FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Möglichkeiten des Nachweises der funktionalen Sicherheit von technischen Systemen

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Wirtschaftsingenieurwesen Informatik

eingereicht von

Duc-Hoa Do
Matrikelnummer 0025533

an der
Fakultät für Informatik der Technischen Universität Wien
Institut für Rechnergestützte Automation

Betreuung
Betreuer: Univ.-Prof. Dr.-Ing. Gerhard-Helge Schildt

Wien, 25.07.2013

(Unterschrift Verfasser) (Unterschrift Betreuer)

Duc-Hoa Do
Hernalser Hauptstrasse 7/23
1170 Wien

Erklärung
„Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe."

Wien, 25.07.2013

(Unterschrift Verfasser)

Abstract

Compliance with basic safety related principles does not only increase dependability of technical systems, but is also often a legal requirement.
This master thesis deals with the two standards EN 61508 and ISO 26262 first. EN 61508 is generic and mainly addresses risk assessment of technical systems, while the focus of ISO 26262 is on production vehicles. Central concept of both standards is the measurement of safety relevance, which is specified in SIL and ASIL. Next an introduction to the fail-safe microcomputer system SIMIS is given and design, setup and mode of operation are explained. The application of FMECA is also a generally accepted method to improve the safety of a technical system. A closer look is taken at the historical development, the requirements and purpose, the FMECA worksheets, the risk priority number, the risk matrix and the limits of FMECA. The usage of watchdogs is also another safety related principle. First, two application examples are given, then the basic structure along with architecture types and designs are presented. A sketch of the optimal watchdog is made and ideas for multi-stage watchdogs are offered. The functioning of the quiescent current principle is described and the implementation is shown in three applications examples. Last chapter shows advantages of dynamization of control signals. Application is demonstrated by a functional circuit with transistor stage and the LOGISAFE system.

Kurzfassung

Die Einhaltung grundlegender sicherheitstechnische Prinzipien erhöht nicht nur die Zuverlässigkeit von technischen Systemen, sondern wird häufig auch vom Gesetz gefordert.
Diese Diplomarbeit beschäftigt sich zuerst mit den zwei Normen EN 61508 und ISO 26262. Während die EN 61508 generisch ist und sich hauptsächlich mit der Risikobewertung von sicherheitsbezogenen technischen Systemen befasst, bezieht sich die ISO 26262 auf Serienfahrzeuge. Zentraler Begriff beider Normen ist das Maß der Sicherheitsrelevanz, die in SIL und ASIL angegeben wird. Danach wird das Sichere Mikrocomputersystem SIMIS vorgestellt und Konzeption, Aufbau und Funktionsweise erläutert. Die Anwendung von FMECA ist ebenso eine anerkannte Methode, um die Sicherheit eines technischen Systems weiter zu verbessern. Hier wird auf die historische Entwicklung, Voraussetzung und Einsatzzweck, die FMECA Arbeitsblätter, Risikoprioritätszahl, Risikomatrix und die Grenzen der FMECA eingegangen. Die Verwendung von Funktionswächtern ist ein weiteres sicherheitstechnisches Prinzip. Nach zwei Einsatzbeispielen wird der grundlegende Aufbau dieser sogenannten Watchdogs erklärt sowie Architekturen und Ausführungen behandelt. Eine Skizzierung des optimalen Watchdogs wird vorgenommen und Ideen zu mehrstufigen Funktionswächtern werden angeboten. Die Funktionsweise des Ruhestromprinzips wird beschrieben und die Umsetzung in drei Anwendungsbeispielen veranschaulicht. Als letztes werden die Vorteile der Dynamisierung von Steuersignalen gezeigt. Die Umsetzung wird anhand einer Funktionsschaltung mit Transistorstufe und dem LOGISAFE System demonstriert.

Danksagung

Hiermit möchte ich mich bei meinen Eltern Đỗ Văn Thoại und Đỗ Thúy Nga für ihre unermüdliche Hilfe bedanken. Sie haben mir nicht nur das Studium ermöglicht, sondern auch während meiner gesamten Studienzeit stets an mich geglaubt, mich ermutigt und bestmöglich gefördert.

Meinen beiden Geschwistern Đỗ Đức Huy und Nguyễn Kim Liên möchte ich ebenso für ihre Unterstützung danken.

Ebenfalls möchte ich mich beim Herrn Univ.-Prof. Dr.-Ing. Gerhard-Helge Schildt bedanken, welcher nicht nur sehr geduldig mit mir war, sondern auch immer mit Wissen, Erfahrung und Ratschlag weiter geholfen hat.

Schlussendlich möchte ich mich bei meiner Frau Đỗ Ngọc Linh bedanken, ohne deren Liebe, Unterstützung und Motivation ich es nicht geschafft hätte.

Inhaltsverzeichnis

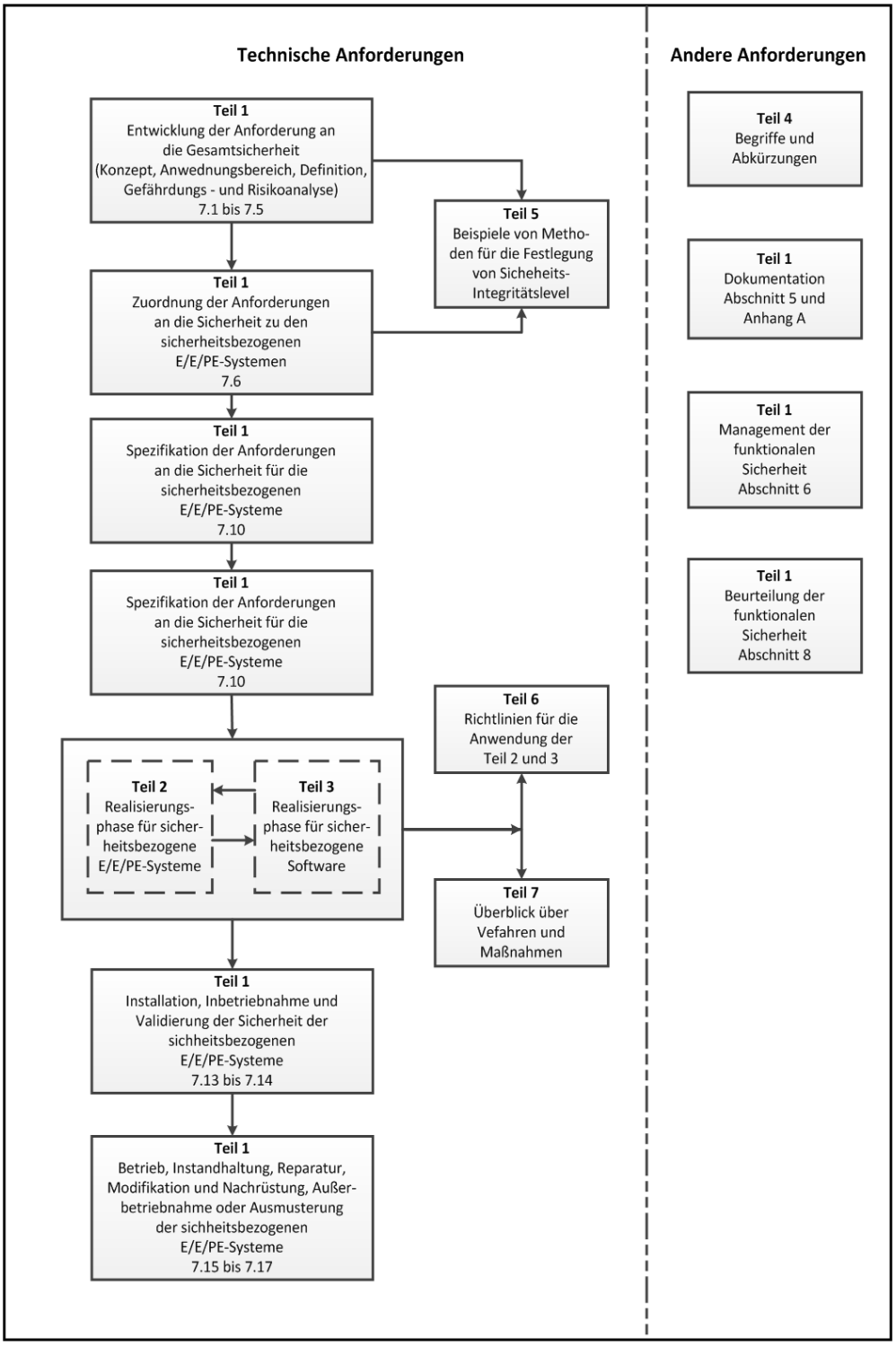
1. EN 61508 Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme

Die EN 61508 ist ein internationaler Standard und befasst sich mit der Entwicklung von elektrischen, elektronischen und programmierbar elektronischen (E/E/PE) Systemen, die eine Sicherheitsfunktion ausführen. Im Jahr 1998 wurde sie unter IEC 61508 veröffentlicht und 2001 wurde sie durch die Europäische Normenorganisation CENELEC (European Committee for Electrotechnical Standardization) ratifiziert. Die derzeit aktuelle und gültige Fassung stammt aus dem Jahr 2010, die im Februar 2011 ins Deutsche übersetzt wurde.

Die EN 61508 ist eine branchenunabhängige und generisch verfasste Richtlinie, die auf alle sicherheitsgerichteten Systeme angewendet werden kann [21][55]. Sie gilt als Stand der Technik [7][83] und beschreibt den kompletten Lebenszyklus des gesamten sicherheitsgerichteten Systems von der Planung bis zur Außerbetriebnahme [86]. Alle Aspekte, die mit der Nutzung und Anforderung an elektrische/elektronische/programmierbare elektronische Systeme (kurz E/E/PE-Systeme) für sicherheitsrelevante Funktionen werden darin ausführlich beleuchtet [85].

