese Tätigkeiten mit 7.4.8 kombiniert werden.

7.5.2 Anforderungen

ANMERKUNG Zur Auswahl geeigneter Verfahren und Maßnahmen (siehe Anhänge A und B). Zur Implementierung der Anforderungen dieses Abschnitts sollten die folgenden Eigenschaften für die Integration (zur Anleitung siehe Anhang C) berücksichtigt werden:

- Vollständigkeit der Integration in Bezug auf die Spezifikation des Entwurfs;
- Korrektheit der Integration in Bezug auf die Spezifikationen der Entwürfe (erfolgreiche Fertigstellung);
- Wiederholbarkeit;
- genaue Beschreibung der Konfiguration der Integration.

7.5.2.1 Die Integrationstests müssen während der Entwurfs- und Entwicklungsphase (siehe 7.4.3) spezifiziert werden, um die Verträglichkeit der Hard- und Software der sicherheitsbezogenen programmierbaren Elektronik sicherzustellen.

ANMERKUNG Eine enge Zusammenarbeit mit dem Entwickler des E/E/PE-Systems kann notwendig sein, um die Tests der Integration zu entwickeln.

7.5.2.2 Die Software/PE Integrationstest-Spezifikation (Hardware und Software) muss folgendes festlegen:

- a) die Aufteilung des Systems in Integrationsstufen;
- b) die Testfälle und Testdaten;
- c) die durchzuführenden Arten von Tests;
- d) die Testumgebung mit Werkzeugen, Hilfssoftware und Beschreibung der Konfiguration;
- e) die Testkriterien, ab wann der Test als ausgeführt angesehen wird.

7.5.2.3 Die Software/PE Integrationstest-Spezifikation (Hardware und Software) muss zwischen den Tätigkeiten unterscheiden, die durch den Entwickler im Haus durchgeführt werden können und denen, die den Zugriff am Standort des Anwenders benötigen.

7.5.2.4 Die Software/PE Integrationstest-Spezifikation (Hardware und Software) muss zwischen folgenden Tätigkeiten unterscheiden:

- a) der Zusammenführung des Softwaresystems mit der Zielhardware der programmierbaren Elektronik;
- b) der E/E/PE-Integration, d. h. Hinzufügen der Schnittstellen wie Sensoren und Aktoren;
- c) der Anwendung des sicherheitsbezogenen E/E/PE-Systems in dem EUC.

ANMERKUNG Die Punkte b) und c) werden durch die IEC 61508-1 und IEC 61508-2 behandelt und sind hier der Vollständigkeit halber aufgeführt, und um Punkt a) im Zusammenhang darzustellen. Diese liegen normalerweise nicht im Verantwortungsbereich der Softwareentwickler.

7.5.2.5 Die Software muss in die Hardware der sicherheitsbezogenen programmierbaren Elektronik in Übereinstimmung mit der Software/PE Integrationstest-Spezifikation (Hardware und Software) integriert werden.

7.5.2.6 Während des Integrationstests der sicherheitsbezogenen programmierbaren Elektronik (Hardware und Software) muss jede Modifikation des integrierten Systems einer Einflussanalyse unterzogen werden, um die betroffenen Softwaremodule zu ermitteln, ebenso wie die notwendigen Tätigkeiten für die Neuverifikation.

7.5.2.7 Testfälle und ihre Ergebnisse müssen für eine spätere Analyse dokumentiert werden.

7.5.2.8 Der Integrationstest der sicherheitsbezogenen programmierbaren Elektronik (Hardware und Software) muss einschließlich der Testergebnisse dokumentiert werden und es müssen Aussagen getroffen werden, ob die Ziele und Kriterien des Tests erreicht wurden. Sollte ein Versagen festgestellt werden, müssen die Gründe für dieses Versagen dokumentiert werden. Jede Modifikation des integrierten Systems muss einer Einflussanalyse unterzogen werden, um die betroffenen Softwareelemente/-module zu ermitteln sowie die notwendigen Tätigkeiten für Neuverifikation und Neuentwurf.

7.6 Software-Betriebs- und Modifikationsverfahren

ANMERKUNG Diese Phase entspricht dem Kasten 10.5 in Bild 3.

7.6.1 Ziel

Das Ziel der Anforderungen dieses Unterabschnittes ist es, notwendige Informationen und Verfahren hinsichtlich der Software zu liefern, um sicherzustellen, dass die funktionale Sicherheit des sicherheitsbezogenen E/E/PE-Systems während des Betriebs und bei Modifikationen aufrecht erhalten bleibt.

7.6.2 Anforderungen

Die Anforderungen werden in 7.6 der IEC 61508-2 und in 7.8 dieser Norm gegeben.

ANMERKUNG In dieser Norm kann Software (im Gegensatz zu Hardware) nicht Instand gehalten werden: sie wird immer modifiziert.

