DIN EN ISO 25119-3

ICS 35.240.99; 65.060.01

Einsprüche bis 2018-12-12
Vorgesehen als Ersatz für
DIN EN 16590-3:2015-04

Entwurf

Traktoren und Maschinen für die Land- und Forstwirtschaft – Sicherheitsbezogene Teile von Steuerungen – Teil 3: Serienentwicklung, Hardware und Software (ISO/FDIS 25119-3:2018); Deutsche und Englische Fassung FprEN ISO 25119-3:2018

Tractors and machinery for agriculture and forestry –
Safety-related parts of control systems –
Part 3: Series development, hardware and software (ISO/FDIS 25119-3:2018);
German and English version FprEN ISO 25119-3:2018

Tracteurs et matériels agricoles et forestiers –
Parties des systèmes de commande relatives à la sécurité –
Partie 3: Développement en série, matériels et logiciels (ISO/FDIS 25119-3:2018);
Version allemande et anglaise FprEN ISO 25119-3:2018

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2018-10-12 wird der Öffentlichkeit zur Prüfung und Stellungnahme vorgelegt.

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

Stellungnahmen werden erbeten

- vorzugsweise online im Norm-Entwurfs-Portal von DIN unter www.din.de/go/entwuerfe bzw. für Norm-Entwürfe der DKE auch im Norm-Entwurfs-Portal der DKE unter www.entwuerfe.normenbibliothek.de, sofern dort wiedergegeben;
- oder als Datei per E-Mail an nam@din.de möglichst in Form einer Tabelle. Die Vorlage dieser Tabelle kann im Internet unter www.din.de/go/stellungnahmen-norm-entwuerfe oder für Stellungnahmen zu Norm-Entwürfen der DKE unter www.dke.de/stellungnahme abgerufen werden;
- oder in Papierform an den DIN-Normenausschuss Maschinenbau (NAM), 60498 Frankfurt am Main, Postfach 71 08 64, Lyoner Str. 18, 60528 Frankfurt am Main.

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

Gesamtumfang 141 Seiten

DIN-Normenausschuss Maschinenbau (NAM)

AVA Documents Document Sharing Hub

Full document details: E DIN EN ISO 25119-3:2018-11

Nationales Vorwort

Dieser Norm-Entwurf enthält sicherheitstechnische Festlegungen.

Sie beinhaltet die Deutsche Fassung der vom Technischen Komitee ISO/TC 23 „Tractors and Machinery for Agriculture and Forestry“ in Zusammenarbeit mit dem Technischen Komitee CEN/TC 144 „Traktoren und land- und forstwirtschaftliche Maschinen“ im Europäischen Komitee für Normung (CEN) ausgearbeiteten FprEN ISO 25119-3:2018.

Um Zweifelsfälle in der Übersetzung auszuschließen, ist die englische Originalfassung beigefügt. Die Nutzungsbedingungen für den deutschen Text des Norm-Entwurfes gelten gleichermaßen auch für den englischen Text.

Die nationalen Interessen bei der Erarbeitung wurden vom Ausschuss NA 060-16-12 AA „TA E - Elektronik - Landtechnik“ im Fachbereich „Landtechnik“ des DIN-Normenausschusses Maschinenbau (NAM) wahrgenommen. Vertreter der Hersteller und Anwender von sicherheitsbezogenen Teilen von Steuerungen für Traktoren und Maschinen für die Land- und Forstwirtschaft sowie der Berufsgenossenschaften waren an der Erarbeitung beteiligt.

Dieser Norm-Entwurf konkretisiert einschlägige Anforderungen von Anhang I der EU-Maschinenrichtlinie 2006/42/EG an erstmals im Europäischen Wirtschaftsraum (EWR) in Verkehr gebrachte Maschinen, um den Nachweis der Übereinstimmung mit diesen Anforderungen zu erleichtern.

Die im Abschnitt 2 und den Literaturhinweisen zitierten Internationalen Normen sind, sofern diese gleichzeitig als Europäische Normen übernommen worden sind, als DIN-EN-ISO-Normen mit gleicher Zählnummer veröffentlicht.

Für die zitierten Internationalen Normen und Dokumente, sofern sie nicht als DIN-ISO-Normen mit gleicher Zählnummer veröffentlicht sind, gibt es keine nationalen Entsprechungen.

Änderungen

Gegenüber DIN EN 16590-3:2015-04 wurden folgende Änderungen vorgenommen:

a) ISO 25119-3 wurde überarbeitet und an EN 16590-3 angepasst, sodass keine modifizierte Übernahme unter anderer Normnummer notwendig ist.

Documents
Document Sharing Hub

Full document details: E DIN EN ISO 25119-3:2018-11

– Entwurf –

CEN/TC 144

Datum: 2018-07

FprEN ISO 25119-3:2018

CEN/TC 144

Sekretariat: AFNOR

Traktoren und Maschinen für die Land- und Forstwirtschaft — Sicherheitsbezogene Teile von Steuerungen — Teil 3: Serienentwicklung, Hardware und Software (ISO/FDIS 25119-3:2018)

Tracteurs et matériels agricoles et forestiers — Parties des systèmes de commande relatives à la sécurité — Partie 3 : Développement en série, matériels et logiciels (ISO/FDIS 25119-3:2018)

Tractors and machinery for agriculture and forestry — Safety-related parts of control systems — Part 3: Series development, hardware and software (ISO/FDIS 25119-3:2018)

ICS:

Deskriptoren:

Dokument-Typ: Europäische Norm
Dokument-Untertyp:
Dokumentstufe: parallele formelle Abstimmung vorgelegt
Dokumentsprache: D

STD Version 2.9d

Documents
Document Sharing Hub

Full document details: E DIN EN ISO 25119-3:2018-11

Inhalt

Seite

Europäisches Vorwort 4
Vorwort 5
Einleitung 6
1 Anwendungsbereich 8
2 Normative Verweisungen 9
3 Begriffe 9
4 Abkürzungen 9
5 Systementwurf 10
5.1 Ziele 10
5.2 Allgemeines 10
5.3 Voraussetzungen 11
5.4 Anforderungen 11
5.4.1 Gliederung der Sicherheitsanforderungen 11
5.4.2 Technisches Sicherheitskonzept 12
5.5 Arbeitsprodukte 14
6 Hardware 14
6.1 Ziele 14
6.2 Allgemeines 14
6.3 Voraussetzungen 14
6.4 Anforderungen 15
6.5 Hardwarekategorien 16
6.6 Arbeitsprodukte 17
7 Software 18
7.1 Software-Entwicklungsplanung 18
7.1.1 Ziele 18
7.1.2 Allgemeines 18
7.1.3 Voraussetzungen 18
7.1.4 Anforderungen 18
7.1.5 Arbeitsprodukte 21
7.2 Spezifikation der Sicherheitsanforderungen an die Software 21
7.2.1 Ziele 21
7.2.2 Allgemeines 21
7.2.3 Voraussetzungen 21
7.2.4 Anforderungen 22
7.2.5 Arbeitsprodukte 25
7.3 Gestaltung der Software-Architektur 25
7.3.1 Ziele 25
7.3.2 Allgemeines 26
7.3.3 Voraussetzungen 26
7.3.4 Anforderungen 26
7.3.5 Arbeitsprodukte 28
7.4 Gestaltung und Implementierung von Softwarekomponenten 28
7.4.1 Ziele 28
7.4.2 Allgemeines 28

Documents
Document Sharing Hub

Full document details: E DIN EN ISO 25119-3:2018-11

7.4.3 Voraussetzungen ... 28
7.4.4 Anforderungen ... 28
7.4.5 Arbeitsprodukte ... 40
7.5 Softwarekomponentenprüfung ... 40
7.5.1 Ziele ... 40
7.5.2 Allgemeines ... 40
7.5.3 Voraussetzungen ... 40
7.5.4 Anforderungen ... 41
7.5.5 Arbeitsprodukte ... 50
7.6 Softwareintegration und -prüfung ... 50
7.6.1 Ziele ... 50
7.6.2 Allgemeines ... 50
7.6.3 Voraussetzungen ... 50
7.6.4 Anforderungen ... 51
7.6.5 Arbeitsprodukte ... 53
7.7 Software-Sicherheitsprüfung ... 53
7.7.1 Ziele ... 53
7.7.2 Allgemeines ... 53
7.7.3 Voraussetzungen ... 53
7.7.4 Anforderungen ... 53
7.7.5 Arbeitsprodukte ... 57
7.8 Softwarebasierte Parametrierung ... 57
7.8.1 Ziel ... 57
7.8.2 Allgemeines ... 58
7.8.3 Voraussetzungen ... 58
7.8.4 Anforderungen ... 58
7.8.5 Arbeitsprodukte ... 59

Anhang A (informativ) Beispiel einer Agenda für die Beurteilung der funktionalen Sicherheit bei AgPL = e ... 60
A.1 Funktionen des Systems ... 60
A.2 Hardware ... 60
A.3 Sicherheitskonzept ... 60
A.4 Sicherheitsanalyse und Sicherheitsdaten ... 60
A.5 Sicherheitsentwurfsprozess für die Phasen des Lebenszyklus ... 61
A.6 Software-Entwicklung ... 61
A.7 Verifizierung und Prüfung ... 61
A.8 Dokumentation und Sicherheitsdokumentation ... 61
A.9 Zusammenfassung und Beurteilung ... 61

Anhang B (normativ) Unabhängigkeit durch Softwarepartitionierung ... 62
B.1 Übersicht ... 62
B.2 Begriffe und Abkürzungen ... 62
B.2.1 Begriffe ... 62
B.2.2 Abkürzungen ... 65
B.3 Ziele ... 66
B.4 Allgemeines ... 66
B.5 Anforderungen ... 66
B.5.1 Allgemeine Anforderungen ... 66
B.5.2 Mehrere Partitionen in einem einzelnen Mikrocontroller ... 67
B.5.3 Mehrere Partitionen im Rahmen eines Mikrocontroller-Netzwerks ... 69

Anhang ZA (informativ) Zusammenhang zwischen dieser Europäischen Norm und den grundlegenden Anforderungen der EU-Richtlinie 2006/42/EG ... 73

Literaturhinweise ... 74

AVA Documents
Document Sharing Hub

Full document details: E DIN EN ISO 25119-3:2018-11

Europäisches Vorwort

Dieses Dokument (FprEN ISO 25119-3:2018) wurde vom Technischen Komitee ISO/TC 23 „Tractors and Machinery for Agriculture and Forestry“ in Zusammenarbeit mit dem Technischen Komitee CEN/TC 144 „Traktoren und land- und forstwirtschaftliche Maschinen“ erarbeitet, dessen Sekretariat von AFNOR gehalten wird.

Dieses Dokument ist derzeit zur parallelen formellen Abstimmung vorgelegt.

Dieses Dokument wird EN 16590-3:2014 ersetzen.

Dieses Dokument wurde im Rahmen eines Normungsauftrages erarbeitet, den die Europäische Kommission und die Europäische Freihandelszone CEN erteilt haben, und unterstützt grundlegende Anforderungen der EU-Richtlinien.

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

Anerkennungsnotiz

Der Text von ISO/FDIS 25119-3:2018 wurde von CEN als FprEN ISO 25119-3:2018 ohne irgendeine Abänderung genehmigt.

Documents
Document Sharing Hub

Full document details: E DIN EN ISO 25119-3:2018-11

Vorwort

ISO (die Internationale Organisation für Normung) ist eine weltweite Vereinigung nationaler Normungsorganisationen (ISO-Mitgliedsorganisationen). Die Erstellung von Internationalen Normen wird üblicherweise von Technischen Komitees von ISO durchgeführt. Jede Mitgliedsorganisation, die Interesse an einem Thema hat, für welches ein Technisches Komitee gegründet wurde, hat das Recht, in diesem Komitee vertreten zu sein. Internationale staatliche und nichtstaatliche Organisationen, die in engem Kontakt mit ISO stehen, nehmen ebenfalls an der Arbeit teil. ISO arbeitet bei allen elektrotechnischen Themen eng mit der Internationalen Elektrotechnischen Kommission (IEC) zusammen.

Die Verfahren, die bei der Entwicklung dieses Dokuments angewendet wurden und die für die weitere Pflege vorgesehen sind, werden in den ISO/IEC-Direktiven, Teil 1 beschrieben. Es sollten insbesondere die unterschiedlichen Annahmekriterien für die verschiedenen ISO-Dokumentenarten beachtet werden. Dieses Dokument wurde in Übereinstimmung mit den Gestaltungsregeln der ISO/IEC-Direktiven, Teil 2 erarbeitet (siehe www.iso.org/directives).

Es wird auf die Möglichkeit hingewiesen, dass einige Elemente dieses Dokuments Patentrechte berühren können. ISO ist nicht dafür verantwortlich, einige oder alle diesbezüglichen Patentrechte zu identifizieren. Details zu allen während der Entwicklung des Dokuments identifizierten Patentrechten finden sich in der Einleitung und/oder in der ISO-Liste der erhaltenen Patenterklärungen (siehe www.iso.org/patents).

Jeder in diesem Dokument verwendete Handelsname dient nur zur Unterrichtung der Anwender und bedeutet keine Anerkennung.

Eine Erläuterung zum freiwilligen Charakter von Normen, der Bedeutung ISO spezifischer Begriffe und Ausdrücke in Bezug auf Konformitätsbewertungen, sowie Informationen darüber, wie ISO die Grundsätze der Welthandelsorganisation (WTO) hinsichtlich technischer Handelshemmnisse (TBT) berücksichtigt, enthält der folgende Link www.iso.org/iso/foreword.html.

Dieses Dokument wurde vom Technischen Komitee ISO/TC 23, *Tractors and machinery for agriculture and forestry*, Subcommittee SC 19, *Agricultural electronics* erarbeitet.

Diese zweite Ausgabe annulliert und ersetzt die erste Ausgabe (ISO 25119-1:2010), die technisch überarbeitet wurde. Die wesentlichen Änderungen im Vergleich zur Vorgängerausgabe sind folgende:

— Die Einleitung wurde geändert, um spezifische Informationen zu Sicherheitsnormen hinzuzufügen;

— Die Voraussetzungen für die Funktionssicherheit wurden festgelegt;

— Abschnitt 5 wurde geändert, um:

  — die Voraussetzungen für die Funktionssicherheit festzulegen und

  — die allgemeinen Anforderungen technischer Sicherheitskonzepte zu vereinfachen;

— Zusätzliche Anweisungen wurden im gesamten Dokument hinzugefügt, um die Konsistenz der Prüfspezifikationen und Berichte zu verifizieren;

— Anhang B wurde in einen normativen Anhang geändert;

— Das Dokument wurde redaktionell überarbeitet.

Rückmeldungen oder Fragen zu diesem Dokument sollten an die nationale Normungsorganisation des Anwenders gerichtet werden. Ein vollständiges Verzeichnis dieser Organisationen kann unter www.iso.org/members.html abgerufen werden.

AVA Documents
Document Sharing Hub

Full document details: E DIN EN ISO 25119-3:2018-11

Einleitung

ISO 25119 (alle Teile) beschreibt den Ansatz der Bewertung, Gestaltung und Verifizierung für alle Aktivitäten bezüglich des Sicherheitslebenszyklus für sicherheitsbezogene Teile, die elektrische und/oder elektronische und/oder programmierbare elektronische Systeme (E/E/PES) enthalten, welche für land- und forstwirtschaftliche Traktoren, selbstfahrende Aufsitzmaschinen sowie Anbau-, Aufsattel- und Anhängemaschinen für landwirtschaftliche Geräte verwendet werden. Sie ist auch auf mobile kommunale Geräte anwendbar.