Die Norm besteht aus sieben Teilen, wobei die ersten vier normative Anforderungen enthalten. Die letzten drei Teile sind rein informativ und enthalten praktische Beispiele, um den Umgang mit der Norm verständlicher zu machen.

Abbildung 1: Gesamtrahmen der Normreihe IEC 61508



[Quelle: eigene Darstellung in Anlehnung an European Committee for Electrotechnical Standardization: *EN 61508 Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme*. 2010, Teil 1, S. 9]

Teil 1: Allgemeine Anforderungen

Der erste Teil enthält allgemeine und grundlegende Anforderungen, die angewendet werden sollen, wenn sonst keine anwendungsspezifische abgeleitete Norm vorliegt. Sind solche verfügbar, soll die EN 61508 ergänzend genutzt werden [85]. Der zeitliche Anwendungsbereich dieser Norm erstreckt sich über den gesamten Lebenszyklus des sicherheitsgerichteten Systems – von der Planung über Konzeption, Entwicklung, Instandhaltung und Wartung bis zur Außerbetriebnahme [86]. Es wird ein Sicherheitslebenszyklus definiert, um alle notwendigen Tätigkeiten und Maßnahmen systematisch abhandeln zu können.

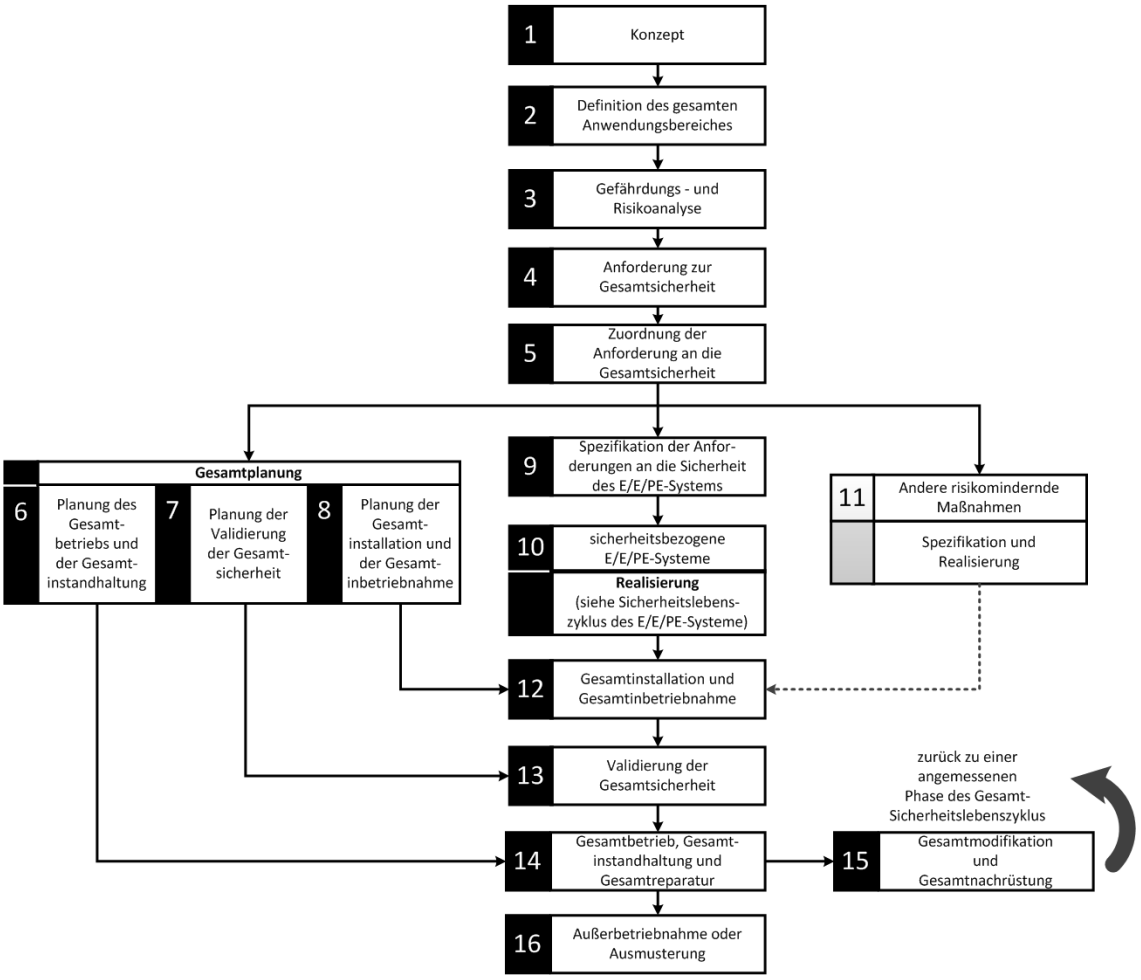
Sicherheitslebenszyklus (safety lifecycle):

Notwendige Tätigkeiten im Rahmen der Realisierung von sicherheitsbezogenen Systemen während eines Zeitraumes, der mit der Konzeptphase eines Projektes beginnt und endet, wenn alle sicherheitsbezogenen E/E/PE-Systeme und andere risikomindernde Maßnahmen nicht mehr für die Verwendung verfügbar sind.[1]

Maßnahmen zum Management, der Verifikation und der Beurteilung der funktionalen Sicherheit müssen in allen Phasen Sicherheitslebenszyklus umgesetzt und dokumentiert werden [21]. Die Verantwortung für die Durchführung wird an qualifizierte Personen oder Organisationen übergeben. Es muss außerdem überprüft werden, ob die durchgeführten Management-Tätigkeiten wie gewünscht wirken.
Alternativ können die Prozesse dafür in der ISO 9001 (Qualitätsmanagement) des Unternehmens integriert werden [85].
Vier Sicherheitsanforderungsstufen, sogenannte Sicherheits-Integritätslevel (safety integrity level, SIL), werden eingeführt. Diese werden im nächsten Kapitel näher erläutert.

[1] EN 61508 - Funktionale Sicherheit sicherheitsbezogener elektrischer / elektronischer / programmierbarer elektronischer Systeme, Teil 4: Begriffe und Abkürzungen, 2010, Abschnitt 3.7.1

Abbildung 2: Gesamt-Sicherheitslebenszyklus



[Quelle: eigene Darstellung in Anlehnung an European Committee for Electrotechnical Standardization: *EN 61508 Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme.* 2010, Teil 1, S. 17]

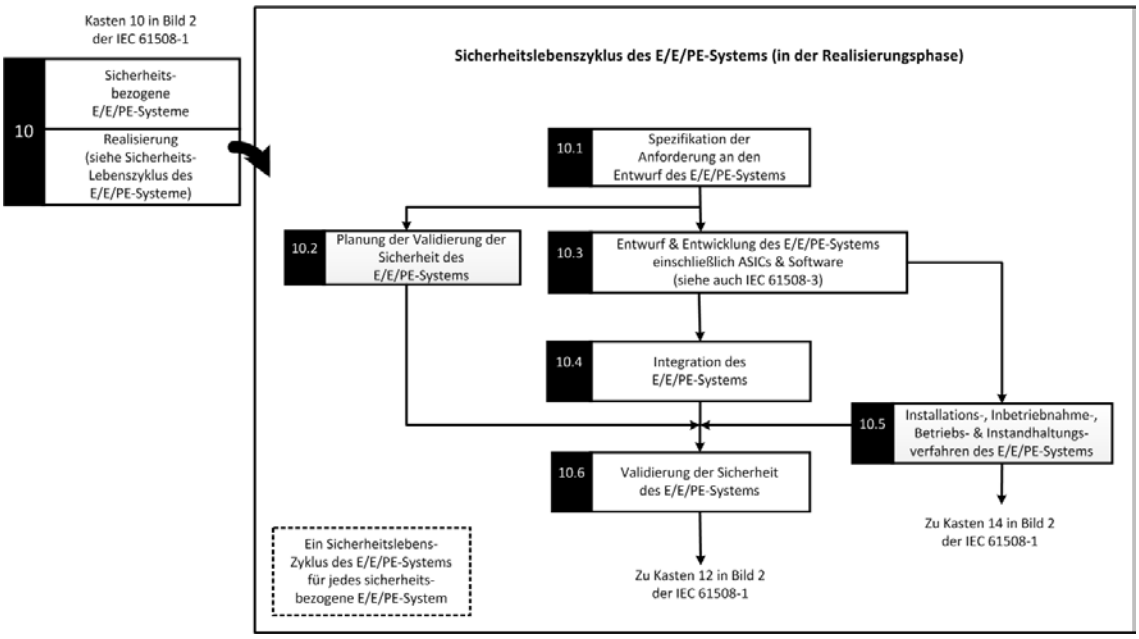
Anhang A enthält ein Beispiel einer Dokumentationsstruktur und eine Methode zur Festlegung der Dokumente zur Strukturierung der Informationen.

Teil 2: Anforderungen an sicherheitsbezogene elektrische/elektronische/programmierbare elektronische Systeme

Teil 2 der EN 61508 stellt die Hardware des sicherheitsrelevanten Systems in den Mittelpunkt. Er wird auf jedes sicherheitsbezogene System/Teilsystem angewendet, dass mindestens ein E/E/EP-Bauteile enthält [9].