7.7 Softwareaspekte bezüglich der Validierung der Systemsicherheit

ANMERKUNG 1 Siehe auch die Tabellen A.7, B.3 und B.5.

ANMERKUNG 2 Diese Phase entspricht Kasten 10.6 in Bild 3.

ANMERKUNG 3 Software kann üblicherweise nicht separat zu seiner zugehörigen Hardware und Systemumgebung validiert werden.

7.7.1 Ziel

Das Ziel der Anforderungen dieses Unterabschnittes ist es, sicherzustellen, dass das integrierte System mit der Spezifikation der Software-Sicherheitsanforderungen, mit dem erforderlichen Sicherheits-Integritätslevel übereinstimmt.

7.7.2 Anforderungen

ANMERKUNG Zur Auswahl angemessener Verfahren und Maßnahmen (siehe Anhänge A und B) um die Anforderungen dieses Abschnitts zu implementieren, sollten folgende Eigenschaften (zur Anleitung siehe Anhang C) berücksichtigt werden:

- die Vollständigkeit der Validierung in Bezug zur Spezifikation des Softwareentwurfs;
- die Korrektheit der Validierung in Bezug zur Spezifikation des Softwareentwurfs (erfolgreiche Fertigstellung);
- die Wiederholbarkeit;
- eine Genaue Definition der Validierungskonfiguration.

7.7.2.1 Wenn die Übereinstimmung mit den Anforderungen an die sicherheitsbezogene Software bereits als Teil der Planung der Validierung der Sicherheit des sicherheitsbezogenen E/E/PE-Systems (siehe IEC 61508-61508-2, 7.7) nachgewiesen wurde, dann muss die Validierung nicht wiederholt werden.

7.7.2.2 Die Validierung muss, wie im Validierungsplan für die Softwareaspekte der Systemsicherheit spezifiziert, durchgeführt werden.

7.7.2.3 In Abhängigkeit von der Natur der Softwareentwicklung, kann die Verantwortlichkeit für die Übereinstimmung mit 7.7 auf mehreren Beteiligten ruhen. Die Aufteilung der Verantwortlichkeiten muss während der Sicherheitsplanung dokumentiert werden (siehe IEC 61508-1, 6)

7.7.2.4 Die Ergebnisse der Validierung der Softwareaspekte der Systemsicherheit müssen dokumentiert werden.

7.7.2.5 Für jede Sicherheitsfunktion muss die Validierung der Sicherheit der Software folgende Ergebnisse dokumentieren:

- a) die chronologische Aufzeichnung der Validierungstätigkeiten die es erlauben die Folge von Tätigkeiten zurück zu verfolgen;

ANMERKUNG Bei der Aufzeichnung der Testergebnisse ist es wichtig, dass es möglich ist die Reihenfolge der Tätigkeiten zurück zu verfolgen. Die Gewichtung dieser Anforderung liegt auf der Rückverfolgbarkeit der Tätigkeiten und nicht bei der Erzeugung einer zeitlichen/datierten Liste von Dokumenten.

- b) die Version des verwendeten Validierungsplans für die Softwareaspekte der Systemsicherheit (siehe 7.3);
- c) die (durch Testen oder Analyse) validierte Sicherheitsfunktion, zusammen mit dem Bezug zum Validierungsplan für die Softwareaspekte der Systemsicherheit;

- d) die verwendeten Werkzeuge und Ressourcen mit den Daten der Kalibrierung;
- e) die Ergebnisse der Validierungstätigkeiten;
- f) die Abweichungen zwischen den erwarteten und den tatsächlichen Ergebnissen.

7.7.2.6 Wenn Abweichungen zwischen den erwarteten und den tatsächlichen Ergebnissen auftreten, müssen die durchgeführten Analysen und getroffenen Entscheidungen, ob die Validierung fortgeführt oder eine Modifikationsanfrage herausgegeben und zu einem früheren Teil des Entwicklungslebenszykluses zurück gekehrt wird, als Teil der Ergebnisse der Validierung der Softwareaspekte der Systemsicherheit dokumentiert werden.

ANMERKUNG Die Anforderungen von 7.7.2.2 bis 7.7.2.6 basieren auf den allgemeinen Anforderungen der IEC 61508-1, 7.14.

7.7.2.7 Die Validierung der sicherheitsbezogenen Softwareaspekte der Systemsicherheit muss die folgenden Anforderungen abdecken:

- a) die hauptsächliche Validierungsmethode für die Software muss der Test sein; die Analyse, Animation und Modellbildung kann die Validierung ergänzen;
- b) die Software muss ausgeführt werden mit Simulation der
 - 1) Eingangssignale des normalen Betriebs,
 - 2) erwarteten Ereignissen,
 - 3) unerwünschten Bedingungen, die eine Systemreaktion erfordern;
- c) der Lieferant und/oder Entwickler muss dem Systementwickler alle dokumentierten Ergebnisse (oder die verschiedenen Beteiligten mit Verantwortung zur Übereinstimmung) der Validierung der Software bezüglich der Sicherheit und alle weiteren relevanten Dokumente verfügbar machen, um es ihm zu ermöglichen, die Anforderungen der IEC 61508-1 und IEC 61508-2 zu erfüllen.