Die Durchführung einer geeigneten Gefährdungserkennung und Risikoanalyse (z. B. ISO 12100) für die gesamte Maschine ist eine Voraussetzung für die Anwendung der Norm ISO 25119 (alle Teile). Aus diesem Grund wird häufig ein E/E/PES zugewiesen, um Sicherheitsfunktionen zu bieten, die sicherheitsbezogenes Teil einer Steuerung (SRP/CS) bilden. Diese Teile können aus Hardware und Software bestehen, können separat von der oder in die Steuerung integriert sein und entweder allein Sicherheitsfunktionen ausführen oder Teil einer Betriebsfunktion sein.

Im Allgemeinen fasst der Konstrukteur (und bis zu einem gewissen Grad der Nutzer) den Entwurf und die Validierung dieser sicherheitsbezogenen Teile als Teil der Risikobeurteilung zusammen. Dadurch soll das mit einer bestimmten Gefahr (oder Gefahrensituation) verbundene Risiko unter allen Nutzungsbedingungen der Maschine gemindert werden. Dies darf durch Anwendung verschiedener Maßnahmen (sowohl SRP/CS als auch Nicht-SRP/CS) erreicht werden, wobei das Endergebnis das Erzielen eines sicheren Zustands ist.

ISO 25119 (alle Teile) unterteilt die Fähigkeit sicherheitsbezogener Teile, eine Sicherheitsfunktion unter vorhersehbaren Bedingungen auszuführen, in fünf Stufen, sogenannte Performance Levels. Der Performance Level eines gesteuerten Kanals hängt von mehreren Faktoren ab: vom Systemaufbau (Kategorie), vom Umfang des Fehlererkennungsmechanismus (Diagnosedeckungsgrad), von der Zuverlässigkeit der Komponenten (mittlere Zeit bis zum gefahrbringenden Ausfall, Ausfälle gemeinsamer Ursache), von den Entwurfsverfahren, von der Beanspruchung im Betrieb, von Umweltbedingungen und von Arbeitsabläufen. Drei Arten von Ausfällen, die E/E/PES-Funktionsstörungen, die zu potenziellen gefahrbringenden Situationen führen, hervorrufen können, werden berücksichtigt: systematische Ausfälle, Ausfälle gemeinsamer Ursache und Zufallsausfälle.

Darüber hinaus legt ISO 25119 (alle Teile) einen Ansatz auf Basis einer Klassifikation der Architektur mit verschiedenen Konstruktionsmerkmalen und einem bestimmten Verhalten im Fehlerfall fest, um den Konstrukteur während der Gestaltung und Verifizierung anzuleiten und die Beurteilung des erreichten Performance Levels zu erleichtern.

Die Performance Level und Kategorien lassen sich anwenden auf Steuerungen aller Arten fahrbarer Maschinen: von einfachen Systemen (z. B. Anlassventilen) bis hin zu komplexen Systemen (z. B. Steer-by-Wire) sowie Steuerungen von Schutzeinrichtungen (z. B. Verriegelungsvorrichtungen, druckempfindliche Einrichtungen).

ISO 25119 (alle Teile) übernimmt für die Ermittlung der Risiken einen auf dem Risiko basierenden Ansatz zur Angabe des Soll-Performance-Levels für die von den E/E/PES-sicherheitsrelevanten Kanälen auszuführenden Sicherheitsfunktionen. Sie liefert die mit den Performance Levels verbundenen Anforderungen für den gesamten E/E/PES-Lebenszyklus (Entwicklung, Fertigung, Betrieb, Wartung, Außerbetriebnahme), die zur Erzielung der erforderlichen Funktionssicherheit für E/E/PES und der zugehörigen Performance Levels notwendig sind.

Documents
Document Sharing Hub

Full document details: E DIN EN ISO 25119-3:2018-11

Sicherheitsnormen im Bereich Maschinenbau gliedern sich in folgende Bereiche:

a) Typ-A-Normen (Grundnormen) enthalten Grundbegriffe, Gestaltungsleitsätze und allgemeine Aspekte, die für alle Maschinen angewendet werden können.

b) Typ-B-Normen (Fachgrundnormen) behandeln einen oder mehrere Sicherheitsaspekte oder eine oder mehrere Arten von Schutzeinrichtungen, die bei einer Reihe von Maschinen angewendet werden können:

   — Typ-B1-Normen behandeln bestimmte Sicherheitsaspekte (z. B. Sicherheitsabstände, Oberflächentemperatur, Geräusch).

   — Typ-B2-Normen behandeln Schutzeinrichtungen (z. B. Zweihandschaltungen, Verriegelungen, Kontaktmatten, trennende Schutzeinrichtungen).

c) Typ-C-Normen (Maschinensicherheitsnormen) behandeln detaillierte Sicherheitsanforderungen für eine bestimmte Maschine oder Gruppe von Maschinen.

Dieses Dokument ist eine Typ-B1-Norm wie in ISO 12100 angegeben.

Dieses Dokument ist insbesondere für die folgenden Akteurgruppen als Marktteilnehmer in Bezug auf die Maschinensicherheit relevant:

— Maschinenhersteller (kleine, mittelständische und große Unternehmen);

— Gesundheits- und Sicherheitsinstitutionen (Regulierungsbehörden, Unfallverhütungs-Organisationen, Marktaufsicht usw.).

Andere Gruppen, die von dem mit den Mitteln des Dokuments von den oben genannten Akteurgruppen erreichten Grad an Maschinensicherheit betroffen sein können:

— Maschinenbetreiber/Arbeitgeber (kleine, mittlere und große Unternehmen);

— Maschinenbetreiber/Angestellte (z. B. Gewerkschaften, Organisationen für Menschen mit besonderen Bedürfnissen);

— Dienstleister, für Instandhaltung (kleine, mittlere und große Unternehmen);

— Endkunde (im Fall von Maschinen, die für die Verwendung durch den Endkunden vorgesehen sind).

Den vorgenannten Interessengruppen wurde die Möglichkeit gegeben, an dem Entwurfsprozess zu diesem Dokument teilzunehmen.

Des Weiteren ist dieses Dokument für Normungsinstitute gedacht, die Typ-C-Normen erarbeiten.

Die Anforderungen in diesem Dokument können durch eine Typ-C-Norm ergänzt oder modifiziert werden.

Für Maschinen, die vom Anwendungsbereich einer Typ-C-Norm behandelt werden und entsprechend den Anforderungen dieser Norm gestaltet und erstellt wurden, haben die Anforderungen dieser Typ-C-Norm Vorrang.

AVA Documents
Document Sharing Hub

Full document details: E DIN EN ISO 25119-3:2018-11

1 Anwendungsbereich

Dieses Dokument stellt allgemeine Grundlagen und Leitsätze heraus für die Gestaltung und Entwicklung sicherheitsbezogener Teile von Steuerungen (SRP/CS) in land- und forstwirtschaftlichen Traktoren, selbstfahrenden Aufsitzmaschinen sowie Anbau-, Aufsattel- und Anhängemaschinen für landwirtschaftliche Geräte. Sie kann auch auf fahrbare kommunale Geräte (z. B. Straßenkehrmaschinen) angewendet werden.

Dieses Dokument gilt nicht für:

— Flugzeuge und Luftkissenfahrzeuge für den landwirtschaftlichen Einsatz;

— Rasen- und Gartengeräte.

Dieses Dokument legt die Merkmale und Kategorien von SRP/CS-Systemen fest, die zur Ausführung von Sicherheitsfunktionen erforderlich sind. Es enthält keine Performance Level für bestimmte Anwendungen.

ANMERKUNG 1 Maschinenspezifische Typ-C-Normen können Performance Level (AgPL) für Sicherheitsfunktionen von Maschinen in ihrem Anwendungsbereich festlegen. Andernfalls ist der Hersteller für die Festlegung des AgPl verantwortlich.

Dieses Dokument gilt für sicherheitsbezogene Teile elektrischer/elektronischer/programmierbarer elektronischer Systeme (E/E/PES) als Teil mechatronischer Systeme. Er behandelt die möglichen Gefahren durch Fehlfunktionen von sicherheitsbezogenen E/E/PES einschließlich der Wechselwirkungen dieser Systeme. Das Dokument behandelt nicht Gefahren in Verbindung mit Stromschlag, Feuer, Rauch, Hitze, Strahlung, Toxizität, Entflammbarkeit, Reaktivität, Korrosion, Freisetzung von Energie usw. soweit diese nicht direkt durch eine Fehlfunktion der sicherheitsbezogenen Systeme von E/E/PES verursacht werden. Darüber hinaus behandelt das Dokument die Fehlfunktionen sicherheitsbezogener Systeme von E/E/PES, die an Schutzmaßnahmen, Schutzeinrichtungen oder Sicherheitsfunktionen als Reaktion auf nicht durch E/E/PES verursachte Gefahren beteiligt sind.

Im Anwendungsbereich dieses Dokuments enthaltene Beispiele sind:

— sicherheitsbezogene Teile einer Steuerung (SRP/CS), die den Stromfluss in elektrischen Hybridsystemen begrenzen, um Isolierungsausfälle und Gefahren durch elektrischen Schlag zu verhindern;

— elektromagnetische Störungen mit dem SRP/CS;

— für Brandschutzzwecke gestaltetet SRP/CS.

Im Anwendungsbereich dieses Dokuments nicht enthaltene Beispiele sind:

— Isolierungsausfälle durch Reibung, die zu der Gefahr eines elektrischen Schlags führen;

— elektromagnetische Nennstrahlung, die nahegelegene Maschinensteuerungssysteme beeinflusst;

— Korrosion, die zu einer Überhitzung von Stromkabeln führt.

Dieses Dokument ist nicht anzuwenden auf nicht-E/E/PES-Systeme (z. B. hydraulische, mechanische und pneumatische).

ANMERKUNG 2 Siehe auch ISO 12100 zu Gestaltungsleitsätzen für die Sicherheit von Maschinen.

Dieses Dokument gilt nicht für sicherheitsbezogene Teile von Steuerungen, die vor dem Veröffentlichungsdatum dieses Dokuments hergestellt wurden.

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 25119-3:2018-11

2 Normative Verweisungen

Die folgenden Dokumente werden im Text in solcher Weise in Bezug genommen, dass einige Teile davon oder ihr gesamter Inhalt Anforderungen des vorliegenden Dokuments darstellen. 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 25119-1, *Tractors and machinery for agriculture and forestry — Safety-related parts of control systems — Part 1: General principles for design and development*

ISO 25119-2:2018, *Tractors and machinery for agriculture and forestry — Safety-related parts of control systems — Part 2: Concept phase*

ISO 25119-4:2018, *Tractors and machinery for agriculture and forestry — Safety-related parts of control systems — Part 4: Production, operation, modification and supporting processes*

3 Begriffe

Für die Anwendung dieses Dokuments gelten die Begriffe nach ISO 25119-1.

ISO und IEC stellen terminologische Datenbanken für die Verwendung in der Normung unter den folgenden Adressen bereit:

— ISO Online Browsing Platform: unter https://www.iso.org/obp

— IEC Electropedia: unter http://www.electropedia.org/

4 Abkürzungen

Für die Anwendung dieses Dokuments gelten die folgenden Abkürzungen.

AgPL	landwirtschaftlicher Performance Level (en: agricultural performance level)
$AgPL_r$	geforderter landwirtschaftlicher Performance Level (en: required agricultural performance level)
CAD	computergestützte Gestaltung (en: computer-aided design)
Kat	Hardwarekategorie
CCF	Ausfall infolge gemeinsamer Ursache (en: common-cause failure)
DC	Diagnosedeckungsgrad (en: diagnostic coverage)
DC_{avg}	durchschnittlicher Diagnosedeckungsgrad (en: average diagnostic coverage)
ECU	elektronische Steuereinheit (en: electronic control unit)
ETA	Ereignisbaumanalyse (en: event tree analysis)
E/E/PES	elektrisches/elektronisches/programmierbares elektronisches System (en: electrical/electronic/programmable electronic systems)
EMV	elektromagnetische Verträglichkeit (en: EMC, electromagnetic compatibility)
FMEA	Ausfallart- und einflussanalyse (en: Failure Mode and Effects Analysis)
FSM	Funktionssicherheitsmanagement (en: functional safety management)
FTA	Fehlerbaumanalyse (en: Fault Tree Analysis)
HARA	Gefahrenanalyse und Risikobeurteilung

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 25119-3:2018-11

HIL	Hardware in the Loop
MTTF	mittlere Zeit bis zum Ausfall (en: mean time to failure)
$MTTF_D$	mittlere Zeit bis zum gefahrbringenden Ausfall (en: mean time to dangerous failure)
$MTTF_{DC}$	mittlere Zeit bis zum gefahrbringenden Ausfall für jeden Kanal (en: mean time to dangerous failure for each channel)
PES	programmierbares elektronisches System (en: programmable electronic system)
QM	Qualitätssicherungsmaßnahmen (en: quality measures)
RAM	Arbeitsspeicher (en: random-access memory)
SOP	Produktionsanlauf (en: start of production)
SRL	Anforderungsniveau an die Software (en: software requirement level)
SRP	sicherheitsbezogene Teile (en: safety-related parts)
SRP/CS	sicherheitsbezogene Teile von Steuerungen (en: safety-related parts of control systems)
UML	vereinheitlichte Modellierungssprache (en: unified modelling language)

5 Systementwurf

5.1 Ziele

Ziel ist die Definition einer Systemgestaltung zur Herstellung eines Entwurfs, der die Sicherheitsanforderungen an das sicherheitsbezogene Gesamtsystem erfüllt.

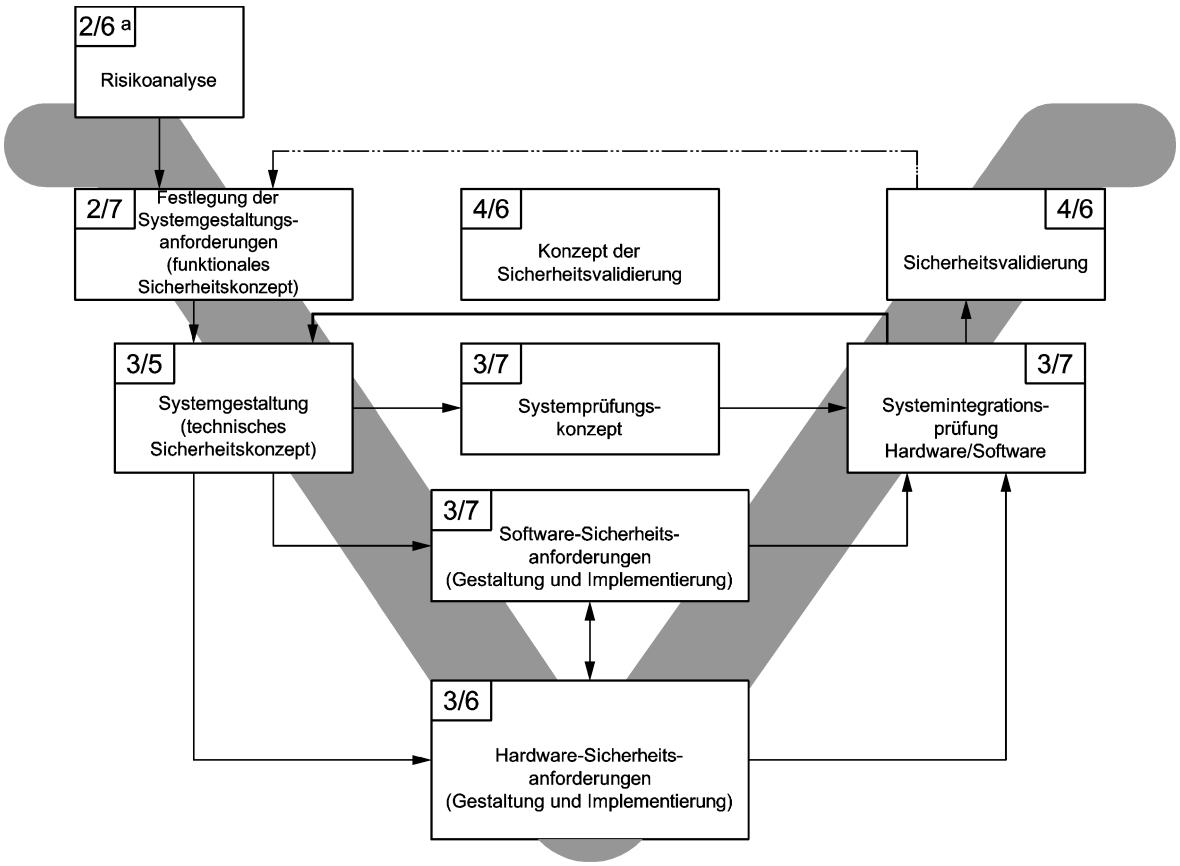
5.2 Allgemeines

Sicherheitsanforderungen machen alle Anforderungen aus, die das Erreichen und Sicherstellen der funktionalen Sicherheit zum Ziel haben. Während des Sicherheitslebenszyklus werden die Sicherheitsanforderungen in immer näheren Einzelheiten in hierarchischen Ebenen detailliert und spezifiziert. Die verschiedenen Ebenen der Sicherheitsanforderungen sind in Bild 1 dargestellt. Für eine Gesamtdarstellung des Verfahrens zur Entwicklung der Sicherheitsanforderungen siehe auch 5.4 dieser Norm. Zur Unterstützung des Managements der Sicherheitsanforderungen wird die Anwendung geeigneter Werkzeuge für das Anforderungsmanagement empfohlen.