Innerhalb des Gesamt-Sicherheitslebenszyklus wird für die Hardware des Systems ein detaillierter Sicherheitslebenszyklus definiert [85].

Abbildung 3: Sicherheitslebenszyklus des E/E/PES (in der Realisierungsphase)



[Quelle: eigene Darstellung in Anlehnung an European Committee for Electrotechnical Standardization: *EN 61508 Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme*. 2010, Teil 1, S. 18]

Die Phasen des E/E/PES-Lebenszyklus werden zunächst in einer systematischen Art strukturiert (siehe Abbildung 3). Jede Phase enthält eigene Ziele und Anforderungen und wird in elementare Tätigkeiten unterteilt.

Neben der Spezifikation der Anforderungen an den Entwurf werden auch Pläne zur Validierung der Sicherheit der Hardware erstellt. Diese Pläne enthalten u.a. Strategien, Methoden, erforderliche Umgebung, Testverfahren sowie Bewertungskriterien und -verfahren [20].
Es ist wichtig zu prüfen, ob die festgelegten Anforderungen angemessen sind. Insbesondere müssen die Sicherheitsfunktion, die Anforderungen an die Sicherheitsintegrität und die Betriebsmittel- und Bedienerschnittstellen betrachtet werden [20].

Jede abgeschlossene Phase des E/E/PES-Sicherheitslebenszyklus wird anhand der definierten Anforderung verifiziert, bewertet und dokumentiert werden.

Funktionale Sicherheit
Teil der Gesamtsicherheit, bezogen auf die EUC (Equipment Under Control) und das EUC-Leit- oder Steuerungssystem, der von der korrekten Funktion des sicherheitsbezogenen E/E/PE-Systems und anderer risikomindernder Maßnahmen abhängt.[2]

Anhang A und B enthalten Verfahren und Maßnahmen, um systematische Hardwarefehler und Ausfälle während des Betriebes zu beherrschen. Anhang C behandelt die Berechnung des Diagnosedeckungsgrades und den Anteil der sicheren Ausfälle eines Hardwareelements. Anhang D definiert ein Sicherheitshandbuch für konforme Objekte, um die Integration desselben in ein sicherheitsbezogenes System zu ermöglichen. Die Anhänge E und F beschreiben schließlich besondere Architekturanforderungen an integrierte Schaltkreise (ICs) mit On-Chip-Redundanz, sowie Verfahren und Maßnahmen zur Vermeidung von systematischen Ausfällen für ASICs (anwendungsspezifische integrierte Schaltkreise).

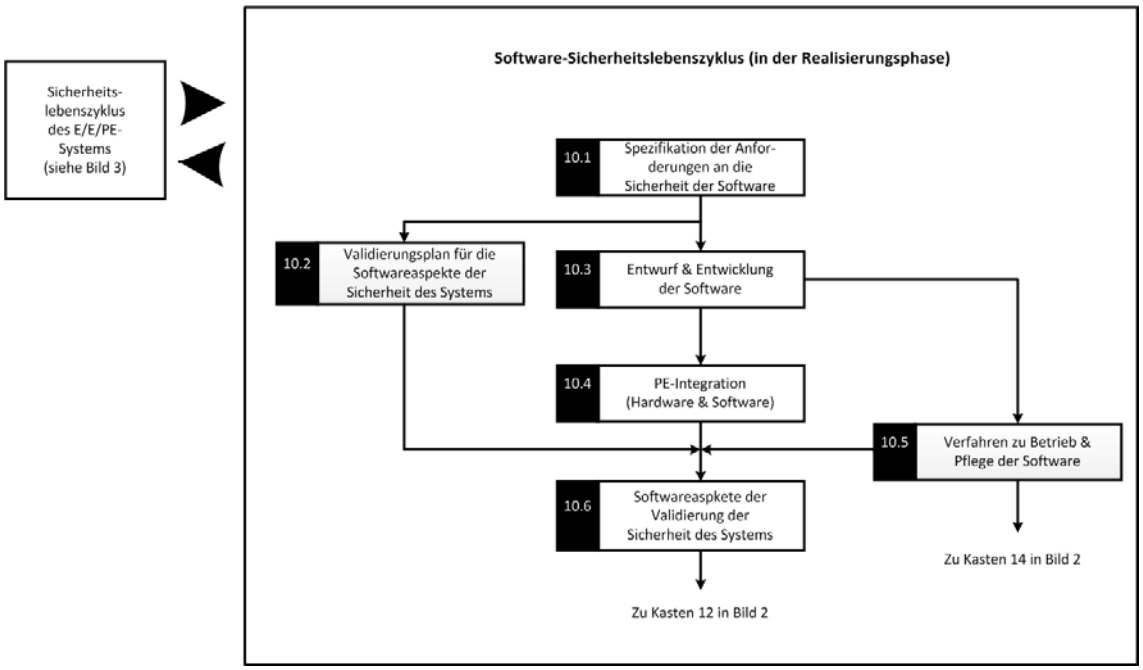
Teil 3: Anforderungen an Software
Der dritte Teil der Norm beschreibt Techniken und Verfahren, wie sicherheitsgerichtete Software entwickelt und dokumentiert werden soll. Auch hier wird speziell für die Software des E/E/PE-Systems ein detaillierter Sicherheitslebenszyklus definiert [85].
Analog zum 2.Teil müssen auch in allen Phasen des Softwarelebenszyklus Anforderungen, Ziele und Maßnahmen geplant, durchgeführt und dokumentiert werden.

Spezifiziert werden nicht nur Anforderungen an die Sicherheit der Software, sondern auch die Anforderungen an die zu verwendenden Werkzeuge. Im Besonderen gelten diese für Werkzeuge zur Entwicklung und zum Entwurf,

[2] EN 61508: a.a.O., Abschnitt 3.1.12

Übersetzer, Werkzeuge für Test und Debugging und Werkzeuge für das Konfigurationsmanagement [20].

Abbildung 4: Software-Sicherheitslebenszyklus (in der Realisierungsphase)



[Quelle: eigene Darstellung in Anlehnung an European Committee for Electrotechnical Standardization: *EN 61508 Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme*. 2010, Teil 1, S. 18]

Die Sicherheitsanforderungen an die Software sollten aus den entsprechenden Anforderungen an das E/E/PE-System und den gesamten Sicherheitsanforderungen aus Teil 1 hergeleitet sein. Auf die Rückverfolgbarkeit zwischen den Sicherheitsanforderungen, der Implementierung und den entsprechenden Tests auf den verschiedenen Ebenen des gesamten Lebenszyklus soll ebenso geachtet werden [85].
In Anhang A und B schlägt die Norm Maßnahmen und Verfahren vor, um Software entsprechend dem geforderten Sicherheits-Integritätslevel (SIL) zu entwerfen, zu implementieren und zu testen [21]. Jedes Verfahren bzw. jede Maßnahme wird hinsichtlich der SIL-Anforderungen bewertet. Anhang C gibt eine Orientierungshilfe bei der Auswahl von Verfahren aus den Anhängen A und

B, um die systematische Eignung der Software zu gewährleisten. Anhang D definiert ein Sicherheitshandbuch für konforme Objekte speziell für Softwareelemente und Anhang E stellt Teil 2 und Teil 3 nochmals in Beziehung.

Teil 4: Begriffe und Abkürzungen
Dieser Teil beinhaltet Definitionen und Beschreibungen der Begriffe, die in allen Teilen der EN 61508 verwendet werden.

Teil 5: Beispiele zur Ermittlung der Stufe der Sicherheitsintegrität (safety integrity level)
Der fünfte Teil besteht lediglich aus den Anhängen A bis F. Die ersten zwei Anhänge liefern Informationen über die zugrunde liegenden Konzepte des Risikos und den Zusammenhang zwischen Risiko und Sicherheitsintegrität. In den restlichen Anhängen werden darüber hinaus eine Auswahl von Methoden zur Bestimmung der Anforderung an den Sicherheits-Integritätslevels vorgestellt [21].

Teil 6: Anwendungsrichtlinie für IEC 61508-2 und IEC 61508-3
Auch dieser Abschnitt besteht aus den Anhängen A bis E. Es wird nochmal ein Überblick über Teil 2 und Teil 3 hinsichtlich Anforderungen und funktionaler Schritte für die Anwendung gegeben. Außerdem enthält dieser Teil Beispiele zur Berechnung und Bewertung von Hardwareausfällen, für die Berechnung des Diagnosedeckungsgrads sowie eine Methode zur Quantifizierung der Auswirkungen von hardwarebezogenen Ausfällen infolge gemeinsamer Ursache. Schließlich werden noch Beispiele für die Anwendung der Tabellen zur Sicherheitsintegrität von Software, Blockdiagramme und Formeln für die Berechnung von PFD (Probability of Failure on Demand) und PFH (Probability of Dangerous Failure per Hour) und Tabellen mit berechneten PFD und PFH-Zahlen angeführt.