7.7.2.8 Softwarewerkzeuge müssen die Anforderungen von 7.4.4 erfüllen.

7.7.2.9 Die Ergebnisse der Validierung der sicherheitsbezogenen Softwareaspekte der Systemsicherheit müssen folgende Anforderungen erfüllen:

- a) Tests müssen zeigen, dass die spezifizierten Anforderungen an die sicherheitsbezogene Software (siehe Spezifikation der Software-Sicherheitsanforderungen) richtig umgesetzt wurden und die Software keine nicht bestimmungsgemäßen Funktionen ausführt;
- b) Testfälle und ihre Ergebnisse müssen für nachfolgende Analysen und unabhängige Beurteilungen, wie durch den Sicherheits-Integritätslevel gefordert (siehe IEC 61508-1, 8.2.12), dokumentiert werden;
- c) die dokumentierten Ergebnisse der Validierung der Softwareaspekte der Systemsicherheit müssen zeigen, ob (1) die Software die Validierung bestanden hat, oder (2) die Gründe warum nicht.

7.8 Softwaremodifikation

ANMERKUNG Diese Phase entspricht dem Kasten 10.5 in Bild 3.

7.8.1 Ziel

Das Ziel der Anforderungen dieses Unterabschnittes ist es eine Anleitung zu geben, Modifikationen, Erweiterungen oder Anpassungen an der validierten Software durchzuführen, wobei sichergestellt wird, dass die systematische Eignung der Software erhalten bleibt.

7.8.2 Anforderungen

ANMERKUNG Zur Auswahl angemessener Verfahren und Maßnahmen (siehe Anhänge A und B) um die Anforderungen dieses Abschnitts zu implementieren, sollten folgende Eigenschaften für die Softwaremodifikation (zur Anleitung siehe Anhang C) berücksichtigt werden:

- Vollständigkeit der Modifikation in Bezug auf ihre Anforderungen;
- Korrektheit der Modifikation in Bezug auf ihre Anforderungen;

- Freiheit von der Einleitung innerer Entwurfsfehler;
- Vermeidung von unerwartetem Verhalten;
- Verifizierbarer und testbarer Entwurf;
- Regressionstest und Verifikationsabdeckung.

7.8.2.1 Bevor eine Softwaremodifikation durchgeführt wird, müssen die Verfahren für die Softwaremodifikation zur Verfügung gestellt werden (siehe IEC 61508-1, 7.16).

ANMERKUNG 1 7.8.2.1 bis 7.8.2.9 finden hauptsächlich Anwendung bei Modifikationen während der Betriebsphase der Software. Sie können jedoch auch während der Phase der Integration der programmierbaren Elektronik und der Phase der Gesamtinstallation und Gesamtinbetriebnahme angewendet werden (siehe IEC 61508-1, 7.13).

ANMERKUNG 2 Ein Beispiel des Modells eines Modifikationsverfahrens wird in IEC 61508-1 Bild 9 gezeigt.

7.8.2.2 Eine Modifikation darf nur dann begonnen werden, wenn eine autorisierte Softwaremodifikationsanforderung unter den während der Sicherheitsplanung spezifizierten Verfahren (siehe 6) besteht, die folgendes beschreibt

- a) die Gefahren, die betroffen sein könnten;
- b) die vorgeschlagene Modifikation;
- c) die Gründe für die Modifikation.

ANMERKUNG Der Grund für eine Modifikationsanforderung könnte z. B. sein

- die funktionale Sicherheit liegt unterhalb der spezifizierten;
- Erfahrungen mit systematischen Fehlern;
- neue oder geänderte Sicherheitsgesetze;
- Modifikationen an der EUC oder ihrer Verwendung;
- Modifikationen der Anforderungen zur Gesamtsicherheit;
- eine Analyse der Leistungsfähigkeit des Betriebs und der Softwarepflege die zeigt, dass die Erwartungen nicht erreicht wurden;
- routinemäßige Audits der funktionalen Sicherheit.

7.8.2.3 Es muss eine Analyse über die Auswirkung der vorgeschlagenen Softwaremodifikation auf die funktionale Sicherheit des sicherheitsbezogenen E/E/PE-Systems durchgeführt werden

- a) um festzustellen, ob eine Gefährdungs- und Risikoanalyse notwendig ist;
- b) um festzustellen, welche Phasen des Software-Sicherheitslebenszyklus wiederholt werden müssen.