Anhang A bietet eine Anleitung für ein Beispielprogramm für die Beurteilung der Funktionssicherheit in Verbindung mit einem AgPL = e.

Documents
Document Sharing Hub

Full document details: E DIN EN ISO 25119-3:2018-11



Legende

→ Folge

← Verifizierung

←··· Validierung

a Die erste der beiden durch einen Schrägstrich getrennten Zahlen bezieht sich auf den jeweiligen Teil der ISO 25119 und die zweite Zahl bezieht sich auf den Abschnitt in diesem Dokument: d. h. 2/6 ist ISO 25119-2:2018/Abschnitt 6, 3/5 ist ISO 25119-3:2018/Abschnitt 5 usw.

Bild 1 — Gliederung der Sicherheitsanforderungen

5.3 Voraussetzungen

Das funktionale Sicherheitskonzept (ISO 25119-2:2018, Abschnitt 7) ist die Voraussetzung.

5.4 Anforderungen

5.4.1 Gliederung der Sicherheitsanforderungen

Während der funktionalen Sicherheitskonzeptphase werden funktionale Sicherheitsanforderungen erstellt, um die grundlegende Funktion des Sicherheitssystems, mit dem die Sicherheitsziele erfüllt werden müssen, zu beschreiben. Die Grundzuordnung der funktionalen Sicherheitsanforderungen zur Systemarchitektur wird durch das technische Sicherheitskonzept in Form technischer Sicherheitsanforderungen festgelegt. Diese Systemarchitektur umfasst Hardware und Software.

AVA Documents
Document Sharing Hub

Full document details: E DIN EN ISO 25119-3:2018-11

Die Sicherheitsanforderungen an die Hardware verfeinern und konsolidieren die technischen Sicherheitsanforderungen. Abschnitt 6 beschreibt, wie die Hardwareanforderungen im Einzelnen festzulegen sind.

Die Sicherheitsanforderungen an die Software sind von den Anforderungen des technischen Sicherheitskonzepts und der zugrunde liegenden Hardware abgeleitet. Die in Abschnitt 7 festgelegten Softwareanforderungen sind zu berücksichtigen.

Dieser Abschnitt enthält Richtlinien für den bei der Festlegung der Anforderungen des technischen Sicherheitskonzepts während des Systementwurfs anzuwendenden Ansatz. Diese Verfahrensrichtlinien liefern eine Grundlage für einen fehlerfreien Systementwurf.

5.4.2 Technisches Sicherheitskonzept

5.4.2.1 Allgemeine Anforderungen des technischen Sicherheitskonzepts

Das Dokument des technischen Sicherheitskonzepts enthält die technischen Sicherheitsanforderungen an das System.

Jeder technischen Sicherheitsanforderung müssen (z. B. durch Querverweis) übergeordnete Sicherheitsanforderungen zugeordnet werden. Übergeordnete Sicherheitsanforderungen können sein:

— andere technische Sicherheitsanforderungen;

— funktionale Sicherheitsanforderungen.

ANMERKUNG Die Rückverfolgbarkeit kann durch Nutzung geeigneter Anforderungsmanagementwerkzeuge sehr erleichtert werden.

Bei der Implementierung jeder Anforderung des technischen Sicherheitskonzepts muss auf deren Durchführbarkeit, Eindeutigkeit, Konsistenz und Vollständigkeit geachtet werden.

Das technische Sicherheitskonzept muss folgende Quellen berücksichtigen:

a) alle Sicherheitsziele und funktionalen Sicherheitsanforderungen;

b) alle einschlägigen Normen, Standards und gesetzlichen Vorschriften;

c) die entsprechenden Ergebnisse von Sicherheitsanalysewerkzeugen (FMEA, FTA usw.); die Sicherheitsanalyse bietet iterative Unterstützung für das technische Sicherheitskonzept während der Systementwicklung.

Die Vollständigkeit des technischen Sicherheitskonzepts nimmt während des Systementwurfs iterativ zu. Um die Vollständigkeit sicherzustellen:

1) müssen die Version des technischen Sicherheitskonzepts und die Version der entsprechenden zugrunde liegenden Quellen angegeben werden;

2) müssen die Anforderungen aus dem Änderungsmanagement (siehe ISO 25119-4:2018, Abschnitt 11) erfüllt werden; daher müssen die technischen Sicherheitsanforderungen so aufgegliedert und formuliert werden, dass ein Änderungsprozess unterstützt wird;

3) müssen die technischen Sicherheitsanforderungen überprüft werden (siehe ISO 25119-4:2018, Abschnitt 6).

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 25119-3:2018-11

Beim technischen Sicherheitskonzept müssen alle Phasen des Lebenszyklus (einschließlich Produktion, Betrieb durch den Kunden, Kundendienst und Außerbetriebnahme) betrachtet werden.

5.4.2.2 Spezifikation des technischen Sicherheitskonzepts

5.4.2.2.1 Allgemeines

Das technische Sicherheitskonzept muss Sicherheitsanforderungen an Hardware und Software enthalten, die für den Entwurf der SRP/CS ausreichend und von den in 5.4.2.1 enthaltenen Regeln bestimmt sind.

5.4.2.2.2 Zustände und Zeiten

Das Verhalten des SRP/CS und seiner Komponenten sowie deren Schnittstellen müssen für alle relevanten Betriebszustände spezifiziert werden, einschließlich

— Anlauf,

— Normalbetrieb,

— Stillsetzung,

— Wiederanlauf nach einer Rückstellung und

— vernünftigerweise vorhersehbare ungewöhnliche Betriebszustände (z. B. herabgesetzte Betriebszustände).

Insbesondere müssen das Ausfallverhalten und die erforderliche Reaktion darauf genau beschrieben werden. Zusätzliche Notbetriebsfunktionen können aufgenommen werden.

Das technische Sicherheitskonzept muss einen sicheren Zustand für jede funktionale Sicherheitsanforderung, den Übergang in den sicheren Zustand sowie die Aufrechterhaltung des sicheren Zustands spezifizieren. Insbesondere ist anzugeben, ob ein Ausschalten des SRP/CS direkt einen sicheren Zustand darstellt oder ob ein sicherer Zustand nur durch eine kontrollierte Abschaltung erreicht werden kann.

Im technischen Sicherheitskonzept muss für jede funktionale Sicherheitsanforderung die maximale Zeit zwischen dem Auftreten eines Fehlers und dem Erreichen eines sicheren Zustands (Fehlertoleranzzeit) angegeben werden. Alle daraus abgeleiteten Zeiten für Teilsysteme und Teilfunktionen müssen in den Anforderungen des Sicherheitskonzepts angegeben werden.

Kann durch eine direkte Abschaltung kein sicherer Zustand erreicht werden, muss die Zeit definiert werden, während der eine als Notbetrieb angegebene Sonderfunktion für alle Teilsysteme und Teilfunktionen aufrechterhalten werden muss (Notbetriebszeit). Diese als Notbetrieb angegebene Sonderfunktion muss in den Anforderungen des Sicherheitskonzepts dokumentiert werden.

5.4.2.2.3 Sicherheitsarchitektur, Schnittstellen und Randbedingungen

Die Sicherheitsarchitektur und ihre Teilkomponenten müssen beschrieben werden. Insbesondere müssen die technischen Maßnahmen angegeben werden. Im technischen Sicherheitskonzept müssen folgende Komponenten (falls vorhanden) separat beschrieben werden:

— Sensorsystem, separat für jeden erfassten physischen Parameter;

— verschiedene digitale und analoge Ein- und Ausgabeeinheiten;

— Verarbeitung, separat für jedes Rechenwerk/jede diskrete Logikeinheit;

Documents
Document Sharing Hub

Full document details: E DIN EN ISO 25119-3:2018-11

— Aktorsystem, separat für jeden Aktor;

— Anzeigen, separat für jede Anzeigeeinheit;

— verschiedene elektromechanische Bauteile;

— Signalübertragung zwischen den Komponenten;

— Signalübertragung von/zu Systemen außerhalb des SRP/CS;

— Stromquelle.

Die Schnittstellen zwischen den Komponenten des SRP/CS, Schnittstellen zu anderen Systemen und Funktionen in der Maschine sowie Benutzerschnittstellen müssen angegeben werden.

Beschränkungen und Randbedingungen des SRP/CS müssen angegeben werden. Dies gilt insbesondere für Extremwerte aller Umgebungsbedingungen in allen Phasen des Lebenszyklus.

5.5 Arbeitsprodukte

Die folgenden Arbeitsprodukte müssen auf die Systemgestaltung anwendbar sein:

a) Spezifikation des technischen Sicherheitskonzepts.

6 Hardware

6.1 Ziele

Ziel ist es, akzeptable Hardware-Architekturen für SRP/CS festzulegen.

6.2 Allgemeines

Die Verbesserung des Hardware-Aufbaus des SRP/CS kann Maßnahmen zur Vermeidung, Erkennung oder Tolerierung von Fehlern liefern. Zu den praktischen Maßnahmen können Redundanz, Diversität und Überwachung zählen.

Im Allgemeinen müssen folgende Fehlerkriterien berücksichtigt werden.

— Wenn infolge eines Fehlers weitere Bauteile ausfallen, müssen der erste Fehler und alle Folgefehler als Einzelfehler betrachtet werden.

— Zwei oder mehr separate Fehler mit gemeinsamer Ursache müssen als Einzelfehler betrachtet werden (dies ist als Ausfall infolge gemeinsamer Ursache bekannt).

— Das gleichzeitige Auftreten von zwei unabhängigen Fehlern ist äußerst unwahrscheinlich und muss daher nicht berücksichtigt werden.

Iterationen zwischen der Hardware- und Software-Entwicklung können erforderlich sein, um die notwendige Hardware-Prüfung durchzuführen.

6.3 Voraussetzungen

Die Voraussetzung besteht in den durch die Hardware zu realisierenden Teilen des funktionalen Sicherheitskonzepts (ISO 25119-2:2018, Abschnitt 7).

Die Voraussetzung für die in Bild 2 für durch Software gesteuerte Hardware dargestellte „Hardwaresystemintegration“ und „Hardwaresicherheitsvalidierung“ ist die Betriebssoftware selbst.

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 25119-3:2018-11

6.4 Anforderungen

Der Hardware-Entwicklungsprozess muss auf der Systemebene beginnen, auf der Sicherheitsfunktionen und die damit verbundenen Anforderungen identifiziert werden (siehe Bild 2).

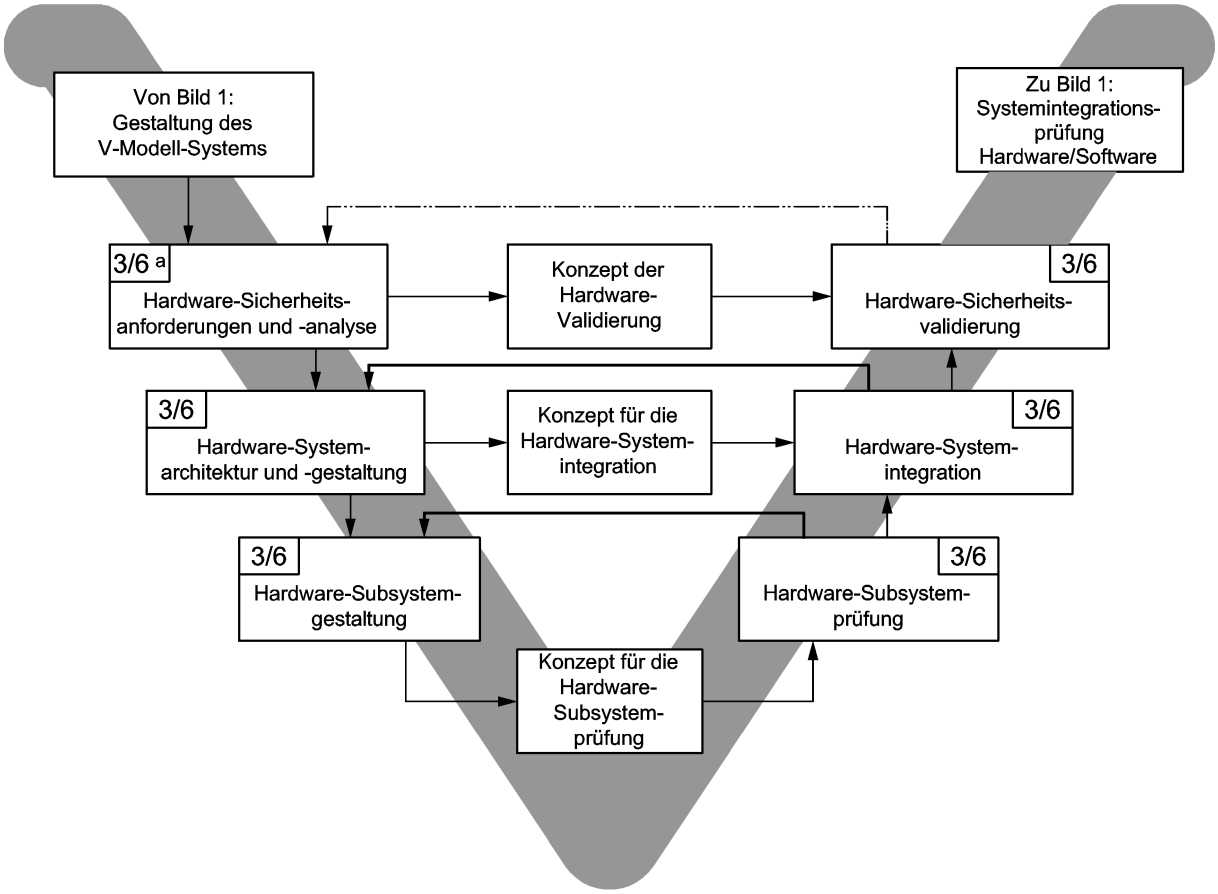
Das funktionale Sicherheitskonzept muss für die Identifizierung des Performance Levels (AgPL) für jede Sicherheitsfunktion des Systems angewendet werden (siehe ISO 25119-2).

Die Auswahl von Hardware-Kategorien, $MTTF_{DC}$, DC und SRL muss so erfolgen, dass der resultierende AgPL des einzelne oder der Kombination von SRP/CS alle $AgPL_r$ der zugeordneten funktionalen Sicherheitsanforderungen erfüllt oder übertrifft.