Diagnosedeckungsgrad DC (diagnostic coverage)
Anteil der gefahrbringenden Ausfälle, die durch automatische diagnostische Online-Prüfungen erkannt werden. Der Anteil der gefahrbringenden Ausfälle wird mittels der Raten gefahrbringender Ausfälle, die zu den erkannten

gefahrbringenden Ausfällen gehören, geteilt durch die Gesamtrate der gefahrbringenden Ausfälle berechnet.[3]

PFD (probability dangerous failure on demand):
Wahrscheinlichkeit eines gefahrbringenden Ausfalls bei Anforderung[3]

PFD_{avg} (average probability dangerous failure on demand):
Mittlere Wahrscheinlichkeit eines gefahrbringenden Ausfalls bei Anforderung der Sicherheitsfunktion[3]

PFH (average frequency of a dangerous failure per hour):
Mittlere Häufigkeit eines gefahrbringenden Ausfalls eines sicherheitsbezogenen E/E/PE-Systems, die festgelegte Sicherheitsfunktion über einen gegebenen Zeitraum auszuführen[3]

Teil 7: Überblick über Verfahren und Maßnahmen
Der letzte Teil beinhaltet einen vollständigen Überblick über verschiedenen Sicherheitsverfahren und -maßnahmen, die in den Teilen 2 und 3 vorgestellt wurden [9]. Dieser Teil dient als Nachschlagewerk und ist gegliedert in Beherrschung von zufälligen Hardwareausfällen, Vermeidung von systematischen Ausfällen, Verfahren und Maßnahmen zum Erreichen der Sicherheitsintegrität der Software, probabilistischer Ansatz zur Bestimmung der Sicherheitsintegrität von vorentwickelter Software und Verfahren und Maßnahmen für den Entwurf von ASICs (Application Specific Integrated Circuit).

[3] EN 61508: a.a.O., Abschnitt 3.8.6

2. Sicherheitsintegritätstufen SIL (safety integrity level)

Eine Hauptforderung der Norm besteht darin, den quantitativen Nachweis für das bleibende Restrisiko bezüglich der funktionalen Sicherheit zu erbringen. Funktionale Sicherheit ist gegeben, wenn jede spezifizierte Sicherheitsfunktion ausgeführt und der für jede Sicherheitsfunktion geforderte Erfüllungsgrad erreicht wird [16]. Das Risikopotential eines Systems muss bestimmt werden, bevor eine Risikominimierung durchgeführt werden kann. Zur Verringerung des Risikos führt die Norm zwei essentielle Schritte an:

- Definition und Bewertung des Risikos basierend auf detaillierten Ausfallwahrscheinlichkeiten für den gesamten Lebenszyklus
- Periodische Kontrolle der Einhaltung von Vorgaben. Der betrachtete Gesamt-Sicherheitslebenszyklus ist in 16 Teilaspekte unterteilt (siehe Abbildung 2)

Erfolgt die Risikominimierung mithilfe elektrischer Komponenten, so müssen diese die Anforderungen der EN 61508 erfüllen.
Die Norm verlangt, dass die Sicherheitsintegrität als Sicherheits-Integritätslevel (SIL) angegeben werden muss, wenn sicherheitstechnische Funktionen von einem E/E/EP System ausgeführt werden sollen.

Die Sicherheitsintegrität (safety integrity) ist definiert als „*die Wahrscheinlichkeit, dass ein sicherheitstechnisches System die erforderliche sicherheitstechnische Funktion unter allen festgelegten Bedingungen innerhalb eines festgelegten Zeitraumes ausführt*“.[4]

Ein Sicherheits-Integritätslevel (SIL) ist eine von vier diskreten Stufen, wobei jede Stufe einem Bereich für die Ausfallwahrscheinlichkeit einer Sicherheitsfunktion entspricht. SIL 1 stellt die niedrigste Stufe dar, SIL 4 die

[4] EN 61508: a.a.O., Abschnitt 3.5.4

höchste. Es ist zu beachten, dass ein Sicherheits-Integritätslevel eine Eigenschaft eines Systems oder Teilsystems darstellt und nicht die einer Komponente [83].

Die Sicherheitsintegrität der Software (software safety integrity) ist definiert als „*Teil der Sicherheitsintegrität eines sicherheitsbezogenen Systems, der sich auf systematisches Versagen mit gefahrbringender Versagensart, die der Software zuordenbar ist, bezieht*“.[5]

Tabelle 1: Ausfallgrenzwerte für eine Sicherheitsfunktion, die in der Betriebsart mit niedriger Anforderungsrate betrieben wird

Sicherheits-Integritätslevel (SIL)	**Mittlere Wahrscheinlichkeit eines gefahrbringenden Ausfalls bei Anforderung der Sicherheitsfunktion (PFD_{avg})**
4	$\geq 10^{-5}$ bis $< 10^{-4}$
3	$\geq 10^{-4}$ bis $< 10^{-3}$
2	$\geq 10^{-3}$ bis $< 10^{-2}$
1	$\geq 10^{-2}$ bis $< 10^{-1}$

[**Quelle: eigene Darstellung in Anlehnung an European Committee for Electrotechnical Standardization: *EN 61508 Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme*. 2010, Teil 1, S. 36**]

Die Spezifikation umfasst die sicherheitstechnischen Funktionen ebenso wie die Maßnahmen, die als Reaktion auf das Vorliegen bestimmter Bedingungen zu ergreifen sind, sowie die Zeit, die für diese Reaktion benötigt wird. Der Sicherheitsintegritätslevel gilt deshalb als Maß für die Zuverlässigkeit der sicherheitstechnischen Funktion, die gemäß der Spezifikation arbeitet [60].

[5] EN 61508: a.a.O., Abschnitt 3.5.5

Tabelle 2: Ausfallgrenzwerte für eine Sicherheitsfunktion, die in der Betriebsart mit hoher Anforderungsrate oder in der Betriebsart mit kontinuierlicher Anforderung betrieben wird[6]

Sicherheits-Integritätslevel (SIL)	**Mittlere Häufigkeit eines gefahrbringenden Ausfalls der Sicherheitsfunktion [h^{-1}] (PFH)**
4	$\geq 10^{-9}$ bis $< 10^{-8}$
3	$\geq 10^{-8}$ bis $< 10^{-7}$
2	$\geq 10^{-7}$ bis $< 10^{-6}$
1	$\geq 10^{-6}$ bis $< 10^{-5}$

[Quelle: eigene Darstellung in Anlehnung an European Committee for Electrotechnical Standardization: *EN 61508 Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme.* 2010, Teil 1, S. 37]

Je nach *Betriebsart (mode of operation)* werden die sicherheitstechnischen Funktionen nach drei Klassen von SIL identifiziert:

- **Betriebsart mit niedriger Anforderungsrate (low demand mode):** *wobei die Sicherheitsfunktion nur auf Anforderung ausgeführt wird, um die EUC (Equipment Under Control) in einen festgelegten sicheren Zustand zu überführen, und wobei die Häufigkeit von Anforderungen nicht mehr als einmal je Jahr beträgt, oder*
- **Betriebsart mit hoher Anforderungsrate (high demand mode):** wobei die *Sicherheitsfunktion nur auf Anforderung ausgeführt wird, um die EUC in einen festgelegten sicheren Zustand zu überführen, und wobei die Häufigkeit von Anforderungen mehr als einmal je Jahr beträgt, oder*

[6] Quelle: eigene Darstellung in Anlehnung an European Committee for Electrotechnical Standardization (CENELEC): EN 61508 - Funktionale Sicherheit sicherheitsbezogener elektrischer / elektronischer / programmierbarer elektronischer Systeme. 2010, Teil 1, S. 37

- **Betriebsart mit kontinuierlicher Anforderung (continuous mode):** *wobei die Sicherheitsfunktion die EUC in einem sicheren Zustand als Teil des normalen Betriebs hält*

Die EN 61508 stellt mehrere Verfahren vor, die zur Bestimmung des Sicherheits-Integritätslevels verwendet werden können. Keines der Verfahren eignet sich für alle Anwendungen. Das am besten geeignete Verfahren sollte ausgewählt werden [44].

ALARP-Methode
ALARP steht für *As Low As Reasonably Practicable*. Die ALARP-Methode kann sowohl eigenständig als auch zusammen mit anderen Methoden verwendet werden. Sie kann in einer qualitativen als auch quantitativen Art angewendet werden.

Quantitative Methode
Mit Hilfe von Fehlerbäumen wird das Gefährdungsmodell dargestellt. Diese Methode ist sowohl für einfache als auch komplexe Anwendungen geeignet und von besonderem Wert, wenn das tolerierbare Risiko in einer numerischen Art und Weise zu bestimmen ist (z. B. dass eine bestimmte Auswirkung nicht mit einer Häufigkeit größer als einmal in 10^4 Jahren auftreten sollte).

Risikograph-Methode
Diese Methode ermöglicht es, den Sicherheits-Integritätslevel aufgrund der Kenntnis der Risikofaktoren zu bestimmen. Die Art der Gefährdungssituation wird mithilfe einer Reihe von Parametern beschrieben, falls die sicherheitsbezogenen Systeme versagen oder nicht zur Verfügung stehen.

Analyse der Schutzebenen (LOPA – Layer of Protection Analysis)
LOPA analysiert Gefährdungen, um festzustellen, ob Sicherheitsfunktionen erforderlich sind. Falls eine zusätzliche Risikominderung erforderlich ist und wenn diese in Form eines sicherheitsbezogenen E/E/PE-Systems erfolgt, ermöglicht die LOPA-Methode die Bestimmung des geeigneten SIL. Für jede

Gefährdung wird ein geeigneter Sicherheits-Integritätslevels zur Verringerung der Risiken auf ein tolerierbares Niveau bestimmt.

Matrix des Ausmaßes des gefährlichen Vorfalls
Bei dieser Methode wird angenommen, dass eine Größenordnung der Risikominimierung erreicht wird, wenn eine Schutzebene hinzugefügt wird. Außerdem wird angenommen, dass sicherheitsbezogene E/E/EP-Systeme und andere Maßnahmen zur Risikominimierung unabhängig von der Ursache der Anforderung und unabhängig voneinander sind.