7.8.2.4 Die aus 7.8.2.3 erhaltenen Ergebnisse der Einflussanalyse müssen dokumentiert werden.

7.8.2.5 Alle Modifikationen, die einen Einfluss auf die funktionale Sicherheit des sicherheitsbezogenen E/E/PE-Systems haben, müssen die Rückkehr zu einer angemessenen Phase des Software-Sicherheitslebenszyklus einleiten. Alle nachfolgenden Phasen müssen dann in Übereinstimmung mit den für diese Phase nach den Anforderungen dieser Norm spezifizierten Verfahren durchgeführt werden. Eine Sicherheitsplanung (siehe 6) muss alle weiteren nachfolgenden Tätigkeiten aufführen.

ANMERKUNG Es kann notwendig werden eine vollständige Gefährdungs- und Risikoanalyse durchzuführen, die andere Sicherheits-Integritätslevel erbringt als die die aktuell, für die Sicherheitsfunktionen durch das sicherheitsbezogene E/E/PE-Systeme realisierten, festgelegt worden sind.

7.8.2.6 Die Sicherheitsplanung für Modifikationen an der sicherheitsbezogenen Software muss die Anforderungen der IEC 61508-1, 6 erfüllen, im Besonderen:

- a) die Benennung des Personals und Festlegung deren benötigter Fähigkeiten;
- b) eine genaue Spezifikation der Modifikation;

- c) die Planung der Verifikation;
- d) den Umfang von Neuvalidierung und Test der Modifikation, in dem durch den Sicherheits-Integritätslevel geforderten Maß.

7.8.2.7 Die Modifikation muss wie geplant durchgeführt werden.

7.8.2.8 Die Einzelheiten aller Modifikationen müssen dokumentiert werden mit Hinweisen auf

- a) die Anforderung der Modifikation/Nachrüstung;
- b) die Ergebnisse der Einflussanalyse, die die Auswirkungen der vorgeschlagenen Softwaremodifikation auf die funktionale Sicherheit beurteilt und die getroffenen Entscheidungen mit den zugehörigen Begründungen;
- c) den Werdegang des Konfigurationsmanagements der Software;
- d) die Abweichung vom normalen Betrieb und von normalen Bedingungen;
- e) alle dokumentierten Informationen, die durch die Modifikationen betroffen sind.

7.8.2.9 Informationen (z. B. eine Aufzeichnung) zu den Einzelheiten aller Modifikationen müssen dokumentiert werden. Die Dokumentation muss die Neuverifikation und Neuvalidierung von Daten und Ergebnissen beinhalten.

7.8.2.10 Die Beurteilung der geforderten Modifikation oder Nachrüstung muss von den Ergebnissen der Einflussanalyse und der systematischen Eignung der Software abhängig sein.

7.9 Software-Verifikation

7.9.1 Ziel

Das Ziel der Anforderungen dieses Unterabschnittes ist es, in dem wie durch den Sicherheits-Integritätslevel geforderten Umfang, die Ausgaben einer gegebenen Phase des Software-Sicherheitslebenszykluses zu testen und zu bewerten, um die Korrektheit und Konsistenz bezüglich der Ausgaben und Normen, die als Eingaben für diese Phase vorgesehen sind, sicherzustellen.

ANMERKUNG 1 Dieser Unterabschnitt betrachtet die allgemeinen Aspekte der Verifikation, die in mehreren Phasen des Lebenszyklus gleich sind. Dieser Unterabschnitt stellt weder weitere Anforderungen an die Teile der Verifikation, die einen Test darstellen, wie 7.4.7 (Test der Softwaremodule), 7.4.8 (Softwareintegration) und 7.5 (Integration der programmierbaren Elektronik) und in sich selbst Verifikationstätigkeiten darstellen, noch fordert dieser Unterabschnitt zusätzliche Verifikationen zur Software-Validierung (siehe 7.7), die in dieser Norm die Übereinstimmung mit der Spezifikation der Sicherheitsanforderungen (Durchgehende Verifikation) darlegt. Die Kontrolle, ob die Spezifikation der Sicherheitsanforderungen selbst korrekt ist, wird durch Fachsachverständige durchgeführt.

ANMERKUNG 2 In Abhängigkeit von der Softwarearchitektur darf die Verantwortung für die Tätigkeiten der Verifikation zwischen allen an der Entwicklung und Modifikation der Software beteiligten Organisationen aufgeteilt werden.

7.9.2 Anforderungen

ANMERKUNG Zur Auswahl angemessener Verfahren und Maßnahmen (siehe Anhänge A und B) um die Anforderungen dieses Abschnitts zu implementieren, sollten folgende Eigenschaften für die Software-Verifikation (zur Anleitung siehe Anhang C) berücksichtigt werden:

- die Vollständigkeit der Verifikation in Bezug auf die vorherige Phase;
- die Korrektheit der Verifikation in Bezug auf die vorherige Phase (erfolgreiche Fertigstellung);
- die Wiederholbarkeit;
- eine Genaue Definition der Verifikationskonfiguration;

7.9.2.1 Die Verifikation der Software muss gleichzeitig mit der Entwicklung für jede Phase des Software-Sicherheitslebenszyklus geplant (siehe 7.3) und dokumentiert werden.