Das System kann in Teilsysteme zerlegt werden, um die Entwicklung zu vereinfachen.

Jede Phase des Entwicklungszyklus muss validiert werden.

Die in Bild 1 dargestellte Prüfung der Hardware/Software-Systemintegration muss unter Anwendung zuvor geprüfter Hardware- und Softwarekomponenten durchgeführt werden.



Legende

——→ Folge

←—— Verifizierung

←—·—— Validierung

a Die erste der beiden durch einen Schrägstrich getrennten Ziffern verweist auf dieses Dokument und die zweite Ziffer bezeichnet den Abschnitt 6.

Bild 2 — V-Modell der Hardwareentwicklung

AVA Documents
Document Sharing Hub

Full document details: E DIN EN ISO 25119-3:2018-11

Das Verfahren für die Gestaltung der Hardware-Architektur stellt sich wie folgt dar.

a) Eine Hardware-Kategorie wird unter Berücksichtigung der verfügbaren $MTTF_{DC}$, SRL und DC ausgewählt (siehe ISO 25119-2:2018, Bild 2).

b) Identifizierung der Betriebsumgebung und der Beanspruchungshöhe von Bauteilen.

c) Auswahl der Bauteile.

d) Berechnung und Verifizierung, ob die $MTTF_{DC}$ der geforderten Höhe entspricht (siehe ISO 25119-2:2018, 7.3.3).

e) Berechnung und Verifizierung, ob die DC der geforderten Höhe entspricht (siehe ISO 25119-2:2018, 7.3.4).

f) Betrachtung von CCF (siehe ISO 25119-2:2018, Anhang D).

g) Betrachtung der systematischen Ausfälle (siehe ISO 25119-2:2018, Anhang E).

h) Betrachtung anderer Sicherheitsfunktion (siehe ISO 25119-2:2018, Anhang F).

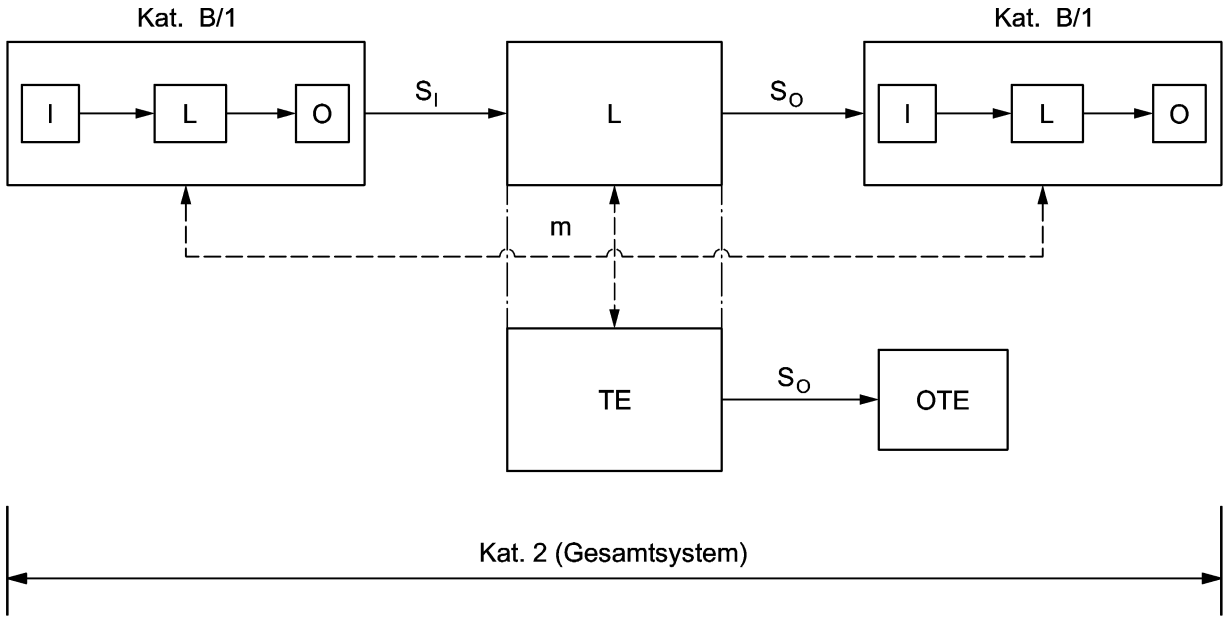
Für die vorstehenden Schritte ist eventuell eine Wiederholung erforderlich.

6.5 Hardwarekategorien

Die sicherheitsbezogenen Teile von Steuerungen müssen in Übereinstimmung mit den Anforderungen einer oder mehrerer der fünf in ISO 25119-2:2018, Anhang A, beschriebenen Kategorien unter Berücksichtigung von ISO 25119-2:2018, Anhänge H, I und J gestaltet werden.

Wenn eine Sicherheitsfunktion durch eine integrierte Kombination verschiedener Hardwarekategorien erfüllt wird, wird die resultierende allgemeine Hardwarekategorie durch die Eigenschaften des Gesamtsystems bestimmt (siehe Bild 3).

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 25119-3:2018-11



Legende

I	Eingabegerät (z. B. Sensor)	S_I	Verbundene Signaleingabe
L	Logik	S_O	Verbundene Signalausgabe
O	Ausgabegerät (z. B. Aktor)	m	Überwachung
TE	Prüfeinrichtung	Kat	Hardwarekategorie
OTE	Ausgang des Prüfgeräts		

Bild 3 — Beispiel eines integrierten Systems für die Hardwarekategorie 2

Zur Bestimmung des Gesamt-SRL siehe 7.3.4.7.

6.6 Arbeitsprodukte

Die folgenden Arbeitsprodukte müssen auf die Hardwaregestaltung anwendbar sein:

a) Prüfplan für die Sicherheitsvalidierung der Hardware;

b) Prüfspezifikation für die Sicherheitsvalidierung der Hardware;

c) Prüfergebnisse der Sicherheitsvalidierung der Hardware;

d) Hardware-Systemintegrationstestplan/Hardware-Teilsystemeprüfplan;

e) Prüfspezifikation für die Hardware-Systemintegration;

f) Hardware-Systemintegrations-Prüfergebnisse/Hardware-Teilsystem-Prüfergebnisse;

g) Hardwarearchitekturgestaltung, die die Anforderungen der geforderten Kategorie und des AgPL erfüllt;

h) Hardware-Sicherheitsanforderungen.

AVA Documents
Document Sharing Hub

Full document details: E DIN EN ISO 25119-3:2018-11

7 Software

7.1 Software-Entwicklungsplanung

7.1.1 Ziele

Ziel ist es, die einzelnen Phasen der Software-Entwicklung festzulegen und zu planen. Dies umfasst den Prozess der Software-Entwicklung selbst, der in diesem Abschnitt beschrieben wird, sowie alle notwendigen unterstützenden Prozesse nach ISO 25119-4:2018, Abschnitt 11.

7.1.2 Allgemeines

Bild 4 veranschaulicht den Prozess zur Entwicklung der Software. In den folgenden Absätzen und Tabellen wird jeder Kasten des Schemas näher erläutert.

Geeignete Methoden/Maßnahmen müssen entsprechend dem erforderlichen SRL gewählt werden. Angesichts der Vielzahl von Faktoren, welche die Sicherheitsintegrität der Software beeinflussen, kann kein Algorithmus für das Kombinieren der Methoden und Maßnahmen angegeben werden, welcher für jede mögliche Anwendung richtig ist. Für eine bestimmte Anwendung ist die passende Kombination von Methoden oder Maßnahmen im Laufe der Software-Sicherheitsplanung anzugeben, wobei geeignete Methoden oder Maßnahmen entsprechend den Anforderungen von 7.1.4 auszuwählen sind.

7.1.3 Voraussetzungen

Die Voraussetzungen in dieser Phase sind

— der erforderliche SRL wie durch den AgPL, $MTTF_{DC}$, DC und die Kategorie aus dem funktionalen Sicherheitskonzept bestimmt,

— der Systemverifizierungsplan,

— das technische Sicherheitskonzept und

— die funktionalen Sicherheitsanforderungen und die Sicherheitsziele.

7.1.4 Anforderungen

7.1.4.1 Bestimmung der Phasen

Für den Software-Entwicklungsprozess muss festgelegt werden, wie Phasen der Software-Entwicklung (siehe Bild 4) auszuführen sind. Umfang und Komplexität des Projekts müssen berücksichtigt werden. Die Phasen dürfen ohne Modifikation entsprechend Bild 4 ausgeführt werden. Es können jedoch auch einzelne Phasen miteinander kombiniert werden, sobald alle Arbeitsprodukte der kombinierten Phasen generiert sind.

ANMERKUNG Es ist üblich, einzelne Phasen zu kombinieren, wenn das verwendete Verfahren eine klare Unterscheidung zwischen den Phasen erschwert. So können beispielsweise der Entwurf der Software-Architektur und die Software-Implementierung nacheinander mit demselben rechnergestützten Entwicklungswerkzeug generiert werden, wie dies im modellbasierten Entwicklungsprozess der Fall ist.

Weitere Phasen dürfen durch Aufteilung der Aktivitäten und Aufgaben hinzugefügt werden.

BEISPIEL Die Anwendung von Daten kann als separate Phase vor der Sicherheitsvalidierung der elektronischen Steuereinheit eingefügt werden. Die Sicherheitsvalidierung des elektronischen Steuergeräts kann je nach Verteilung der Funktionen unterschiedlich durchgeführt werden — als Prüfung bestimmter elektronischer Steuergeräte oder als Prüfung des kombinierten Steuerungsnetzwerks. So könnte sie am Prüfort der Bauteilsysteme oder im Laborfahrzeug durchgeführt werden.

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 25119-3:2018-11

7.1.4.2 Prozessflexibilität

Die Aktivitäten und Aufgaben können von einer Phase zu einer anderen verlagert werden.

7.1.4.3 Prozesszeitplan

Es muss ein Zeitplan erstellt werden, welcher die Beziehung zwischen den einzelnen Phasen der Software-Entwicklung und dem Produktentwicklungsprozess einschließlich der Integrationsschritte auf Maschinenebene aufzeigt.

7.1.4.4 Anwendbarkeit

Nachdem die Sicherheitsanforderungen an die Software nach Tabelle 1 vollständig spezifiziert wurden, muss festgelegt werden, welche Sicherheitsanforderungen an die Software auf welche Prozessaktivitäten anzuwenden sind.

7.1.4.5 Unterstützende Verfahren

Die folgenden unterstützenden Prozesse müssen als Teil des Softwareentwicklungsprozesses geplant werden:

a) Durchführen einer Risikoanalyse nach ISO 25119-4:2018, Abschnitt 13;

b) Software-Änderungsmanagement nach ISO 25119-4:2018, Abschnitt 11;

c) Arbeitsprodukt-Konfigurationsmanagement.

ANMERKUNG Der unterstützende Prozess b) beinhaltet eine Strategie für die Behandlung der verschiedenen Teile der Software, die durch Änderungen entstehen, einschließlich der Zusammenführung dieser Teile.

7.1.4.6 Phasen der Software-Entwicklung

Für jede Phase der Software-Entwicklung müssen die entsprechenden Entwicklungsverfahren und Maßnahmen, die entsprechenden Tools und die Richtlinien für die Implementierung der Verfahren, Maßnahmen und Tools in Übereinstimmung mit dem SRL (z. B. McCabe-Metrik, Prüfumfang, Fehlerraten) ausgewählt werden.

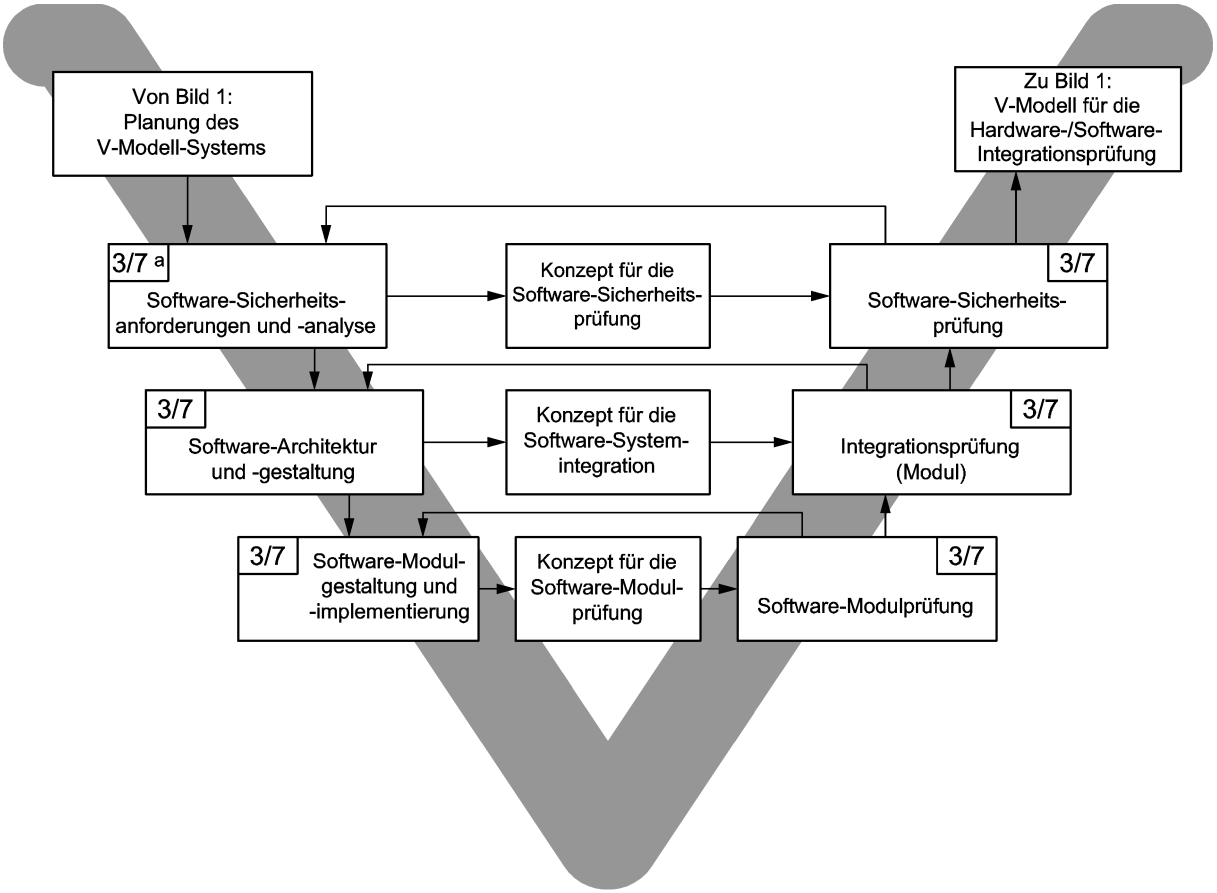
Diese Auswahl muss hinsichtlich ihrer Eignung für das Anwendungsgebiet begründet werden, und sie muss zu Beginn jeder Entwicklungsphase durchgeführt werden.

Bei der Auswahl der Methoden und Maßnahmen muss daran gedacht werden, dass neben der manuellen Codierung eine modellbasierte Entwicklung durchgeführt werden darf, bei der der Quellcode oder der Objektcode automatisch aus Modellen generiert wird.

ANMERKUNG Die Auswahl koordinierter Verfahren und Maßnahmen bietet die Möglichkeit, die Komplexität der Software-Entwicklung zu reduzieren.