Der erreichbare Sicherheits-Integritätslevel ist begrenzt durch die Fehlertoleranz der Hardware, den Anteil an ungefährlichen Ausfällen, Architektureinschränkungen, Ausfallwahrscheinlichkeit der Funktion im Anforderungsfall und von Instandhaltungsintervallen [20].

3. ISO 26262 Road vehicles – Functional safety

Einführung

Die ISO 26262 ist eine speziell für die Automobilindustrie entwickelte Norm für elektrische/elektronische (kurz E/E) Systeme und Komponenten in Personenkraftfahrzeugen bis zu 3,5 t, die eine Sicherheitsfunktion ausführen. Nach IEC 61508 würde die Validierung erst nach Fertigstellung erfolgen. Bei einem Massenprodukt wie einem PKW stellte diese Tatsache die Automobilhersteller vor großen Problemen.

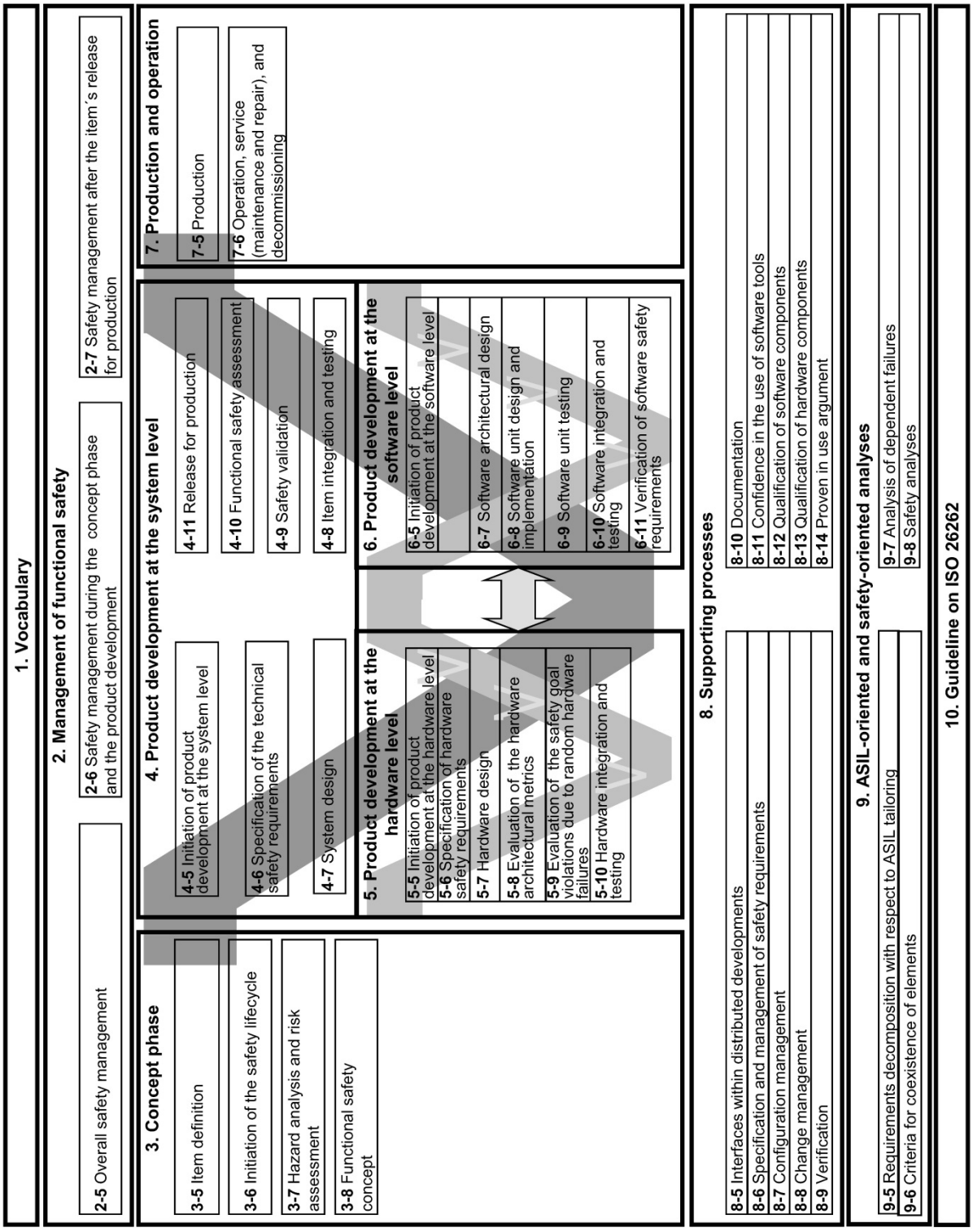
Im Jahr 2002 wurde erstmals eine eigenständige Norm gefordert, 2005 übernahm die Internationale Organisation für Normung (ISO) die Arbeit an dieser, die im Grunde eine Adaptierung der EN 61508 an die spezifischen Gegebenheiten im Automobilbereich darstellte.

Die ISO 26262 gliedert sich in 10 Teilen, wovon die Teile 1 – 9 am 14. November 2011 in Kraft getreten sind. Teil 10 wurde nach einer kurzen Überarbeitungszeit am 1. August 2012 veröffentlicht.

Die Norm behandelt u.a. auch mögliche Gefahren, die auftreten können, falls diese sicherheitsrelevanten E/E-Systeme ausfallen. Außeracht gelassen werden Risiken, die durch Stromschlag, Feuer, Rauch, Hitze, Radioaktivität, Giftigkeit, Entzündbarkeit, Reaktionsvermögen, Korrosion, Energiefreisetzung und ähnlichen, es sei denn, diese werden direkt durch einen Ausfall der E/E-Systeme verursacht [35].

Überblick und Gliederung

Abbildung 5: Übersicht über die ISO 26262



[**Quelle: International Organization for Standardization: *ISO 26262 Road vehicles. Functional safety.* 2011, Teil 1**]

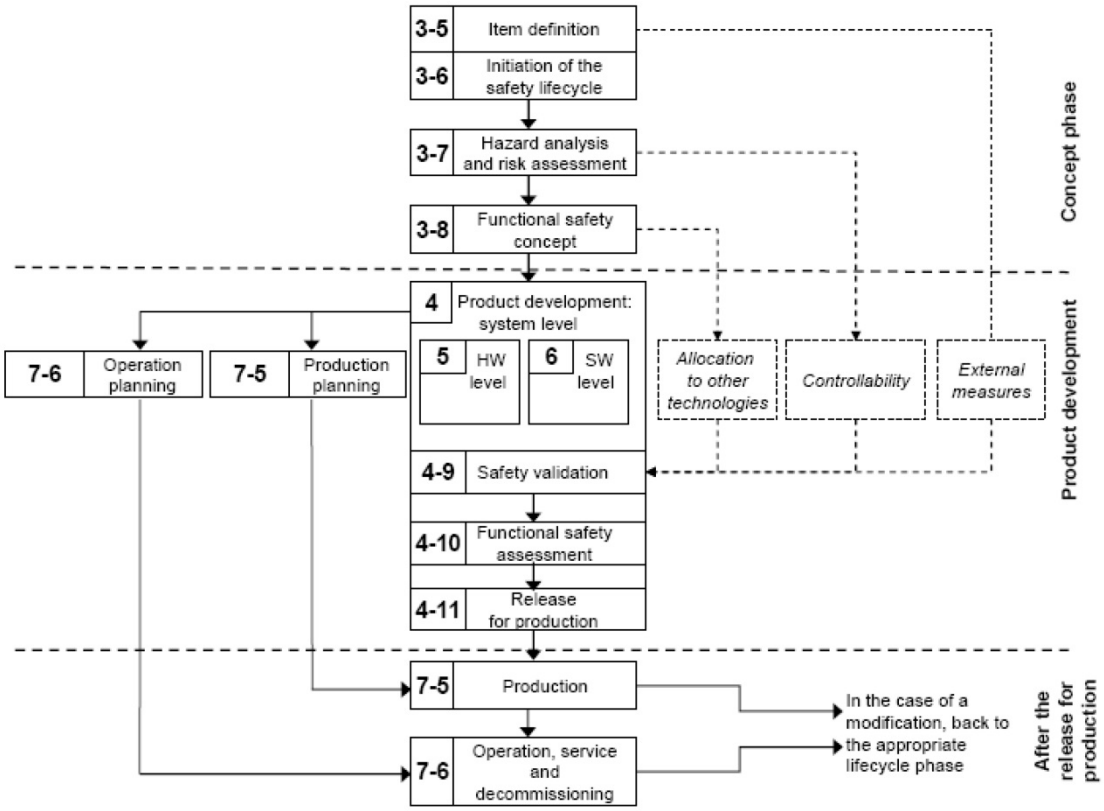
ISO 26262-1:2011 Part 1: Vocabulary

Begriffe, Definitionen und Abkürzungen werden spezifiziert, die in allen Teilen der Norm Verwendung finden.

ISO 26262-2:2011 Part 2: Management of functional safety

Es werden Anforderungen für das Management der Funktionalen Sicherheit definiert, die sowohl unabhängig vom Projekt die beteiligte Organisationen betrifft, als auch projektabhängig die Managementtätigkeiten im Sicherheitslebenszyklus.