7.9.2.2 Die Planung der Softwareverifikation muss auf die Kriterien, Verfahren und Werkzeuge, die während der Verifikation verwendet werden hinweisen und muss ansprechen

- f) die Bewertung der Anforderungen der Sicherheitsintegrität;
- g) die Auswahl und Dokumentation der Verifikationsstrategien, Tätigkeiten und Verfahren;
- h) die Auswahl und Anwendung der Verifikationswerkzeuge (Prüfeinrichtungen, spezielle Testsoftware, Simulatoren für Ein- und Ausgänge usw.);
- i) die Bewertung der Verifikationsergebnisse;
- j) die getroffenen Berichtigungsmaßnahmen.

7.9.2.3 Die Verifikation der Software muss wie geplant durchgeführt werden.

ANMERKUNG Die Auswahl der Verfahren, die Maßnahmen für die Verifikation und der Grad der Unabhängigkeit der Verifikation hängen von mehreren Faktoren ab und dürfen in anwendungsbezogenen Normen festgelegt werden. Die Faktoren können z. B. folgendes beinhalten

- die Größe des Projektes;
- den Grad der Komplexität;
- den Neuheitsgrad des Entwurfs;
- den Neuheitsgrad der Technologie.

7.9.2.4 Der Nachweis, dass die verifizierte Phase in jeder Hinsicht zufrieden stellend abgeschlossen wurde, muss dokumentiert werden.

7.9.2.5 Nach jeder Verifikation muss die Dokumentation der Verifikation enthalten

- a) die Kennzeichnung der verifizierten Punkte;
- b) die Kennzeichnung der Informationen, gegen die die Verifikation durchgeführt wurde;
- c) Nichtübereinstimmungen.

ANMERKUNG Beispiele für Nichtübereinstimmungen beinhalten Softwaremodule, Datenstrukturen und Algorithmen, die schlecht auf das Problem angepasst wurden.

7.9.2.6 Alle wesentlichen Informationen der Phase N des Software-Sicherheitslebenszyklus, die für die ordnungsgemäße Ausführung der Phase N+1 benötigt werden, müssen verfügbar und sollten verifiziert sein. Die Ausgänge der Phase N beinhalten

- a) eine ausreichende Spezifikation, Beschreibung des Entwurfs oder Codes in Phase N, für
 - 1) die Funktionalität;
 - 2) die Sicherheitsintegrität, die Leistungsfähigkeit und anderen Anforderungen an die Planung der Sicherheit (siehe Abschnitt 6);
 - 3) die Lesbarkeit durch das Entwicklungsteam;
 - 4) die Prüfbarkeit für weitere Verifikationen;
 - 5) die sichere Modifikation, um eine spätere Weiterentwicklung zu ermöglichen;
- b) eine ausreichende Planung der Validierung und/oder Tests für Phase N, um den Entwurf der Phase N zu beschreiben und zu spezifizieren;
- c) Kontrolle auf Unverträglichkeiten zwischen
 - 1) den spezifizierten Tests in Phase N und den Tests, die in der vorherigen Phase N -1 spezifiziert sind;
 - 2) den Ausgaben innerhalb der Phase N.

7.9.2.7 In Abhängigkeit des Lebenszyklus der Softwareentwicklung (siehe 7.1.2.1), müssen die folgenden Verifikationstätigkeiten durchgeführt werden:

- a) die Verifikation der Software-Sicherheitsanforderungen;
- b) die Verifikation der Software-Architektur;

- c) die Verifikation des Software-Systementwurfs;
- d) die Verifikation des Entwurfs der Softwaremodule;
- e) die Verifikation des Codes;
- f) die Verifikation der Daten;
- g) die Verifikation des zeitlichen Leistungsverhaltens;
- h) der Test der Softwaremodule (siehe 7.4.7);
- i) der Softwareintegrationstest (siehe 7.4.8);
- j) der Test der Integration der programmierbaren Elektronik (siehe 7.5);
- k) die Softwareaspekte bezüglich der Validierung der Systemsicherheit (siehe 7.7).

ANMERKUNG Zu den Anforderungen von (a) bis (g) siehe unten.