AVA Documents
Document Sharing Hub

Full document details: E DIN EN ISO 25119-3:2018-11



Legende

——► Folge

◄—— Verifizierung

a Die erste der beiden durch einen Schrägstrich getrennten Ziffern verweist auf dieses Dokument und die zweite Ziffer bezeichnet den Abschnitt 7

Bild 4 — V-Modell der Software-Entwicklung

7.1.4.7 Anwendung der Tabellen

Die Tabellen 1 bis 6 geben für jedes Entwicklungsverfahren einen Eintrag für jeden der vier SRL mit dem Symbol „+“, „o“ oder „x“ an:

— \+ das Verfahren oder die Maßnahme ist für diesen SRL anzuwenden, sofern keine Gründe dagegen sprechen; in diesem Fall muss der Grund während der Planungsphase dokumentiert werden;

— o es gibt keine Empfehlung zugunsten der oder gegen die Anwendung dieses Verfahrens oder diese Maßnahme für diesen SRL;

— x das Verfahren oder die Maßnahme ist nicht für die Erfüllung dieses SRL geeignet.

Es müssen die dem jeweiligen SRL entsprechenden Verfahren und Maßnahmen gewählt werden. Alternative oder gleichwertige Verfahren und Maßnahmen sind durch Buchstaben nach der Zahl gekennzeichnet. Es muss mindestens eine(s) der alternativen oder gleichwertigen Verfahren und Maßnahmen gewählt werden, die mit einem „+“ gekennzeichnet ist.

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 25119-3:2018-11

Ist eine spezielle Methode oder Maßnahme nicht in den Tabellen aufgeführt, so ist nicht anzunehmen, dass sie nicht berücksichtigt werden darf. Wird eine nicht aufgeführte Methode oder Maßnahme durch eine andere ersetzt, so muss diese eine gleiche oder höhere Wertigkeit haben.

7.1.5 Arbeitsprodukte

Das auf diese Phase anwendbare Arbeitsprodukt ist der Software-Projektplan entsprechend den Anforderungen in 7.1.1 bis 7.1.4.6 (siehe auch ISO 25119-4:2018, Abschnitt 6).

7.2 Spezifikation der Sicherheitsanforderungen an die Software

7.2.1 Ziele

Das erste Ziel besteht in der Herleitung der Sicherheitsanforderungen an die Software einschließlich SRL aus den technischen Sicherheitsanforderungen.

Das zweite Ziel besteht in der Verifizierung, dass die Sicherheitsanforderungen an die Software dem technischen Sicherheitskonzept entsprechen.

7.2.2 Allgemeines

Die Spezifikation der Sicherheitsanforderungen an die Software muss von den Anforderungen des technischen Sicherheitskonzepts für das System abgeleitet und als Software-Sicherheitsanforderungen gekennzeichnet werden. Mindestens folgende Punkte müssen berücksichtigt werden:

a) ausreichende Implementierung des technischen Sicherheitskonzepts in der Software;

b) Systemkonfiguration und -architektur;

c) Gestaltung der Hardware des E/E/PES-Systems;

d) Ansprechzeiten der Sicherheitsfunktionen;

e) externe Schnittstellen wie z. B. Kommunikationsschnittstellen;

f) physische Anforderungen und Umgebungsbedingungen, soweit sie die Software beeinflussen;

g) Anforderungen an eine sichere Software-Modifikation.

Zwischen der Hardware-Entwicklung des Systems und der Software-Entwicklung sind Wiederholungen erforderlich. Während des Prozesses der weiteren Spezifizierung und Detaillierung der Sicherheitsanforderungen an die Software und der Software-Architektur kann es zu Rückwirkungen auf die Hardware-Architektur kommen. Daher muss eine enge Zusammenarbeit zwischen der Hardware-Entwicklung und der Software-Entwicklung erfolgen.

7.2.3 Voraussetzungen

Die folgenden Voraussetzungen werden an die Spezifikation der Sicherheitsanforderungen an die Software gestellt:

— Software-Projektplan nach 7.1.1 bis 7.1.4.6;

— technisches Sicherheitskonzept nach 5.4.2;

— Hardwarekategorien nach 6.5.

AVA Documents
Document Sharing Hub

Full document details: E DIN EN ISO 25119-3:2018-11

7.2.4 Anforderungen

7.2.4.1 Spezifikationsmethoden für Sicherheitsanforderungen an die Software

Die Spezifikation der Sicherheitsanforderungen an die Software muss nach den in Tabelle 1 genannten Methoden/Maßnahmen erfolgen.

Tabelle 1 — Spezifikation der Software-Sicherheitsanforderungen

Methoden/Maßnahme[a]	Unterabschnitt	SRL = B	SRL = 1	SRL = 2	SRL = 3
1 Spezifikation der Anforderungen in natürlicher Sprache	7.2.4.1.1	+	+	+	+
2a Informelle Verfahren[a, b]	7.2.4.1.2	+	+	x	x
2b Halbformelle Verfahren[b]	7.2.4.1.3	+	+	+	+
2c Formelle Verfahren[b]	7.2.4.1.4	+	+	+	+
3 Rechnergestützte Spezifikationswerkzeuge	7.2.4.1.5	o	o	+	+
4 Inspektion der Software-Sicherheitsanforderungen[a]	7.2.4.7	+	+	+	+
4b Walkthrough der Sicherheitsanforderungen an die Software	7.2.4.7	+	+	x	x

ANMERKUNG 1 Wege zur Modellverarbeitung mit einer vollständigen Syntaxdefinition und einer vollständigen semantischen Definition mit Berechnung sind in Punkt 2c zusammengefasst. Formale Methoden ermöglichen eine formale Verifizierung und eine automatische Testfallerzeugung. Beispiele sind unter anderem mit einer formalen Verifizierung verbundene „endliche Automaten“.

ANMERKUNG 2 Wege zur Modellverarbeitung mit einer vollständigen Syntaxdefinition und einer vollständigen semantischen Definition ohne Berechnung sind in Punkt 2b zusammengefasst. Beispiele sind unter anderem strukturierte Analyse/Entwurf und graphische Modelliermethoden wie z. B. UML-Klassen- oder Blockdiagramme. Siehe Anweisungen aus 7.1.4.7 für die Verwendung der Tabellen 1 bis 6.

[a] Es müssen geeignete Methoden/Maßnahmen entsprechend dem SRL gewählt werden. Alternative oder gleichwertige Methoden/Maßnahmen sind durch einen Buchstaben nach der Zahl gekennzeichnet. Nur eine der Methoden/Maßnahmen muss erfüllt sein.

[b] Im Fall einer modellbasierten Entwicklung mit Code-Generierung müssen die Methoden und Maßnahmen für den Entwurf der Software-Architektur auf das funktionale Modell angewendet werden, welches als Grundlage für die Code-Generierung dient.

7.2.4.1.1 Spezifikation der Anforderungen in natürlicher Sprache

7.2.4.1.1.1 Ziel

Die Spezifikationsanforderungen müssen in natürlicher Sprache (in Wort und Schrift) vorgestellt werden.

7.2.4.1.1.2 Beschreibung

Die Spezifikation der Sicherheitsanforderungen an die Software muss stets eine Beschreibung des Problems in natürlicher Sprache sowie, falls erforderlich, weitere informelle Verfahren, z. B. Bilder und Diagramme, umfassen.

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 25119-3:2018-11

7.2.4.1.2 Informelle Verfahren

7.2.4.1.2.1 Ziel

Die Darstellung eines vollständigen Konzepts muss in natürlicher Sprache erfolgen.

7.2.4.1.2.2 Beschreibung

Informelle Verfahren liefern ein Mittel, um eine Beschreibung des Systems in der entsprechende Phase seiner Entwicklung zu erstellen, d. h. Spezifikation, Entwurf oder Konstruktion mithilfe von natürlicher Sprache, Diagrammen, Bildern usw.

7.2.4.1.3 Halbformelle Verfahren

7.2.4.1.3.1 Ziel

Halbformelle Verfahren müssen ein Konzept, eine Spezifikation, eine Gestaltung oder eine Konstruktion eindeutig und konsequent darstellen, sodass Fehler und Auslassungen erkannt werden können.

7.2.4.1.3.2 Beschreibung

Halbformelle Verfahren der Software-Implementierung müssen angewendet werden, um ein Mittel zur Entwicklung einer Systembeschreibung in einer entsprechenden Phase ihrer Entwicklung wie etwa Spezifikation, Gestaltung oder Konstruktion zu liefern.

BEISPIEL Datenflussdiagramme und „Endliche-Automaten-/Zustandsübergangsdiagramme".

Gegebenenfalls muss die Beschreibung maschinell analysiert oder zur Anzeige verschiedener Aspekte des Systemverhaltens animiert werden. Die Animation schafft zusätzliches Vertrauen, dass das System die Anforderungen auch erfüllt.

7.2.4.1.4 Formelle Verfahren

7.2.4.1.4.1 Ziel

Die Entwicklung der Software auf einer mathematischen Grundlage muss formelle Konzeptions-, Spezifikations-, Gestaltungs- und Konstruktionstechniken umfassen.

7.2.4.1.4.2 Beschreibung

Formelle Verfahren liefern ein Mittel, um eine Beschreibung des Systems in einer entsprechenden Phase seiner Entwicklung zu erstellen, d. h. Spezifikation, Entwurf oder Implementierung. Die daraus resultierende Beschreibung muss in strikter Notation erfolgen, die einer mathematischen Analyse unterzogen werden kann, um verschiedene Klassen von Inkonsistenzen oder Fehlern zu erkennen. Darüber hinaus muss die Beschreibung gegebenenfalls maschinell mit einer Genauigkeit analysiert werden, die der Syntaxprüfung eines Quellprogramms durch einen Compiler ähnlich ist, oder animiert werden, um verschiedene Aspekte des Verhaltens des beschriebenen Systems darzustellen. Die Animation schafft zusätzliches Vertrauen, dass das System die formal spezifizierte Anforderung erfüllt, weil sie das Erkennen des spezifizierten Verhaltens durch den Menschen erleichtert.

Ein formelles Verfahren muss allgemein eine Notation (im Allgemeinen wird eine Art diskreter Mathematik verwendet), ein Methode zur Herleitung einer Beschreibung in dieser Notation sowie mehrere Analyseformen zur Prüfung einer Beschreibung auf verschiedene Korrektheitseigenschaften bieten.

Documents
Document Sharing Hub

Full document details: E DIN EN ISO 25119-3:2018-11

7.2.4.1.5 Rechnergestützte Spezifikationswerkzeuge

7.2.4.1.5.1 Ziel

Formelle Spezifikationsmethoden müssen angewendet werden, um das automatische Erkennen von Mehrdeutigkeiten und Vollständigkeit zu erleichtern.

7.2.4.1.5.2 Beschreibung

Diese Methode muss eine Spezifikation in einem elektronischen Format liefern, das die Inspektion zur Bewertung der Konsistenz und Vollständigkeit unterstützt. Im Allgemeinen unterstützt diese Methode nicht nur die Erstellung der Spezifikation, sondern auch des Entwurfs und anderer Lebenszyklusphasen des Projekts.

7.2.4.2 Nicht sicherheitsbezogene Funktionen

Werden neben den Sicherheitsfunktionen andere Funktionen vom E/E/PES-System ausgeführt, müssen diese Funktionen spezifiziert werden, oder es muss auf ihre Spezifikationen Bezug genommen werden.

Enthält die Spezifikation der Anforderungen sowohl die nicht sicherheitsbezogenen als auch die sicherheitsbezogenen Anforderungen, so müssen die letzteren eindeutig als solche gekennzeichnet sein.

7.2.4.3 Detailtiefe

Die Spezifikation der Sicherheitsanforderungen an die Software muss ausreichend detailliert sein, damit die Sicherheitsfunktion in die Software implementiert werden kann.

7.2.4.4 Konsistenz

Es ist eine bidirektionale Verfolgbarkeit zwischen Software-Sicherheitsanforderungen und technischen Sicherheitsanforderungen zu realisieren.

7.2.4.5 Abhängigkeiten zwischen Hardware und Software

Die Spezifikation der Sicherheitsanforderungen an die Software muss, falls zutreffend, die sicherheitsbezogenen Abhängigkeiten zwischen der Hardware und der Software spezifizieren.

7.2.4.6 Spezifikation der Sicherheitsanforderungen an die Software

Die Spezifikation der Software-Sicherheitsanforderungen muss, falls zutreffend, die Sicherheitsanforderungen an die Software für Folgendes beschreiben:

— Funktionen, die es dem System ermöglichen, einen sicheren Zustand zu erreichen oder aufrechtzuerhalten;

— Funktionen zur Erkennung, Anzeige und Handhabung von Fehlern im elektronischen Steuergerät, den Sensoren, Aktoren und im Kommunikationssystem;

— Funktionen zur Erkennung, Anzeige und Handhabung von Fehlern in der Software selbst (Selbstüberwachung der Software).

ANMERKUNG 1 Dies beinhaltet sowohl die Selbstüberwachung der Software im Betriebssystem als auch eine anwendungsspezifische Selbstüberwachung der Software, die systematische Fehler in der Anwendungssoftware erkennen soll.

— Funktionen für Online- und Offline-Tests der Sicherheitsfunktionen.

Documents
Document Sharing Hub

Full document details: E DIN EN ISO 25119-3:2018-11

ANMERKUNG 2 Selbstprüfungen können im Betrieb und beim Anlassen des Fahrzeugs durchgeführt werden.

ANMERKUNG 3 Dies bezieht sich insbesondere auf die Überprüfbarkeit der Sicherheitsfunktionen im Kundendienst oder durch andere E/E/PES-Systeme.

— Funktionen, die eine sichere Durchführung von Software-Modifikationen ermöglichen.

— Schnittstellen mit nicht sicherheitsbezogenen Funktionen.

— Leistung und Ansprechzeit.

— Schnittstellen zwischen Software und Hardware des elektronischen Steuergeräts.

ANMERKUNG 4 Die Schnittstellen beinhalten auch Programmierung und Konfiguration.

Die Spezifikation der Software-Sicherheitsanforderungen muss Folgendes enthalten:

a) den SRL für jede der vorstehend aufgeführten Funktionen;

b) die Abnahmekriterien für die Software-Sicherheitsvalidierung der Sicherheitsanforderungen an die Software.

7.2.4.7 Verifizierung der Sicherheitsanforderungen an die Software

Die Sicherheitsanforderungen an die Software müssen verifiziert werden. Es muss geprüft werden, ob sie den in 7.2.4.1 bis 7.2.4.6 genannten Anforderungen entsprechen. Die Sicherheitsanforderungen an die Software müssen ebenfalls verifiziert werden, um zu prüfen, ob sie mit dem technischen Sicherheitskonzept übereinstimmen. Die Prüfspezifikation und der Prüfbericht nach Tabelle 6 müssen ebenfalls geprüft werden, um zu verifizieren, ob diese mit der Software-Sicherheitsanforderung konsistent sind. Die Software-Entwickler müssen an den Verifizierungsaktivitäten teilnehmen. Zwei Arten von Verifizierungsverfahren sind möglich: Inspektion oder Walkthrough (wie in ISO 25119-1 festgelegt).

7.2.5 Arbeitsprodukte

Folgende Arbeitsprodukte sind auf diese Phase anwendbar:

a) Spezifikation der Software-Sicherheitsanforderungen nach 7.2.4.1 und 7.2.4.3 bis 7.2.4.6;

b) Spezifikation der nicht sicherheitsbezogenen Software-Anforderungen nach 7.2.4.2;

c) Verifizierungsbericht über die Spezifikation der Sicherheitsanforderungen an die Software, resultierend aus den Anforderungen nach 7.2.4.7.

7.3 Gestaltung der Software-Architektur

7.3.1 Ziele

Ziel der Software-Architektur ist es, alle Software-Anforderungen durch die Anwendung von Softwarekomponenten, die bestimmen, wie die Software implementiert werden muss, zu strukturieren. Die Software-Architektur muss sicherstellen, dass alle Sicherheitsanforderungen an die Software von den ihnen zugeordneten Softwarekomponenten erfüllt werden.