Abbildung 6: Phasen des Sicherheitslebenszyklus nach ISO 26262



[Quelle: eigene Darstellung in Anlehnung an International Organization for Standardization: *ISO 26262 Road vehicles. Functional safety.* 2011, Teil 1]

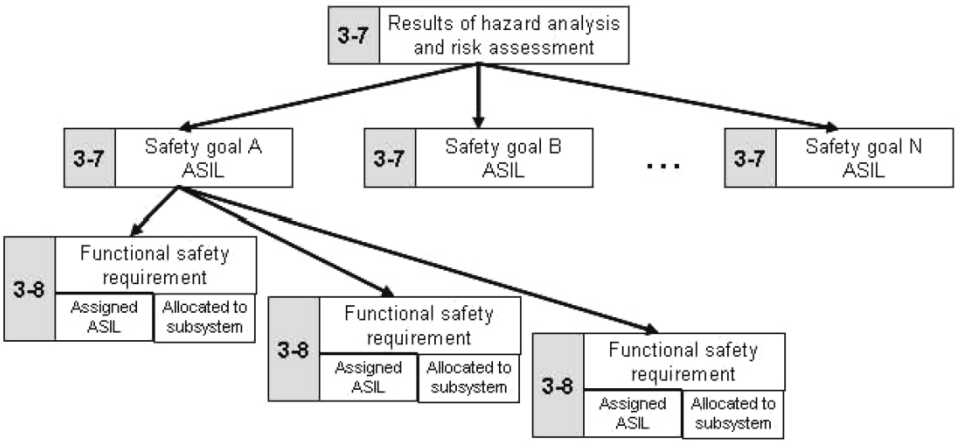
Die Organisation des Projektmanagements über den gesamten Produktlebenszyklus hinweg wird beschrieben, sowie Absicherungsmaßnahmen zum Nachweis der Normkonformität. Eine umfangreiche Dokumentationspflicht während jedes Entwicklungsschrittes wird ebenso festgehalten.

Teile 3 bis 7 sind als übergreifende Einheit zu verstehen. Sie definieren den *Produktlebenszyklus*, beginnend mit der Konzeptphase über die Produktentwicklung auf System-, Hardware- und Softwareebene bis zur Produktion, Betrieb und Außerbetriebnahme. Der Produktlebenszyklus besteht aus den Phasen *concept*, *product development* und *after the release for production*.

ISO 26262-3:2011 Part 3: Concept phase

Hier werden Anforderungen für die Konzeptionsphase der Fahrzeuganwendungen besprochen, die vor allem die Objekt Definition, Einführung des Sicherheitslebenszyklus, Risikoanalyse und -beurteilung und das Konzept der Funktionalen Sicherheit betreffen.

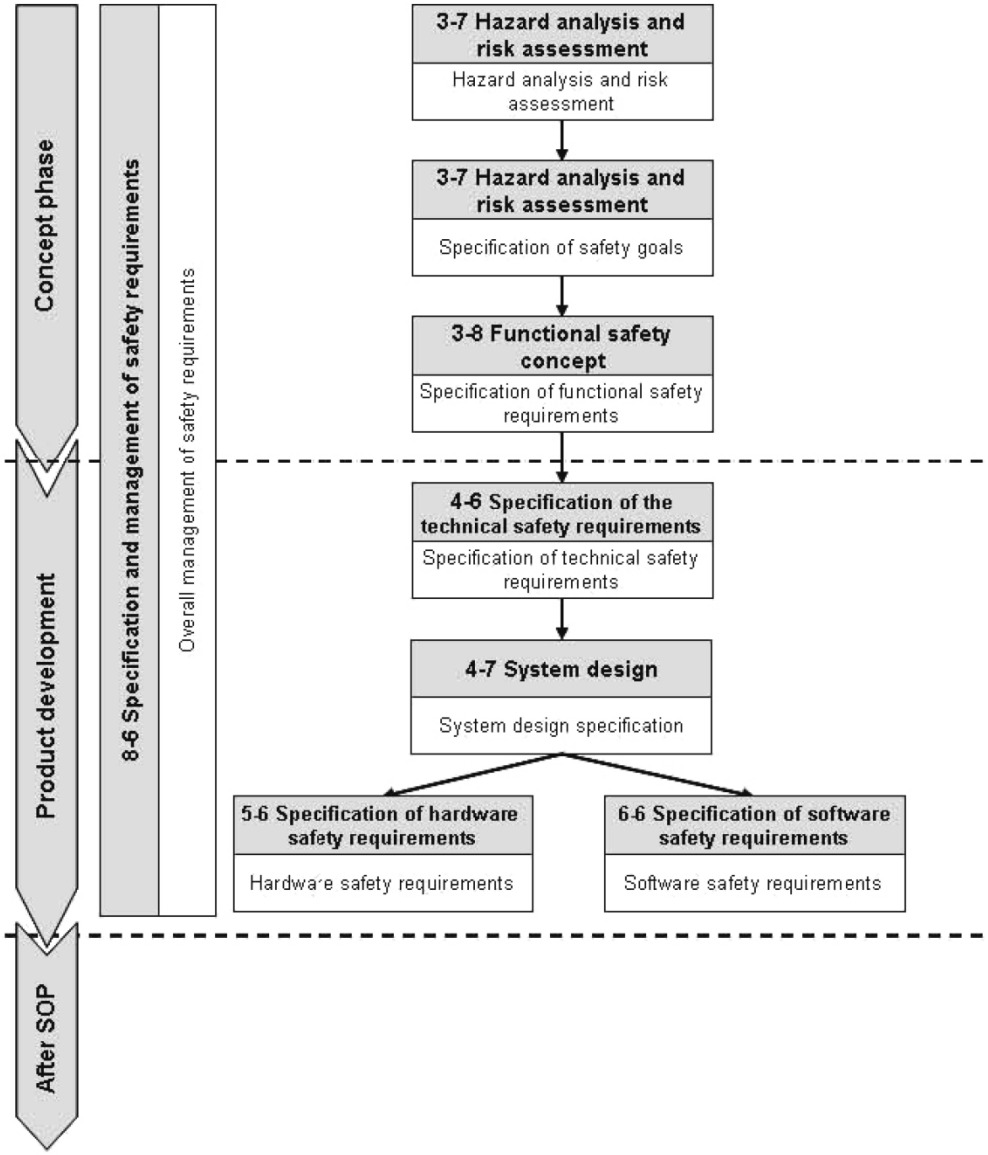
Abbildung 7: Hierarchy of safety goals and functional safety requirements



[Quelle: International Organization for Standardization: *ISO 26262 Road vehicles. Functional safety.* 2011, Teil 3]

Zuerst wird das Bauteil bzw. das System definiert. Danach erfolgen Gefährdungs- und Risikoanalyse, woraus dann Sicherheitsziele abgeleitet und den Automotiv Safety Integrity Level (ASIL) zugeordnet werden [35]. In dieser Phase müssen alle möglichen Fehlfunktionen in ihrer Gesamtheit und alle Situationen, in denen ihr Auftreten relevant sein kann, erfasst werden. Die Konzeptphase wird abgeschlossen mit der Erstellung eines funktionalen Sicherheitskonzeptes.

Abbildung 8: Structure of the safety requirements



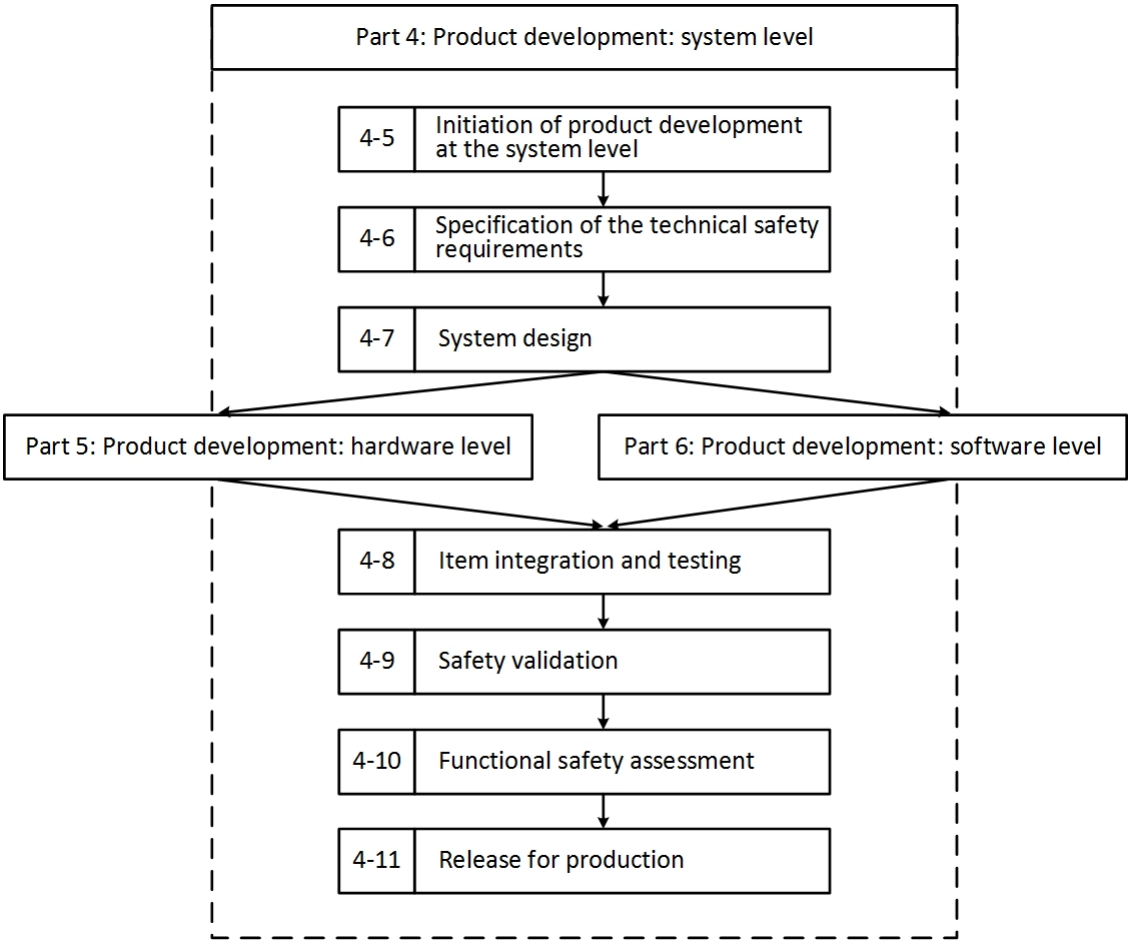
[Quelle: International Organization for Standardization: *ISO 26262 Road vehicles. Functional safety.* 2011, Teil 3]