7.9.2.8 Die Verifikation der Software-Sicherheitsanforderungen: Nachdem die Spezifikation der Software-Sicherheitsanforderungen vervollständigt wurde und bevor die nächste Phase des Entwurfs und der Entwicklung der Software beginnt, muss die Verifikation

- a) berücksichtigen, ob die Spezifikation der Software-Sicherheitsanforderungen, die E/E/PE-System Sicherheits-Anforderungsspezifikation (siehe IEC 61508-2, 7.10) für Funktionalität, Sicherheitsintegrität, die Leistungsfähigkeit und für alle weiteren Anforderungen der Planung der Sicherheit, angemessen erfüllt;
- b) berücksichtigen, ob der Validierungsplan für die Softwareaspekte der Systemsicherheit, die Spezifikation der Software-Sicherheitsanforderungen angemessen erfüllt;
- c) Unverträglichkeiten kontrollieren, und zwar zwischen
 - 1) der Spezifikation der Software-Sicherheitsanforderungen und der E/E/PE-System Sicherheits-Anforderungsspezifikation (siehe IEC 61508-1, 7.10 und IEC 61508-2, 7.2),
 - 2) der Spezifikation der Software-Sicherheitsanforderungen und dem Validierungsplan für die Softwareaspekte der Systemsicherheit.

7.9.2.9 Verifikation der Software-Architektur: Nachdem der Entwurf der Softwarearchitektur abgeschlossen wurde, muss die Verifikation

- a) berücksichtigen, ob der Entwurf der Softwarearchitektur die Spezifikation der Software-Sicherheitsanforderungen angemessen erfüllt;
- b) berücksichtigen, ob die spezifizierten Tests der Integration, zum Entwurf der Softwarearchitektur angemessen sind;
- c) berücksichtigen, ob die Eigenschaften jeder Hauptkomponente, jedes Teilsystems angemessen sind, mit Bezug zur
 - 1) Durchführbarkeit der erforderlichen Sicherheitsleistung;
 - 2) Testbarkeit für weitere Verifikationen;
 - 3) Lesbarkeit durch das Entwicklungs- und Verifikationsteam;
 - 4) sicheren Modifikation, um eine Weiterentwicklung zu ermöglichen.
- d) Unverträglichkeiten überprüfen, und zwar zwischen:
 - 1) dem Entwurf der Softwarearchitektur und der Spezifikation der Software-Sicherheitsanforderungen;
 - 2) dem Entwurf der Softwarearchitektur und seiner Integrationstests;
 - 3) den Integrationstests des Entwurfs der Softwarearchitektur und dem Validierungsplan für die Softwareaspekte der Systemsicherheit.

7.9.2.10 Verifikation des Software-Systementwurfs: Nach dem der Software-Systementwurf abgeschlossen wurde, muss die Verifikation

- a) berücksichtigen, ob der Software-Systementwurf (siehe 7.4.5) den Entwurf der Softwarearchitektur angemessen erfüllt;

- b) berücksichtigen, ob die spezifizierten Tests der Integration des Softwaresystems (siehe 7.4.5) den Software-Systementwurf (siehe 7.4.5) angemessen erfüllt;
- c) berücksichtigen, ob die Eigenschaften für jedes Hauptelement der Spezifikation des Software-Systementwurfs (siehe 7.4.5) angemessen sind, mit Bezug zur
 - 1) Durchführbarkeit der erforderlichen Sicherheitsleistung;
 - 2) Testbarkeit für weitere Verifikationen;
 - 3) Lesbarkeit durch das Entwicklungs- und Verifikationsteam;
 - 4) sicheren Modifikation, um eine Weiterentwicklung zu ermöglichen.

ANMERKUNG Die Tests der Integration des Softwaresystems können als Teil der Integrationstests, der Softwarearchitektur spezifiziert werden.

- d) Unverträglichkeiten kontrollieren, zwischen
 - 1) der Spezifikation des Software-Systementwurfs (siehe 7.4.5) und dem Entwurf der Software-Architektur;
 - 2) der Spezifikation des Software-Systementwurfs (siehe 7.4.5) und der Spezifikation der System-Integrations-Tests der Software (siehe 7.4.5);
 - 3) den Tests gefordert durch die Spezifikation der System-Integrations-Tests der Software (siehe 7.4.5) und der Spezifikation der Integrationstests der Software-Architektur (siehe 7.4.3).

7.9.2.11 Verifikation des Entwurfs der Softwaremodule: Nach dem der Entwurf jedes Softwaremoduls abgeschlossen wurde, muss die Verifikation

- a) berücksichtigen, ob die Spezifikation des Entwurfs der Softwaremodule (siehe 7.4.5) die Spezifikation des Software-Systementwurfs (siehe 7.4.5) angemessen erfüllt;
- b) berücksichtigen, ob die Spezifikation des Tests der Softwaremodule (siehe 7.4.5) die Spezifikation des Entwurfs der Softwaremodule (siehe 7.4.5) angemessen erfüllt;
- c) berücksichtigen, ob die Eigenschaften jedes Softwaremoduls angemessen sind, mit Bezug zur
 - 1) Durchführbarkeit der erforderlichen Sicherheitsleistung (siehe Spezifikation der Software-Sicherheitsanforderungen);
 - 2) Testbarkeit für weitere Verifikationen;
 - 3) Lesbarkeit durch das Entwicklungs- und Verifikationsteam;
 - 4) sicheren Modifikation, um eine Weiterentwicklung zu ermöglichen.
- d) Unverträglichkeiten kontrollieren, zwischen
 - 1) der Spezifikation des Entwurfs der Softwaremodule (siehe 7.4.5) und der Spezifikation des Software-Systementwurfs (siehe 7.4.5);
 - 2) der Spezifikation des Entwurfs der Softwaremodule (für jedes Softwaremodul) (siehe 7.4.5) und der Spezifikation des Tests der Softwaremodule (siehe 7.4.5);
 - 3) der Spezifikation des Tests der Softwaremodule (siehe 7.4.5) und der Spezifikation der System-Integrations-Tests der Software (siehe 7.4.5).