AVA Documents
Document Sharing Hub

Full document details: E DIN EN ISO 25119-3:2018-11

7.3.2 Allgemeines

Die Software-Architektur stellt alle Softwarekomponenten und ihre wechselseitigen Beziehungen in einer hierarchischen Struktur dar. Die statischen Aspekte wie Schnittstellen und Datenpfade aller Softwarekomponenten sowie die dynamischen Aspekte wie Prozessabläufe und zeitliches Verhalten müssen beschrieben werden.

7.3.3 Voraussetzungen

Ein ausreichender Reifegrad in der Spezifikation der Software-Sicherheitsanforderungen ist eine Voraussetzung für den Beginn der Entwicklung der Software-Architektur.

7.3.4 Anforderungen

7.3.4.1 Verfahren zur Gestaltung der Software-Architektur

Die Software-Architektur muss entwickelt werden nach Tabelle 2.

Tabelle 2 — Gestaltung der Software-Architektur

Methoden/Maßnahme[a]		Unterabschnitt	SRL = B	SRL = 1	SRL = 2	SRL = 3
1a	Informelle Verfahren[a]	7.2.4.1.2	+	+	x	x
1b	Halbformelle Verfahren	7.2.4.1.3	+	+	+	+
1c	Formelle Verfahren	7.2.4.1.4	+	+	+	+
2	Rechnergestützte Spezifikationswerkzeuge	7.2.4.1.5	o	o	+	+
3a	Inspektion der Software-Architektur[a]	7.3.4.6	+	+	+	+
3b	Walkthrough der Software-Architektur	7.3.4.6	+	+	x	x

Siehe Anweisungen aus 7.1.4.7 für die Verwendung der Tabellen 1 bis 6.

[a] Es müssen geeignete Methoden/Maßnahmen entsprechend dem SRL gewählt werden. Alternative oder gleichwertige Methoden/Maßnahmen sind durch einen Buchstaben nach der Zahl gekennzeichnet. Nur eine der Methoden/Maßnahmen muss erfüllt sein

7.3.4.2 Merkmale des Gestaltungsverfahrens

Das gewählte Gestaltungsverfahren muss Merkmale aufweisen, die Folgendes unterstützen:

a) Abstraktion, Modularität, Kapselung und weitere Merkmale, welche die Komplexität handhabbar machen;

b) die Beschreibung der:

— Funktionalität;

— den Informationsfluss zwischen den Komponenten;

— die Prozesssteuerung und Informationen über die Zeit;

— zeitliche Beschränkungen;

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 25119-3:2018-11

— gleichzeitig ablaufende Prozesse, falls zutreffend;

— Datenstrukturen und ihre Merkmale;

— Annahmen für die Gestaltung und ihre Abhängigkeiten;

c) das Verständnis der Entwickler und anderer beteiligter Parteien;

d) die Fähigkeit zur Software-Modifikation;

e) Prüfung.

7.3.4.3 Struktur der Software-Architektur

Es muss eine Software-Architektur entwickelt werden, die die hierarchische Struktur, basierend auf den Sicherheitsanforderungen an die Software, für alle Sicherheits-Softwarekomponenten beschreibt.

ANMERKUNG Auf der obersten Ebene der Software-Architektur findet in der Regel eine Trennung in Systemsoftware (z. B. Board-Unterstützungspakete, Betriebssysteme, Hardware-Abstraktionsebene) und Anwendungssoftware statt.

7.3.4.4 Detailtiefe

Die hierarchische Struktur der Software-Architektur muss mit den Software-Komponenten auf der unteren Ebene enden.

Bei der Entwicklung der Software-Architektur sollte der Umfang der sicherheitsbezogenen Softwarekomponenten so gering wie möglich gehalten werden.

7.3.4.5 Verfolgbarkeit der Software-Architektur

Es ist eine bidirektionale Verfolgbarkeit zwischen Software-Architektur und Software-Sicherheitsanforderungen zu realisieren.

7.3.4.6 Verifizierung der Software-Architektur

Die Software-Architektur muss verifiziert werden. Es muss geprüft werden, ob die entworfene Architektur die Sicherheitsanforderungen an die Software erfüllt. Die Prüfspezifikation und der Prüfbericht nach Tabelle 5 müssen ebenfalls geprüft werden, um zu verifizieren, ob diese mit der Software-Sicherheitsarchitektur konsistent sind. Die Software-Entwickler müssen an den Verifizierungsaktivitäten teilnehmen. Zwei Arten von Verifizierungsverfahren sind möglich: Inspektion oder Walkthrough (wie in ISO 25119-1) und bei Bedarf (siehe Tabelle 2).

7.3.4.7 Kombination sicherheitsbezogener Softwarekomponenten

Wenn die systemintegrierte Software Softwarekomponenten mit unterschiedlichem SRL oder sicherheitsbezogene und nicht sicherheitsbezogene Softwarekomponenten implementieren muss, dann muss das Gesamt-SRL auf die Softwarekomponente mit dem niedrigsten SRL begrenzt werden, sofern nicht eine ausreichende Unabhängigkeit zwischen den Softwarekomponenten in Übereinstimmung mit den Anforderungen der Steuerung nachgewiesen werden kann. SRL 2 oder 3 müssen die Anforderungen in Anhang B erfüllen.

Documents
Document Sharing Hub

Full document details: E DIN EN ISO 25119-3:2018-11

7.3.5 Arbeitsprodukte

Folgende Arbeitsprodukte sind auf diese Phase anwendbar:

a) Software-Architektur nach 7.3.4.1 bis 7.3.4.5;

b) ein Software-Architektur-Verifizierungsbericht, der sich aus 7.3.4.6 ergibt.

7.4 Gestaltung und Implementierung von Softwarekomponenten

7.4.1 Ziele

Das erste Ziel ist die detaillierte Spezifizierung des Verhaltens der sicherheitsbezogenen Softwarekomponenten, die von der Softwarearchitektur vorgeschrieben sind.

Das zweite Ziel ist die Generierung eines lesbaren, prüfbaren und wartungsfreundlichen Quellcodes (Code, Modell usw.), der zur Umsetzung in Objektcode geeignet ist.

Das dritte Ziel ist die Verifizierung, dass die Software-Architektur vollständig und richtig implementiert wurde.

7.4.2 Allgemeines

7.4.3 Voraussetzungen

Die folgenden Voraussetzungen für Gestaltung und Implementierung von Softwarekomponenten sind zu erfüllen:

— Software-Projektplan (siehe 7.1.1 bis 7.1.4.6);

— Software-Anforderungen [siehe 7.2.5 a) und b)];

— Software-Architektur (siehe 7.3.4.1 bis 7.3.4.5);

— Software-Verifizierungsplan (siehe ISO 25119-4:2018, Abschnitt 6).

7.4.4 Anforderungen

7.4.4.1 Gestaltungs- und Implementierungsverfahren für Softwarekomponenten

Software-Entwurf und -Entwicklung müssen den Methoden/Maßnahmen aus Tabelle 3 entsprechen.

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 25119-3:2018-11

Tabelle 3 — Software-Entwurf und -Entwicklung — Unterstützungswerkzeuge und Programmiersprache

Methoden/Maßnahme[a]		Unterabschnitt	SRL = B	SRL = 1	SRL = 2	SRL = 3
1	**Werkzeuge und Programmiersprache**					
1.1	Geeignete Programmiersprache	7.4.4.1.1	+	+	+	+
1.2	Streng typisierte Programmiersprache	7.4.4.1.2	o	+	+	+
1.3	Sprachteilmenge	7.4.4.1.3	o	+	+	+
1.4	Werkzeuge und Übersetzungsprogramme — Betriebsbewährtheit	7.4.4.1.4	o	+	+	+
1.5	Verwendung vertrauenswürdiger/verifizierter Softwarekomponenten (falls vorhanden)	7.4.4.1.5	o	o	+	+
2	**Entwurfsmethoden**					
2.1a	Informelle Verfahren[a]	7.2.4.1.2	+	+	x	x
2.1b	Halbformelle Verfahren	7.2.4.1.3	+	+	+	+
2.1c	Formelle Verfahren	7.2.4.1.4	+	+	+	+
2.2	Defensive Programmierung	7.4.4.1.6	o	o	o	+
2.3	Strukturierte Programmierung	7.4.4.1.7	o	+	+	+
2.4	Modularisierung					
2.4.1	Größenbeschränkung von Softwarekomponenten	7.4.4.1.8	o	+	+	+
2.4.2	Software-Komplexitätskontrolle	7.4.4.1.9	o	o	o	+
2.4.3	Verbergen/Kapseln von Informationen	7.4.4.1.10	o	o	+	+
2.4.4	Ein Eintritts-/Austrittspunkt in Unterprogrammen und Funktionen	7.4.4.1.8	o	+	+	+
2.4.5	Vollständig festgelegte Schnittstelle	7.4.4.1.8	o	+	+	+
2,5	Bibliothek vertrauenswürdiger/verifizierter Softwarekomponenten	7.4.4.1.11	+	+	+	+
2.6	Rechnergestützte Entwurfswerkzeuge	7.4.4.1.12	o	o	o	+
3	**Entwurfs- und Codierungsstandard**					
3.1	Verwendung von Codierungsstandards	7.4.4.1.13	o	+	+	+
3.2a	Keine dynamischen Variablen oder Objekte	7.4.4.1.14	o	o	o	+
3.2b	Online-Prüfung der Erstellung dynamischer Variablen	7.4.4.1.15	o	o	o	+
3.3.	Eingeschränkte Verwendung von Interrupts	7.4.4.1.16	o	o	o	+

Documents
Document Sharing Hub

Full document details: E DIN EN ISO 25119-3:2018-11

	Methoden/Maßnahme[a]	Unterabschnitt	SRL = B	SRL = 1	SRL = 2	SRL = 3
3.4	Festgelegte Verwendung von Zeigern	7.4.4.1.17	o	o	o	+
3.5	Eingeschränkte Verwendung von Rekursionen	7.4.4.1.18	o	o	o	+
4	**Entwurfs- und Codierungsverifizierung**					
4a	Inspektion der Software-Gestaltung und/oder des Quellcodes[a]	7.4.4.2	+	+	+	+
4b	Walkthrough der Software-Architektur und/oder des Quellcodes	7.4.4.2	+	+	x	x

Siehe Anweisungen aus 7.1.4.7 für die Verwendung der Tabellen 1 bis 6.

[a] Es müssen geeignete Methoden/Maßnahmen entsprechend dem SRL gewählt werden. Alternative oder gleichwertige Methoden/Maßnahmen sind durch einen Buchstaben nach der Zahl gekennzeichnet. Nur eine der Methoden/Maßnahmen muss erfüllt sein.

7.4.4.1.1 Geeignete Programmiersprache

7.4.4.1.1.1 Ziel

Ziel ist die Wahl einer geeigneten Programmiersprache zur Unterstützung der Anforderungen der ISO 25119 (alle Teile), soweit praktikabel, insbesondere der Anforderungen an die defensive Programmierung, strukturierte Programmierung und eventuell Plausibilitätskontrollen. Die gewählte Programmiersprache muss zu einem einfach verifizierbaren Code führen und die Entwicklung, Verifizierung und Wartung des Programms erleichtern.

7.4.4.1.1.2 Beschreibung

Die Sprache muss vollständig und eindeutig definiert sein. Die Sprache muss anwender- oder problemorientiert anstatt prozessor-/plattformmaschinenorientiert sein. Weit verbreitete Sprachen (z. B. C, C++, Ada) oder ihre Teilsprachen werden Spezialsprachen vorgezogen.

Neben den bereits genannten Eigenschaften müssen die Sprache oder die Teildatensätze Folgendes bereitstellen:

— Blockstruktur;

— Translation Time Type Checking.

Die Sprache muss Folgendes unterstützen:

— Verwendung kleiner und handhabbarer Softwarekomponenten (siehe 7.4.4.1.8);

— Beschränkung des Zugriffs auf Daten in speziellen Softwarekomponenten;

— Festlegung von Variablen-Teilbereichen;

— jede andere Art von fehlerbegrenzenden Konstrukten.

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 25119-3:2018-11

Wenn der sichere Betrieb des Systems von Echtzeitbeschränkungen abhängig ist, muss die Sprache auch Fehler-/Interruptbehandlung vorsehen. Es ist wünschenswert, dass die Sprache von einem geeigneten Übersetzungsprogramm (z. B. Compiler, Assembler, Linker und Locator), entsprechenden Bibliotheken bereits vorhandener Softwarekomponenten, einem Diagnoseprogramm und Werkzeugen sowohl für Versionskontrolle als auch für Entwicklung unterstützt wird. Eigenschaften, die die Verifizierung erschweren und daher vermieden werden müssen, sind:

a) unbedingte Programmsprünge, außer Unterprogrammaufrufe;

b) Rekursion;

c) übermäßige Nutzung dynamischer Variablen oder Objekte, Datenhaufen oder Zeiger;

d) Handhabung von Unterbrechungen auf Anwendungsebene;

e) Mehrfachein- oder -aussprünge aus Schleifen, Blöcken oder Unterprogrammen;

f) implizite Variableninitialisierung oder -deklaration;

g) Variantendatensätze und Äquivalenz;

h) Verfahrensparameter.

ANMERKUNG Niedere Programmiersprachen, insbesondere Assemblersprachen, stellen wegen ihrer prozessor-/plattformmaschinenorientierten Art Probleme dar. Eine wünschenswerte Eigenschaft der Sprache ist, dass ihr Entwurf und ihre Verwendung zu Programmen führen, deren Ausführung vorhersagbar ist. Eine geeignete definierte Programmiersprache vorausgesetzt, gibt es eine Sprachteilmenge, die die Vorhersagbarkeit der Programmausführung sicherstellt. Diese Sprachteilmenge kann (im Allgemeinen) nicht statisch bestimmt werden, auch wenn zur Sicherstellung der vorhersagbaren Ausführung viele statische Beschränkungen hilfreich sein können. Dies erfordert in der Regel einen Nachweis, dass die Feldindizes sich innerhalb der Datenfeldgrenzen befinden und dass beispielsweise kein numerischer Überlauf stattfinden kann.

7.4.4.1.2 Streng typisierte Programmiersprache oder Richtlinienüberwachung

7.4.4.1.2.1 Ziel

Die Verringerung der Wahrscheinlichkeit von Fehlern wird durch Verwendung einer Sprache oder eines Programmierverfahrens, die ein hohes Maß an Überwachung durch den Compiler oder ein statisches Analysewerkzeug zulassen, erreicht.

7.4.4.1.2.2 Beschreibung

Beim Übersetzen oder bei der statischen Analyse einer streng typisierten Programmiersprache werden zahlreiche Kontrollen über die Art der Verwendung variabler Typen durchgeführt (z. B. in Prozeduraufrufen und externem Datenzugriff). Für jede Verwendung, die nicht vordefinierten Regeln entspricht, muss das Übersetzen fehlschlagen und eine Fehlermeldung ausgegeben werden.

ANMERKUNG Derartige Sprachen ermöglichen normalerweise die Definition von benutzerdefinierten Datentypen aus den Datentypen der niederen Sprache (wie z. B. ganzzahlig, reell). Diese Typen können dann auf genau dieselbe Weise wie der niedere Typ verwendet werden. Strenge Kontrollen werden auferlegt, um sicherzustellen, dass der korrekte Typ verwendet wird. Diese Kontrollen gelten für das gesamte Programm, selbst wenn dieses aus separat übersetzten Einheiten aufgebaut ist. Die Kontrollen stellen auch sicher, dass die Anzahl und Art der Prozedurargumente zusammenpassen, auch wenn darauf von separat übersetzten Softwarekomponenten verwiesen wird.