ISO 26262-4:2011 Part 4: Product development: system level

Dieser Teil befasst sich mit der Betrachtung von Gefährdungen und der Einschätzung von Risiken, die im Zusammenhang mit der funktionalen Sicherheit von sicherheitsbezogenen Fahrzeugsystemen bestehen [32]. Es werden Anforderungen der Produktentwicklung auf Systemlevel definiert. Diese beinhalten die technische Sicherheit, das technische Sicherheitskonzept, das

Systemdesign, die Integration und Prüfung, die Sicherheitsvalidierung, die Einschätzung der Funktionalen Sicherheit und die Produktveröffentlichung.

Abbildung 9: Reference phase model for the development of a safety-related item



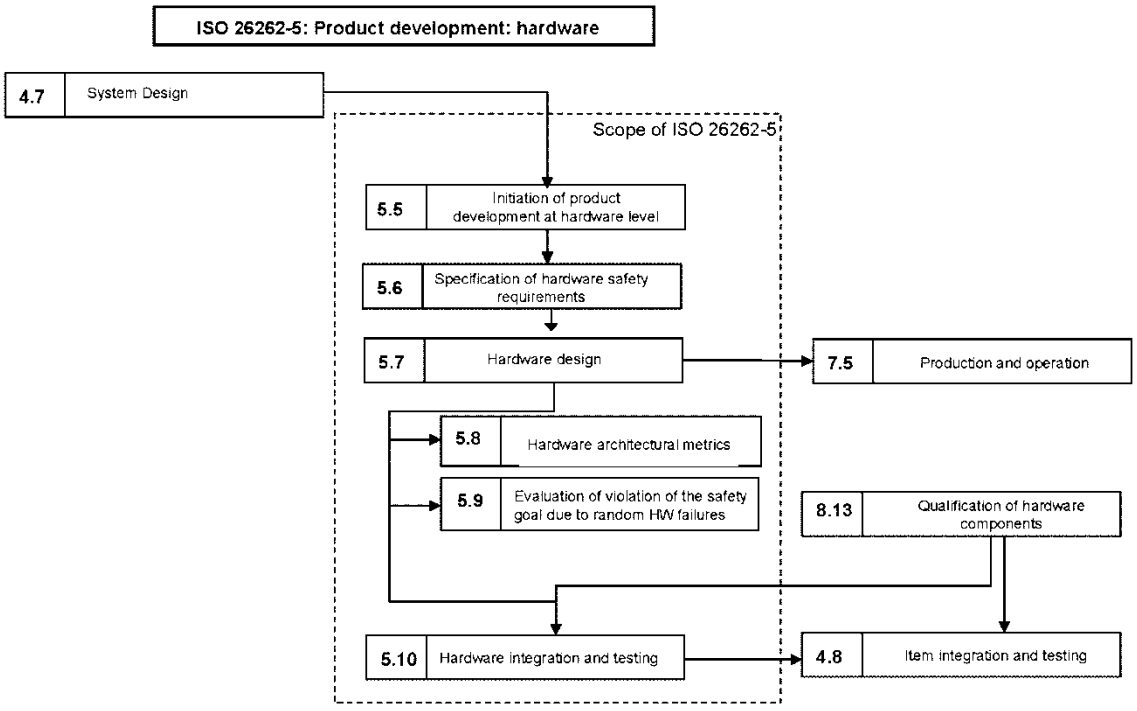
[Quelle: eigene Darstellung in Anlehnung an International Organization for Standardization: *ISO 26262 Road vehicles. Functional safety.* 2011, Teil 3]

Das funktionale Sicherheitskonzept aus Teil 3 wird aufgegriffen und erweitert. Dabei werden aus realisierungsunabhängigen funktionalen Sicherheitsanforderungen realisierungsabhängige technische Sicherheitsanforderungen abgeleitet, die sich auf Hardware- oder Softwareelemente beziehen, welchen sie zugeordnet werden.

Handhabungen, Arbeitsweisen und zu erreichende Arbeitsergebnisse sind vorgeschrieben und durch die drei Bemessungsgrundlagen *optional*, *recommended* und *highly recommended* klassifiziert.

Abbildung 9 zeigt die erforderlichen Schritte bei der der Systementwicklung. Auf die Einleitung der Produktentwicklung folgen die Spezifikation der technischen Sicherheitsanforderungen und das Systemdesign. Die Ergebnisse dieses Prozesses beinhalten auch die technischen Sicherheitsanforderungen für Hardware und Software, die in Teil 5 und Teil 6 behandelt werden. Die Systemintegration geschieht nach Entwicklung der Hardware- und Software-Elemente. Das so entstandene System wird getestet, validiert und dessen funktionale Sicherheit wird beurteilt. Die Produktionsfreigabe erfolgt nach erfolgreicher Prüfung.

Abbildung 10: Product Development Hardware



[Quelle: International Organization for Standardization: *ISO 26262 Road vehicles. Functional safety.* 2011, Teil 5]

ISO 26262-5:2011 Part 5: Product development: hardware level
Teil 5 spezifiziert die Anforderung der Produktentwicklung auf dem Hardware Level. Diese betreffen unter anderem das Hardware Design, Bewertung der Hardwarearchitektur in Bezug auf Behandlung zufälliger Hardwarefehler, Hardware Integration und Prüfung und die Beurteilung der Verletzungen der Sicherheitsvorgaben durch zufällige Hardwareausfälle. Es muss ein Nachweis erbracht werden, dass die Wahrscheinlichkeit einer Verletzung der Sicherheitsziele durch zufällige Hardwareausfälle noch tolerierbar ist [32].

ISO 26262-6:2011 Part 6: Product development: software level
Die Produktentwicklung auf dem Software Level ist Inhalt des 6. Teils. Im Mittelpunkt stehen die Spezifikation der Anforderung an die Softwaresicherheit, Software Architektur Design, Software Unit Design und Implementation, Software Unit Prüfung, Software Integration und Prüfung und die Verifikation der Software Sicherheitsanforderungen. Sicherheitsanforderungen von Software beziehen sich auf softwarebasierte Funktionen. Eine Fehlfunktion kann zu einer Verletzung von technischen Sicherheitsanforderungen führen [32].

ISO 26262-7:2011 Part 7: Product and operation
Dieser Teil der Norm befasst sich mit den Anforderungen der Produktion, des Betriebs, der Instandhaltung und Reparatur und der Außerbetriebnahme. Der Aspekt der Außerbetriebnahme ist in die ISO 26262 aufgenommen worden, um sicherzustellen, dass die Stilllegung bzw. die Entsorgung eines Produktes unter Einhaltung der Sicherheitsaspekte stattfindet und nicht in einem undefinierten Bereich angesiedelt wird. Des Weiteren wird die Vorgehensweise für sicherheitsrelevante Systeme zum Erstellen der benötigten Produktions- und Installationspläne beschrieben.

ISO 26262-8:2011 Part 8: Supporting processes
Der 8. Teil erörtert Unterstützende Prozesse zu Interfaces im Rahmen verteilter Entwicklung, allgemeines Management von Sicherheitsanforderungen, Konfigurationsmanagement, Veränderungsmanagement, Verifikation, Dokumentation, Vertrauen in der Verwendung von Software Tools,

Qualifikation von Software und Hardware Komponenten und das Argument der Langzeiterfahrung (Betriebsbewährung[7]).