7.9.2.12 Die Verifikation des Codes: Der Quelltext muss durch statische Methoden verifiziert werden um sicherzustellen, dass die Spezifikation des Entwurfs der Softwaremodule (siehe 7.4.5) mit den erforderlichen Programmierrichtlinien (siehe 7.4.4) und dem Validierungsplan für die Softwareaspekte der Systemsicherheit, übereinstimmt.

ANMERKUNG In den frühen Phasen des Software-Sicherheitslebenszyklus ist die Verifikation statisch (z. B. Inspektion, Überprüfung, formale Methoden usw.). Die Codeverifikation beinhaltet solche Verfahren wie Softwareinspektion und Walkthrough. Die Kombination der Ergebnisse aus der Codeverifikation und dem Test der Softwaremodule liefert die Gewissheit, dass jedes Softwaremodul seine Spezifikation erfüllt. Ab hier wird der Test das primäre Mittel der Verifikation.

7.9.2.13 Verifikation der Daten

- a) Die Datenstrukturen müssen verifiziert werden.

- b) Die Anwendungsdaten müssen verifiziert werden nach
- 1) Übereinstimmung mit den Datenstrukturen;
 - 2) Vollständigkeit gegen die Anforderungen der Anwendung;
 - 3) Verträglichkeit mit der untergeordneten Systemsoftware (z. B. Reihenfolge der Ausführung, Laufzeit usw.); und
 - 4) Korrektheit der Datenwerte.
- c) Alle Betriebsparameter müssen gegen die Anforderungen der Anwendung verifiziert werden.
- d) Alle Anlagenschnittstellen und zugehörige Software (d. h. Sensoren, Aktoren und Offline-Schnittstellen: siehe 7.2.2.11) müssen verifiziert werden in Bezug auf
- 1) Erkennung von voraussichtlichen Ausfällen/Versagensfällen der Schnittstelle;
 - 2) Toleranz gegen voraussichtliche Ausfälle/Versagensfälle der Schnittstelle.
- e) Alle Kommunikationsschnittstellen und zugehörige Software müssen bis zu einem angemessenen Niveau verifiziert werden, in Bezug auf
- 1) Aufdeckung von Ausfällen;
 - 2) Schutz vor Verfälschung;
 - 3) Validierung der Daten.

7.9.2.14 Verifikation des zeitlichen Leistungsverhaltens: Die Vorhersagbarkeit des Verhaltens innerhalb des Zeitbereichs muss verifiziert werden.

ANMERKUNG Das Zeitverhalten kann beinhalten: die Leistung, die Ressourcen, die Reaktionszeit, die Worst Case Ausführungszeit, eine Überlastung, die Freiheit von (Programm-)verklebungen (en.: deadlock), ein Laufzeitsystem.

8 Beurteilung der funktionalen Sicherheit

ANMERKUNG Zur Auswahl angemessener Verfahren und Maßnahmen (siehe Anhänge A und B) um die Anforderungen dieses Abschnitts zu implementieren, sollten folgende Eigenschaften für die Beurteilung der funktionalen Sicherheit (zur Anleitung siehe Anhang C) berücksichtigt werden:

- die Vollständigkeit der Beurteilung der funktionalen Sicherheit in Bezug zu dieser Norm;
- die Korrektheit der Beurteilung der funktionalen Sicherheit in Bezug zur Spezifikation des Entwurfs (erfolgreiche Fertigstellung);
- ein nachverfolgbarer Abschluss aller identifizierten Probleme;
- die Möglichkeit die Beurteilung der funktionalen Sicherheit nach einer Änderung modifizieren zu können, ohne der Notwendigkeit einer umfangreichen Nacharbeit der Beurteilung;
- die Wiederholbarkeit;
- die Aktualität;
- eine genau definierte Konfiguration.

8.1 Für die Beurteilung der sicherheitsbezogenen Software finden die Ziele und Anforderungen des Abschnittes 8 der IEC 61508-1 Anwendung.

8.2 Wenn es nicht in anwendungsbezogenen internationalen Normen anders festgelegt ist, muss der minimale Grad der Unabhängigkeit derjenigen, die die Beurteilung der funktionalen Sicherheit ausführen, so sein, wie in 8.2 der IEC 61508-1 festgelegt.