AVA Documents
Document Sharing Hub

Full document details: E DIN EN ISO 25119-3:2018-11

7.4.4.1.3 Sprachteilmenge

7.4.4.1.3.1 Ziel

Die Verwendung einer Sprachteilmenge führt zur Verringerung der Wahrscheinlichkeit, dass Programmierfehler eingeführt werden, und erhöht die Wahrscheinlichkeit, dass alle übrigen Fehler erkannt werden.

7.4.4.1.3.2 Beschreibung

Die Sprache muss zwecks Bestimmung von Programmierkonstrukten, die entweder undefiniert, fehleranfällig oder schwer zu analysieren sind, geprüft werden (z. B. mithilfe statischer Analysemethoden). Diese Programmierkonstrukte müssen anschließend ausgeschlossen und ein Sprachteilsatz muss definiert werden. Des Weiteren muss dokumentiert werden, warum diese in der Sprachteilmenge verwendeten Konstrukte sicher sind.

7.4.4.1.4 Werkzeuge und Übersetzungsprogramme — Betriebsbewährtheit

7.4.4.1.4.1 Ziel

Nur bewährte Werkzeuge und Übersetzungsprogramme dürfen zur Anwendung kommen und führen damit zur Vermeidung von Schwierigkeiten durch Fehler von Übersetzungsprogrammen, die bei der Entwicklung, Verifizierung und Wartung eines Softwarepakets entstehen können.

7.4.4.1.4.2 Beschreibung

Ein Übersetzungsprogramm wird verwendet, wenn bei zahlreichen früheren Projekten keine unkorrekte Ausführung nachgewiesen wurde. Übersetzungsprogramme ohne Betriebserfahrung oder mit bekannten schwerwiegenden Fehlern müssen vermieden werden, außer wenn es irgendeine andere Zusicherung der korrekten Ausführung gibt. Hat das Übersetzungsprogramm kleine Mängel aufgewiesen, so werden die zugehörigen Sprachkonstrukte identifiziert und im Verlauf eines sicherheitsbezogenen Projekts sorgfältig vermieden.

ANMERKUNG 1 Diese Empfehlung basiert auf den Erfahrungen aus zahlreichen Projekten. Es hat sich erwiesen, dass unausgereifte Übersetzungsprogramme ein schwerwiegender Nachteil für jede Software-Entwicklung sind und die Entwicklung einer sicherheitsbezogenen Software im Allgemeinen undurchführbar machen.

ANMERKUNG 2 Es ist festzustellen, dass es zurzeit keine Methode zum Nachweis der Korrektheit für alle Werkzeug- oder Übersetzungsprogrammteile gibt.

7.4.4.1.5 Verwendung vertrauenswürdiger/verifizierter Softwarekomponenten (falls vorhanden)

7.4.4.1.5.1 Ziel

Die Verwendung vertrauenswürdiger/verifizierter Softwarekomponenten vermeidet die Notwendigkeit einer umfangreichen erneuten Validierung oder Umarbeitung der Softwarekomponenten oder der Gestaltung von Hardware-Bauteilen für jede neue Anwendung. Dies versetzt den Entwickler in die vorteilhafte Lage, Gestaltungen verwenden zu können, die zwar nicht formal oder streng verifiziert wurden, für die aber eine umfangreiche Betriebshistorie existiert.

7.4.4.1.5.2 Beschreibung

Diese Maßnahme verifiziert, dass die Softwarekomponenten ausreichend frei von systematischen Konstruktionsfehlern und/oder Betriebsausfällen sind. Nur in seltenen Fällen wird der Einsatz vertrauenswürdiger Softwarekomponenten (d. h. die sich im Gebrauch bewährt haben) als alleinige Maßnahme ausreichen, um das Erreichen der notwendigen Sicherheitsintegrität (SRL) sicherzustellen. Bei komplexen Softwarekomponenten mit vielen möglichen Funktionen (z. B. ein Betriebssystem) ist es

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 25119-3:2018-11

notwendig festzustellen, welche Funktionen sich im Gebrauch tatsächlich ausreichend bewährt haben. So kann beispielsweise eine Selbsttestroutine zur Erkennung von Hardwarefehlern zur Verfügung stehen, jedoch nicht als im Gebrauch bewährt angesehen werden, wenn während der Betriebszeit kein Hardwareausfall auftritt.

Eine Softwarekomponente ist nur ausreichend vertrauenswürdig, wenn sie bereits auf den geforderten SRL verifiziert wurde oder folgende Kriterien erfüllt:

— unveränderte Spezifikation mit Auswirkung auf die Sicherheitsfunktion (z. B. Änderungen der Entscheidungspunkte und Nachschlagetabellen sind akzeptabel, sofern durch Verifizierung nachgewiesen wird, dass die Sicherheitsfunktion nicht betroffen war);

— mindestens ein Jahr Instandhaltungsdaten;

— die gesamte Betriebserfahrung muss sich auf ein bekanntes Anforderungsprofil der Funktionen der Softwarekomponente beziehen, um sicherzustellen, dass eine längere Betriebserfahrung (z. B. in Systemen in verschiedenen Anwendungen) wirklich zu einer besseren Kenntnis der Softwarekomponente hinsichtlich dieses Anforderungsprofils führt;

— keine Ausfälle von Sicherheitsfunktionen.

ANMERKUNG 1 Der Ausfall einer nicht sicherheitsbezogenen Funktion in einer Anwendung kann ein Ausfall einer Sicherheitsfunktion in einer anderen Anwendung sein und umgekehrt.

Um verifizieren zu können, dass eine Komponente oder Softwarekomponente die Kriterien erfüllt, muss Folgendes dokumentiert werden:

a) genaue Bezeichnung jedes Systems und seiner Komponenten einschließlich der in dem Verifizierungsprozess verwendeten Versionsnummern (sowohl für Software als auch für Hardware);

b) Bezeichnung einer statistisch signifikanten Benutzerprobe und der Dauer der Anwendung (z. B. Betriebsjahre);

c) Verfahren zur Auswahl der statistisch signifikanten Probe;

d) Verfahren zur Erkennung und Registrierung von Ausfällen und zur Fehlerbeseitigung.

ANMERKUNG 2 Die Anwendung dieses Verfahrens ist für gesamte SW-Systeme möglich (auch durch Lieferanten gelieferte Systeme), wenn die geforderten Kriterien erfüllt und die erforderlichen Nachweise dokumentiert werden.

7.4.4.1.6 Defensive Programmierung

7.4.4.1.6.1 Ziel

Defensive Programmierung muss angewendet werden für das Erstellen von Programmen, welche einen anormalen Steuerungsablauf, Datenfluss oder Datenwerte während ihrer Ausführung erkennen und darauf auf vorherbestimmte und akzeptable Art und Weise reagieren.

AVA Documents
Document Sharing Hub

Full document details: E DIN EN ISO 25119-3:2018-11

7.4.4.1.6.2 Beschreibung

Während des Programmierens können viele Methoden zur Überwachung auf Steuerungs- oder Datenanomalien angewendet werden. Diese müssen systematisch während der gesamten Programmierung eines Systems angewendet werden, um die Wahrscheinlichkeit einer fehlerhaften Datenverarbeitung zu verringern. Inhärent fehlersichere Software ist so gestaltet, dass sie ihre eigenen Entwurfsmängel beseitigt. Diese Mängel können durch Fehler beim Entwurf oder bei der Codierung oder durch fehlerhafte Anforderungen verursacht sein. Es gibt zwei sich überlappende Bereiche defensiver Methoden. Der erste Satz defensiver Methoden umfasst:

— Überprüfung von Variablenbereichen;

— Plausibilitätsprüfung der Werte;

— Überprüfung von Parametern für Prozeduren bei Prozedureingabe auf Typ, Dimension und Wertebereich.

Diese ersten drei defensiven Methoden tragen dazu bei sicherzustellen, dass die vom Programm verarbeiteten Zahlen sowohl in Bezug auf die Programmfunktion als auch auf die physische Bedeutung der Variablen angemessen sind.

Read-only- und Read-write-Parameter müssen getrennt und der Zugriff darauf kontrolliert werden. Funktionen müssen alle Parameter als Read-only behandeln. Literalkonstanten dürfen keinen Schreibzugriff haben. Dies trägt zur Erkennung eines zufälligen Überschreibens oder fälschlichen Gebrauchs von Variablen bei. Fehlertolerante Software muss so gestaltet werden, dass sie Ausfälle in ihrer eigenen Umgebung oder eine Verwendung außerhalb von nominellen oder voraussichtlichen Bedingungen „erwartet“ und sich auf vordefinierte Weise verhält. Der zweite Satz defensiver Methoden umfasst:

— Plausibilitätsprüfungen von Eingangsvariablen und Zwischenvariablen mit physischer Bedeutung;

— Überwachung der Wirkung von Ausgangsvariablen, vorzugsweise durch direkte Beobachtung der damit verbundenen Zustandsänderungen des Systems;

— Überwachung der eigenen Konfiguration durch die Software selbst; dies beinhaltet sowohl das Vorhandensein und die Zugänglichkeit der erwarteten Hardware als auch die Prüfung auf Vollständigkeit der Software selbst — dies ist besonders wichtig zur Aufrechterhaltung der Integrität nach Wartungsprozeduren.

Einige der defensiven Programmiermethoden, wie z. B. Überwachung des Steuerungsablaufs, befassen sich auch mit externen Ausfällen.

7.4.4.1.7 Strukturierte Programmierung

7.4.4.1.7.1 Ziel

Strukturierte Programmierung muss angewendet werden beim Entwurf und der Implementierung des Programms, sodass es praktisch analysiert werden kann, ohne dass es ausgeführt wird.

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 25119-3:2018-11

7.4.4.1.7.2 Beschreibung

Zur Minimierung der strukturellen Komplexität muss Folgendes vorgenommen werden.

a) Unterteilung des Programms in geeignete kleine Softwarekomponenten, die sicherstellen, dass sie soweit wie möglich entkoppelt sind und alle Interaktionen explizit sind.

b) Aufbau des Kontrollflusses der Softwarekomponente unter Verwendung strukturierter Konstrukte, d. h. Abläufe, Iterationen und Auswahl.

c) Die Anzahl möglicher Pfade durch eine Softwarekomponente gering und das Verhältnis zwischen Ein- und Ausgangsparametern so einfach wie möglich halten.

d) Komplizierte Abzweigungen vermeiden. Insbesondere Vermeidung unbedingter Sprünge („go-to“) in übergeordnete Sprachen.

e) Wo möglich, Schleifenbeschränkungen und Verzweigungen mit Eingangsparametern verbinden.

f) Verwendung komplexer Berechnungen als Entscheidungsgrundlage für Verzweigungen und Schleifen vermeiden. Eigenschaften der Programmiersprache, die den vorstehenden Ansatz unterstützen, müssen gegenüber anderen Eigenschaften, die (angeblich) effizienter sind, bevorzugt werden, es sei denn, dass die Effizienz absolute Priorität genießt.

7.4.4.1.8 Modularisierung

7.4.4.1.8.1 Ziel

Modularisierung muss angewendet werden für die Zerlegung eines Softwaresystems in kleine verständliche Teile, um die Komplexität des Systems zu begrenzen.

7.4.4.1.8.2 Beschreibung

Ein modularer Ansatz oder eine Modularisierung setzt verschiedene Regeln für die Entwurfs-, Codierungs- und Wartungsphasen eines Softwareprojekts voraus. Diese Regeln sind je nach der beim Entwurf verwendeten Entwurfsmethode verschieden. Für die Zwecke dieses Dokuments gilt Folgendes.

— Eine Softwarekomponente muss eine einzige gut festgelegte Aufgabe oder Funktion zu erfüllen haben (anwendbar auf die Punkte 2.4.1 und 2.4.4 der Tabelle 3);

— Verbindungen zwischen Softwarekomponenten müssen begrenzt und streng definiert sein, die Kohärenz in einer Softwarekomponente muss hoch sein (anwendbar auf Punkt 2.4.5 in Tabelle 3);

— Unterprogramm-Sammlungen, die mehrere Softwarekomponentenebenen bieten, müssen aufgebaut werden (anwendbar auf Punkt 2.4.5 in Tabelle 3);

— Die Softwarekomponentengröße muss auf einen in dem Codierungsstandard festgelegten spezifischen Wert begrenzt werden (anwendbar auf Punkt 2.4.1 der Tabelle 3);

— Softwarekomponenten müssen einen einzelnen Eintritts- und Austrittspunkt haben (anwendbar auf Punkt 2.4.4 in Tabelle 3);

— Softwarekomponenten müssen mit anderen Softwarekomponenten über ihre Schnittstellen kommunizieren. Wenn globale oder gemeinsame Variablen verwendet werden, müssen diese gut strukturiert sein, der Zugriff muss kontrolliert und ihre Verwendung muss in jedem Einzelfall begründet werden (anwendbar auf Punkt 2.4.5 in Tabelle 3);

Documents
Document Sharing Hub

Full document details: E DIN EN ISO 25119-3:2018-11

— Alle Softwarekomponenten-Schnittstellen müssen vollständig dokumentiert werden (anwendbar auf Punkt 2.4.5 in Tabelle 3);

— Jede Softwarekomponenten-Schnittstelle darf nur die für ihre Funktion notwendigen Parameter enthalten (anwendbar auf Punkt 2.4.5 in Tabelle 3).

7.4.4.1.9 Komplexitätsmesswerte

7.4.4.1.9.1 Ziel

Komplexitätsmesswerte sind anzuwenden, um die Attribute von Programmen auf der Grundlage der Merkmale der Software selbst oder ihrer Entwicklung oder historischer Prüfdaten vorherzusagen.

7.4.4.1.9.2 Beschreibung

Diese Modelle müssen gegebenenfalls angewendet werden, um bestimmte strukturelle Merkmale der Software zu bewerten und diese in Relation zu einem gewünschten Attribut wie etwa Zuverlässigkeit oder Komplexität zu setzen. Software-Werkzeuge werden zur Bewertung der meisten Messwerte benötigt. Einige der anwendbaren Messwerte werden nachfolgend aufgeführt:

— Theoretischer Komplexitätsgraph – dieser Messwert kann frühzeitig im Lebenszyklus angewendet werden, um Kompromisse zu bewerten, und basiert auf der Komplexität des Programmkontrollgraphen, der durch seine zyklomatische Zahl dargestellt wird;

— Anzahl der Verfahren zur Aktivierung einer bestimmten Softwarekomponente (Zugänglichkeit) – je zugänglicher eine Softwarekomponente ist, desto wahrscheinlicher können Programmierungsfehler behoben werden;

— Halstead-Metrik – dieser Messwert berechnet die Programmlänge durch Zählen der Operatoren und Operanden und bietet einen Messwert für die Komplexität und Größe, der als Grundlage für einen Vergleich bei der Abschätzung zukünftiger Entwicklungsressourcen dient;

— Anzahl der Eintritts- und Austrittsstellen pro Softwarekomponente – die Minimierung der Anzahl der Eintritts- und Austrittspunkte ist eine wesentliche Eigenschaft strukturierter Gestaltungs- und Programmierungsmethoden.

7.4.4.1.10 Verbergen/Kapseln von Informationen

7.4.4.1.10.1 Ziel

Informationen müssen verborgen oder gekapselt werden, um einen unbeabsichtigten Zugriff auf Daten oder Verfahren zu verhindern und so eine gute Programmstruktur zu unterstützen.

7.4.4.1.10.2 Beschreibung

Daten, die allgemein für alle Softwareelemente zugänglich sind, können versehentlich oder falsch durch jedes dieser Elemente geändert werden. Jede Veränderung dieser Datenstrukturen kann eine ausführliche Prüfung des Codes und umfangreiche Änderungen erfordern.

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 25119-3:2018-11

Das Verbergen von Informationen ist ein allgemeiner Ansatz, der gegebenenfalls zur Minimierung dieser Probleme angewendet werden muss. Die Hauptdatenstrukturen werden „verborgen“ und können nur durch eine definierte Reihe von Zugangsverfahren manipuliert werden. Dies ermöglicht es, die internen Strukturen zu verändern oder weitere Verfahren hinzufügen, ohne das funktionale Verhalten der übrigen Software zu beeinträchtigen. Ein Namensverzeichnis könnte beispielsweise die Zugangsverfahren „Einfügen“, „Löschen“ und „Suchen“ haben. Die Zugangsverfahren und internen Datenstrukturen können umgeschrieben werden (beispielsweise zur Nutzung eines anderen Suchverfahrens oder um die Namen auf einem Festplattenlaufwerk zu speichern), ohne das logische Verhalten der übrigen Software, die diese Verfahren nutzt, zu beeinträchtigen.

In diesem Zusammenhang sollte das Konzept der abstrakten Datentypen angewendet werden. Falls keine direkte Unterstützung geboten wird, kann es erforderlich sein, zu prüfen, ob die Abstraktion nicht versehentlich beschädigt wurde.

7.4.4.1.11 Bibliothek vertrauenswürdiger/verifizierter Softwarekomponenten

7.4.4.1.11.1 Ziel

Eine Bibliothek vertrauenswürdiger/verifizierter Softwarekomponenten muss Anwendung finden, um zu vermeiden, dass für jede neue Anwendung eine umfangreiche erneute Validierung oder Umarbeitung notwendig wird. Diese Methode erlaubt es dem Entwickler, Entwürfe erneut zu verwenden, die nicht formal oder streng validiert wurden, für die aber eine umfangreiche Betriebshistorie existiert (siehe auch 7.4.4.1.5).

7.4.4.1.11.2 Beschreibung

Einwandfrei konstruierte und strukturierte E/E/PES müssen aus einer Reihe von Hardware- und Softwarekomponenten bestehen, die eindeutig verschieden sind und miteinander auf eindeutig spezifizierte Weise interagieren.

ANMERKUNG 1 Verschiedene für unterschiedliche Anwendungen entworfene E/E/PES enthalten gelegentlich eine Reihe von Softwarekomponenten, die identisch oder sich sehr ähnlich sind. Der Aufbau einer Bibliothek derartiger generell anwendbarer Softwarekomponenten ermöglicht die gemeinsame Nutzung der für die Validierung der Entwürfe notwendigen Ressourcen durch mehr als eine Anwendung.

ANMERKUNG 2 Darüber hinaus liefert die Verwendung solcher Softwarekomponenten in mehreren Anwendungen den empirischen Nachweis des erfolgreichen Betriebseinsatzes. Dieser empirische Nachweis stärkt berechtigterweise das Vertrauen, das Nutzer voraussichtlich in die Softwarekomponenten haben.

7.4.4.1.12 Rechnergestützte Entwurfswerkzeuge

7.4.4.1.12.1 Ziel

Rechnergestützte Entwurfswerkzeuge (CAD) müssen eingesetzt werden, um die Durchführung des Entwurfsverfahrens systematischer zu gestalten und bei der Aufnahme geeigneter automatischer Konstruktionselemente, die bereits vorhanden und getestet sind, zu unterstützen.

7.4.4.1.12.2 Beschreibung

Rechnergestützte Entwurfswerkzeuge (CAD) müssen beim Entwurf sowohl der Hardware als auch der Software eingesetzt werden, sofern sie verfügbar und wegen der Komplexität des Systems gerechtfertigt sind. Die Fehlerfreiheit dieser Werkzeuge muss durch spezielle Tests, durch eine umfangreiche Historie des zufriedenstellenden Einsatzes oder durch unabhängige Verifizierung ihrer Leistung für das speziell im Entwurf befindliche sicherheitsbezogene System nachgewiesen werden.

AVA Documents
Document Sharing Hub

Full document details: E DIN EN ISO 25119-3:2018-11

7.4.4.1.13 Verwendung von Codierungsstandards

7.4.4.1.13.1 Ziel

Codierungsstandards sind zur besseren Verifizierbarkeit des erstellten Codes einzusetzen.

7.4.4.1.13.2 Beschreibung

Die einzuhaltenden detaillierten Regeln müssen vor dem Codieren vollständig vereinbart werden. Diese Regeln umfassen normalerweise:

— Einzelheiten der Modularisierung, z. B. Schnittstellenformen, Größen der Softwarekomponenten;

— Verwendung von Kapselung, Vererbung (mit Tiefenbeschränkung) und Polymorphie im Fall objektorientierter Programmiersprachen;

— begrenzter Einsatz oder Vermeidung von bestimmten Sprachkonstrukten, z. B. „go-to“, „equivalence“, dynamischen Objekten, dynamischen Daten, dynamischen Datenstrukturen, Rekursion, automatische Typenkonvertierung, Zeigern und Aussprüngen;

— Beschränkungen von während der Ausführung von sicherheitskritischen Codes aktivierten Interrupts;

— Code-Layout (Auflistung) und

— keine unbedingten Sprünge (z. B. „go-to“) in Programmen in problemorientierten Sprachen.

Diese Regeln sind so abgefasst, dass sie ein einfaches Testen, Verifizieren, Beurteilen und Warten von Softwarekomponenten zulassen. Daher müssen sie vorhandene Werkzeuge, insbesondere Analysatoren, berücksichtigen.

7.4.4.1.14 Entwurfs- und Codierungsstandards — Keine dynamischen Variablen oder Objekte

7.4.4.1.14.1 Ziel

Entwurfs- und Codierungsstandards müssen zum Ausschluss von dynamischen Variablen oder Objekten eingesetzt werden, wie beispielsweise

— unerwünschte oder unerkannte Speicherüberlagerungen und

— Ressourcenengpässe während der (sicherheitsbezogenen) Ausführungszeit.

7.4.4.1.14.2 Beschreibung

Im Fall dieser Maßnahme sind dynamische Variablen und dynamische Objekte diejenigen Variablen und Objekte, denen während der Ausführungszeit ihr Speicherplatz zugewiesen und die absoluten Adressen festgelegt werden. Der Wert des zugewiesenen Speichers und seiner Adressen hängt vom Zustand des Systems zum Zeitpunkt der Zuweisung ab, d. h. er kann vom Compiler oder irgendeinem anderen Offline-Werkzeug nicht kontrolliert werden.

Weil die Zahl dynamischer Variablen und Objekte und der vorhandene freie Speicherplatz für die Zuweisung neuer dynamischer Variablen oder Objekte vom Zustand des Systems zum Zeitpunkt der Zuweisung abhängen, können beim Zuweisen oder bei der Verwendung der Variablen oder Objekte Fehler auftreten. Reicht beispielsweise der freie Speicherplatz im vom System zugewiesenen Speicherbereich nicht aus, so kann der Speicherinhalt einer anderen Variablen versehentlich überschrieben werden. Diese Fehler werden vermieden, wenn keine dynamischen Variablen oder Objekte verwendet werden.

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 25119-3:2018-11

7.4.4.1.15 Online-Prüfung durch Erstellung dynamischer Variablen oder dynamischer Objekte

7.4.4.1.15.1 Ziel

Die Online-Prüfung während der Erstellung dynamischer Variablen oder Objekte muss angewendet werden, um zu prüfen, ob der den dynamischen Variablen und Objekten zugewiesene Speicherplatz frei ist, bevor die Zuweisung stattfindet. Hierdurch soll sichergestellt werden, dass die Zuweisung der dynamischen Variablen und Objekte während der Laufzeit keine bestehenden Variablen, Daten oder Code beeinträchtigt.

7.4.4.1.15.2 Beschreibung

Für diese Maßnahme sind dynamische Variablen die Variablen, deren Speicherzuweisung und Bestimmung der absoluten Adresse während der Laufzeit stattfindet (Variablen sind in diesem Sinne ebenfalls Attribute der Objektinstanzen). Der Speicher wird durch Hardware oder Software geprüft, um sicherzustellen, dass er frei ist, bevor ihm eine dynamische Variable oder ein Objekt zugewiesen wird (zum Beispiel um einen Stapelüberlauf zu verhindern). Wenn eine Zuweisung nicht erlaubt ist (beispielsweise, wenn an der festgelegten Adresse nicht genügend Speicher frei ist), müssen entsprechende Maßnahmen getroffen werden. Nachdem eine dynamische Variable oder ein Objekt verwendet wurde (z. B. nach dem Verlassen einer Subroutine), muss der gesamte ihr/ihm zugewiesene Speicher freigegeben werden.

7.4.4.1.16 Entwurfs- und Codierungsstandards — Eingeschränkte Verwendung von Interrupts

7.4.4.1.16.1 Ziel

Der Software-Entwickler muss die Verwendung von Interrupts limitieren, damit die Software verifizierbar und testfähig bleibt.

7.4.4.1.16.2 Beschreibung

Die Verwendung von Interrupts muss eingeschränkt werden. Interrupts können verwendet werden, wenn sie das System vereinfachen. Die Interrupt-Behandlung durch die Software muss während kritischer Teile der ausgeführten Funktionen blockiert werden. Werden Interrupts verwendet, dann dürfen nicht unterbrechbare Teile nur eine spezifizierte maximale Rechenzeit aufweisen, damit die maximale Zeit, für die ein Interrupt blockiert ist, berechnet werden kann. Die Verwendung von Interrupts muss genauestens dokumentiert werden.

7.4.4.1.17 Entwurfs- und Codierungsstandards — Definierte Verwendung von Zeigern

7.4.4.1.17.1 Ziel

Die definierte Verwendung von Zeigern muss bei der Vermeidung von Problemen durch Zugriff auf die Daten ohne vorherige Prüfung des Zeigerbereichs und -typs behilflich sein und die modularen Tests und Verifizierungen der Software sowie die Begrenzung der Folgen von Ausfällen unterstützen.

7.4.4.1.17.2 Beschreibung

In der Betriebs- und Anwendungssoftware darf die Zeigerarithmetik auf Quellcodeebene nur verwendet werden, wenn der Zeigerdatentyp und der Wertbereich (um sicherzustellen, dass sich die Zeigerreferenz im richtigen Adressraum befindet) überprüft werden.

7.4.4.1.18 Entwurfs- und Codierungsstandards — Eingeschränkte Verwendung von Rekursionen

7.4.4.1.18.1 Ziel

Die Verwendung von Rekursionen muss zur Vermeidung einer nicht verifizierbaren und instabilen Verwendung von Unterprogrammaufrufen eingeschränkt werden.

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 25119-3:2018-11

7.4.4.1.18.2 Beschreibung

Wird Rekursion verwendet, muss ein eindeutiges Kriterium bezüglich der erlaubten Rekursionstiefe festgelegt werden.

7.4.4.2 Verifizierung der Softwarekomponentengestaltung und -codierung

Gestaltung und Codierung der Softwarekomponente müssen verifiziert werden. Es muss geprüft werden, ob der Entwurf und die Codierung die Sicherheitsanforderungen an die Software erfüllen. Die Prüfspezifikation und der Prüfbericht nach Tabelle 6 müssen ebenfalls geprüft werden, um zu verifizieren, ob diese mit der Gestaltung und Codierung der Softwarekomponente konsistent sind. Die Software-Entwickler müssen an den Verifizierungsaktivitäten teilnehmen. Zwei Arten von Verifizierungsmethoden sind möglich: Inspektion und Walkthrough (wie in ISO 25119-1 definiert). Eine Verifizierung eines automatisch generierten Codes ist nicht zwingend erforderlich.

7.4.5 Arbeitsprodukte

Folgende Arbeitsprodukte sind auf diese Phase anwendbar:

a) die ausführliche Gestaltung der Software nach 7.4.4.1;

b) Software nach 7.4.4.1;

c) Verifizierungsbericht über die Gestaltung und Codierung der Softwarekomponente, der sich aus den Anforderungen in 7.4.4.2 ergibt.

7.5 Softwarekomponentenprüfung

7.5.1 Ziele

Softwarekomponentenprüfungen haben zum Ziel, zu verifizieren, dass die entworfenen und codierten Softwarekomponenten die Softwareanforderungen korrekt implementieren.

7.5.2 Allgemeines

In dieser Phase muss ein Verfahren für das Prüfen der Softwarekomponenten gegen ihre Anforderungen erstellt und die Prüfungen entsprechend diesem Verfahren durchgeführt werden.

7.5.3 Voraussetzungen

Die folgenden Voraussetzungen gelten für die Softwarekomponentenprüfung:

— Software-Projektplan (siehe 7.1.1 bis 7.1.4.6);

— Software-Anforderungen [siehe 7.2.5 a) und b)];

— Software-Verifizierungsplan (siehe ISO 25119-4:2018, Abschnitt 6).

— Softwarekomponenten nach 7.4.4.1;

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 25119-3:2018-11

7.5.4 Anforderungen

7.5.4.1 Verfahren der Softwarekomponentenprüfung

Die Softwarekomponentenprüfung muss nach Tabelle 4 erfolgen. Softwarekomponentenprüfungspläne müssen die zur Verifizierung des Soll-SRL ausgewählten Methoden/Maßnahmen dokumentieren. Die Festlegungen für Softwarekomponentenprüfungen müssen die anzuwendenden Methoden für die Durchführung dieser Methoden/Maßnahmen beschreiben. Softwarekomponenten-Prüfberichte müssen die durchgeführten Prüfungen und die Prüfergebnisse dokumentierten.

Tabelle 4 — Softwarekomponentenprüfung

Methoden/Maßnahme[a]	Unterabschnitt	SRL = B	SRL = 1	SRL = 2	SRL = 3
1 Statische Analyse					
1.1 Grenzwertanalyse	7.5.4.1.1	+	+	+	+
1.2 Checklisten	7.5.4.1.2	o	o	o	o
1.3 Kontrollflussanalyse	7.5.4.1.3	o	o	+	+
1.4 Datenflussanalyse	7.5.4.1.4	o	o	+	+
2 Dynamische Analyse und Tests	7.5.4.1.5				
2.1 Testfallausführung anhand der Grenzwertanalyse	7.5.4.1.6	o	o	o	+
2.2a Umfang der Strukturprüfung (Eintrittspunkte)	7.5.4.1.7	o	o	+	x
2.2b Umfang der Strukturprüfung (Anweisungen)	7.5.4.1.7	o	o	+	+
2.2c Umfang der Strukturprüfung (Verzweigungen)	7.5.4.1.7	o	o	+	+
3 Einheitenprüfung					
3.1 Äquivalenzklassen und Eingabe-Partitionstests	7.5.4.1.8	o	o	+	+
3.2 Grenzwertanalyse	7.5.4.1.1	o	o	+	+
3.3 Prüffallausführung aus modellbasierter Prüffallgenerierung	7.5.4.1.9	o	o	o	+
4 Leistungsnachweis	7.5.4.1.10				
4.1 Ansprechzeiten und Speicherbeschränkungen	7.5.4.1.11	o	+	+	+
4.2 Leistungsanforderungsprüfung	7.5.4.1.12	o	+	+	+
4.3 Belastungs-/Stresstest	7.5.4.1.13	o	o	o	+
5 Schnittstellentests	7.5.4.1.14	o	o	o	+

Siehe Anweisungen aus 7.1.4.7 für die Verwendung der Tabellen 1 bis 6.

[a] Es müssen geeignete Methoden/Maßnahmen entsprechend dem SRL gewählt werden. Alternative oder gleichwertige Methoden/Maßnahmen sind durch einen Buchstaben nach der Zahl gekennzeichnet. Nur eine der Methoden/Maßnahmen muss erfüllt sein.

Documents
Document Sharing Hub
Full document details: E DIN EN ISO 25119-3:2018-11