8.3 Eine Beurteilung der funktionalen Sicherheit darf die Ergebnisse der Tätigkeiten der Tabelle A.10 verwenden.

ANMERKUNG Die Auswahl der Verfahren aus Anhang A und B gibt allein keine Garantie, dass die erforderliche Sicherheitsintegrität erreicht wird (siehe 7.1.2.7). Der Gutachter sollte ebenso berücksichtigen:

- die Natur der Beschaffenheit und Ergänzung der gewählten Methoden, Sprachen und Werkzeuge für den gesamten Entwicklungszyklus;
- ob die Entwickler die Methoden, Sprachen und Werkzeuge, die sie verwenden, vollständig verstehen;
- ob die Methoden, Sprachen und Werkzeuge den während der Entwicklung auftretenden, besonderen Problemen gut angepasst sind.

Anhang A (normativ)

Leitfaden für die Auswahl der Verfahren und Maßnahmen

A.1 Allgemein

Zu einigen Abschnitten dieser Norm gehören Tabellen, z. B. gehört zu 7.2 (Spezifikation der Software-Sicherheitsanforderung) die Tabelle A.1. Ausführlichere Tabellen in Anhang B erweitern die Inhalte der Tabellen des Anhangs A, so erweitert z. B. Tabelle B.2 das Thema dynamische Analyse und Testen der Tabelle A.5.

Die IEC 61508-7 gibt einen Überblick über die speziellen Verfahren und Maßnahmen, die in den Anhängen A und B referenziert werden.

Für jedes Verfahren oder jede Maßnahme ist in den Tabellen eine Empfehlung für die Sicherheits-Integritätslevel 1 bis 4 gegeben. Die Empfehlungen sehen wie folgt aus.

- ++ Das Verfahren oder die Maßnahme wird für diesen Sicherheits-Integritätslevel besonders empfohlen. Wenn diese(s) Verfahren oder diese Maßnahme nicht verwendet wird, dann sollte der Grund mit Referenz zu Anhang C, während der Sicherheitsplanung ausführlich dargelegt und mit dem Gutachter abgestimmt werden.
- + Das Verfahren oder die Maßnahme wird für diesen Sicherheits-Integritätslevel empfohlen. Es handelt sich um eine geringere Empfehlung als die mit ++ gekennzeichnete.
- o Das Verfahren oder die Maßnahme hat keine Empfehlung für oder gegen die Verwendung.
- Das Verfahren oder die Maßnahme wird für diesen Sicherheits-Integritätslevel ausdrücklich nicht empfohlen. Wenn diese(s) Verfahren oder diese Maßnahme verwendet wird, dann sollte der Grund mit Referenz zu Anhang C, während der Sicherheitsplanung ausführlich dargelegt und mit dem Gutachter abgestimmt werden.

Es müssen entsprechend des Sicherheits-Integritätslevels angemessene Verfahren/Maßnahmen ausgewählt werden. Alternative oder gleichwertige Verfahren/Maßnahmen sind durch einen Buchstaben hinter der Nummer gekennzeichnet. Es muss nur eine der alternativen oder gleichwertigen Verfahren/Maßnahmen erfüllt werden.

Es können andere Maßnahmen oder Verfahren angewendet werden um die Anforderungen und Ziele zu erreichen. Zur Anleitung der Auswahl von Verfahren siehe Anhang C.

Die Rangordnung der Verfahren und Maßnahmen steht mit dem Begriff der Wirksamkeit im Zusammenhang, der in der IEC 61508-2 verwendet wird. Mit ++ gekennzeichnete Verfahren sind bei der Vermeidung von systematischen Fehlern während der Softwareentwicklung oder (für den Fall der Softwarearchitektur) bei der Beherrschung in der Software verbliebener Fehler, die während der Ausführung aufgedeckt werden, wirksamer als solche, die mit + gekennzeichnet sind, sofern alle anderen Faktoren gleich sind.

Wegen der großen Zahl von Faktoren, die die systematische Eignung der Software beeinflussen, ist es nicht möglich, eine Vorschrift zur Kombination der Verfahren und Maßnahmen anzugeben, welches auf alle vorhandenen Anwendungen zutrifft. Eine Anleitung zur Begründung der Auswahl, der speziellen Verfahren um die systematische Eignung der Software zu erreichen, wird in Anhang C gegeben.

Für eine spezielle Anwendung ist die geeignete Kombination von Verfahren oder Maßnahmen mit den gewählten Verfahren oder Maßnahmen während der Sicherheitsplanung festzulegen, es sei denn die Anmerkung in der Tabelle gibt andere Anforderungen.

In der IEC 61508-6 wird eine einleitende Anleitung in Form zweier ausgearbeiteter Beispiele zur Interpretation der Tabellen gegeben.