ISO 26262-9:2011 Part 9: ASIL-oriented and safety-oriented analyses
Teil 9 diskutiert Regeln zur ASIL-Dekomposition, die übliche Methodik bei der Durchführung solcher Sicherheitsuntersuchungen in Abhängigkeit von der Phase des Produktlebenszyklus, Kriterien für die Koexistenz von Elementen mit unterschiedlichen ASIL-Einstufungen, Analyse von Gleichartigen Ausfällen und Sicherheitsanalysen.
Unter ASIL-Dekomposition versteht man die Zerlegung eines entsprechenden Systems in kleinere Bestandteile, wodurch Funktionen getrennt betrachtet werden können. Durch sinnvolles und geschicktes Vorgehen kann eine verbesserte und optimierte Architektur erreicht werden [73]

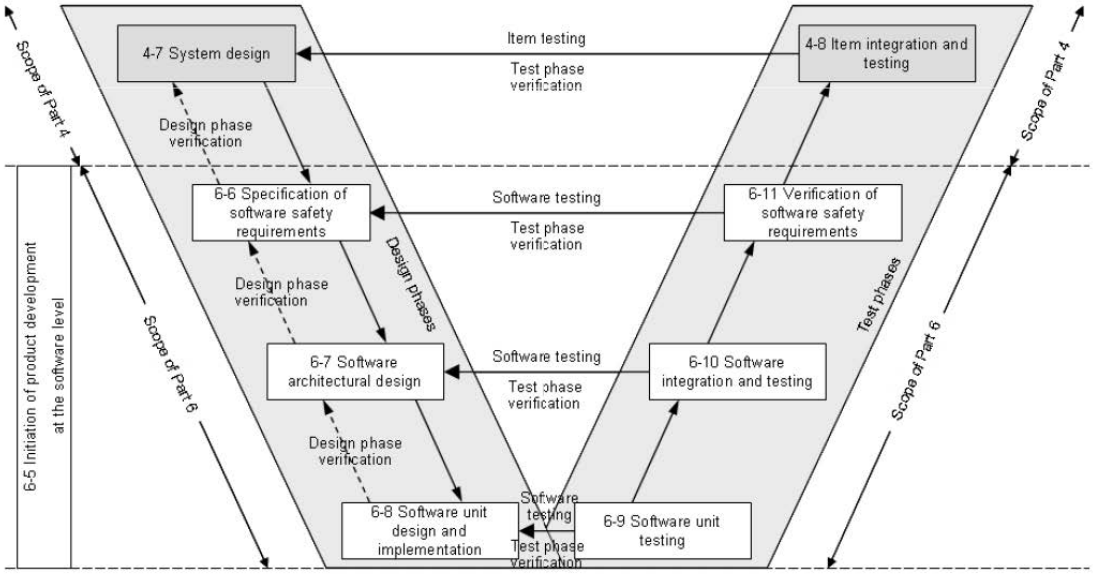
ISO 26262-10:2012 Part 10: Guideline
Der letzte Teil enthält Anwendungsbeispiele, Anmerkungen und weiterführende Information zur Norm. Mit Hilfe einer exemplarischen ASIL-Bewertung, Beispiele zu den zuvor vorgestellten Konzepten und Prozessen wird versucht, die ISO 26262 besser verständlich für Unternehmen zu machen.

V-Modell
In der ISO 26262 werden die Entwicklungsprozesse auf System-, Hardware- und Softwareebene als geschachtelte V-Modelle realisiert. Das V-Modell ist eine Weiterentwicklung des Wasserfallmodells[8] und hat sich bei Softwareentwicklungen als Standard durchgesetzt.

[7] proven in use argument

[8] lineares Vorgehensmodell in der Softwareentwicklung, das sich in Phasen gliedert, welche streng nacheinander ohne Iterationen durchlaufen werden

Abbildung 11: V-Modell der Softwareebene nach ISO 26262



[**Quelle: International Organization for Standardization: *ISO 26262 Road vehicles. Functional safety.* 2011, Teil 6**]

Der Buchstabe V steht für *Validation and Verification*. Genau wie beim Wasserfallmodell hat jede Phase vordefinierte Start- und Endpunkte mit eindeutig definierten Ergebnissen. Die Phasenergebnisse sind bindende Vorgaben für die nächsttiefere Projektphase. Der linke Teil des Modells spiegelt die Projektdefinition wieder. Die linken Phasen werden deshalb auch Spezifizierungsphasen genannt, die mit der Implementierung abschließen. Der rechte Teil des Modells, der Realisierungsphase, wird in mehreren Testphasen unterteilt. Den spezifizierenden Phasen stehen jeweils testende Phasen gegenüber, die Gegenüberstellung soll zu einer möglichst hohen Testabdeckung führen, weil die Spezifikationen der jeweiligen Entwicklungsstufen die Grundlage für die Tests bzw. Testfälle in den entsprechenden Teststufen sind.

4. Automotive Safety Integrity Level (ASIL)

Der Automotive Safety Integrity Level (ASIL) ist ein Maß für die Sicherheitsrelevanz eines Bauteils. Eine Fehlfunktion kann in der Regel in jedem Bauteil auftreten, jedoch hat jede Fehlfunktion unterschiedlich schwerwiegende Konsequenzen, die auch abhängig von der jeweiligen Situation zum Eintrittszeitpunkt ist.

Gefährdungs- und Risikoanalyse (*hazard analysis and risk assessment*)

Um eine ASIL-Einstufung durchführen zu können, müssen zunächst Risiko und Gefährdungen nach folgenden Kriterien klassifiziert werden:

Tabelle 3: Schwere eines möglichen Schadens (Severity S)[9]

Klassifizierung	Beschreibung
S0	Keine Verletzungen
S1	Leichte bis mittlere Verletzungen
S2	Schwere Verletzung, Überleben wahrscheinlich
S3	Lebensgefährliche Verletzungen, Überleben unwahrscheinlich

[Quelle: eigene Darstellung in Anlehnung an International Organization for Standardization: *ISO 26262 Road vehicles. Functional safety.* 2011, Teil 3]

[9] S0 entspricht AIS 0 (Abbreviated Injury Scale), S1 entspricht AIS 1&2, S2 entspricht AIS 3&4, S3 entspricht AIS 5&6

Tabelle 4: Häufigkeit der Fahrsituation (Probability of Exposure E)

Klassifizierung	Beschreibung
E0	Unvorstellbar
E1	Sehr niedrige Wahrscheinlichkeit
E2	Niedrige Wahrscheinlichkeit
E3	Mittlere Wahrscheinlichkeit
E4	Hohe Wahrscheinlichkeit

[**Quelle: eigene Darstellung in Anlehnung an International Organization for Standardization: *ISO 26262 Road vehicles. Functional safety.* 2011, Teil 3**]

Tabelle 5: Beherrschbarkeit durch den Fahrer (Controllability C)[10]

Klassifizierung	Beschreibung
C0	Im allgemeinen Beherrschbar
C1	Einfach beherrschbar
C2	Normalerweise beherrschbar
C3	Schwierig oder nicht beherrschbar

[**Quelle: eigene Darstellung in Anlehnung an International Organization for Standardization: *ISO 26262 Road vehicles. Functional safety.* 2011, Teil 3**]

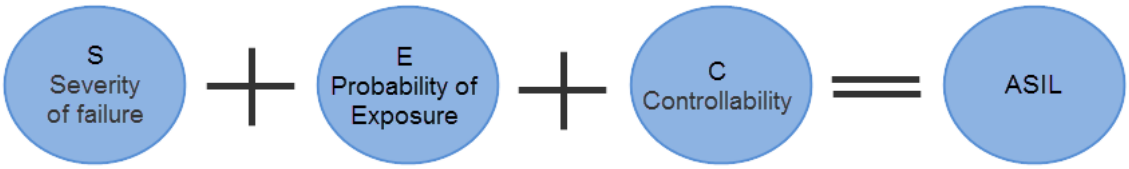
Die Einschätzung des Risikos und der damit verbundenen Einstufung des ASIL basiert auf der Kombination der *Schwere des möglichen Schadens*, der

[10] C0 ist nicht spezifiziert, C1 entspricht 0,01, C2 entspricht 0,1, C4 entspricht 1; C1 bedeutet, dass 99% aller Fahrer oder Verkehrsteilnehmer die Situation beherrschen können. Bei C2 wird von mehr als 90% und bei C3 von weniger als 90% ausgegangen. In Bezug auf den Fahrer wird davon ausgegangen, dass er fahrtüchtig und erfahren (im Sinne des Besitzes einer gültigen Fahrerlaubnis) ist.

Häufigkeit der Fahrsituation und der *Beherrschbarkeit durch den Fahrer*. ASIL befasst sich nicht mit der Technologie, die im System verwendet wird, sondern konzentriert sich nur auf die Auswirkungen auf den Fahrer und den anderen Verkehrsteilnehmern.

ASIL-Klassifikation

Abbildung 12: ASIL-Klassifikation



[Quelle: eigene Darstellung]

Aus den zuvor genannten Parametern ergibt sich die ASIL-Klassifikation auf einer Skala von A bis D, wobei die Stufe A die niedrigste und D die höchste Einstufung ist. Je höher die Einstufung, desto anspruchsvoller bzw. effektiver müssen die Maßnahmen und Techniken zur Risikominimierung sein. Nicht sicherheitsrelevante Systeme werden mit QM (Qualitätsmanagement) eingestuft.

Abbildung 13: Automotive Safety Integrity Level

[Quelle: eigene Darstellung]

Tabelle 6: ASIL Einstufung nach ISO 26262:3

		Controllability C		
Severity S	**Exposure E**	**C1**	**C2**	**C3**
S1	**E1**	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	A
	E4	QM	A	B
S2	**E1**	QM	QM	QM
	E2	QM	QM	A
	E3	QM	A	B
	E4	A	B	C
S3	**E1**	QM	QM	A
	E2	QM	A	B
	E3	A	B	C
	E4	B	C	D

[Quelle: eigene Darstellung in Anlehnung an International Organization for Standardization: *ISO 26262 Road vehicles. Functional safety.* 2011, Teil 3]

Entsprechend der Gefährdungen werden Sicherheitsziele als höchste Ebene von Sicherheitsanforderungen bestimmt. Es werden jedoch keine technischen Realisierungen beschrieben, sondern nur funktionale Zielvorgaben zur Vermeidung von Risiken und Gefährdungen.
Ähnliche Sicherheitsziele für verschiedene Gefährdungen können kombiniert werden, wobei aber zu beachten ist, dass die jeweils höchste ASIL-Einstufung übernommen werden muss.

5. Kritische Bewertung SIL und ASIL

Es darf nie vergessen werden, dass alle Einstufungen von SIL und ASIL auf Wahrscheinlichkeitskenngrößen beruhen. Hohe Systemkomplexität, besonders in Kombination mit Software macht eine SIL-Einstufung schwierig bis unmöglich. Die allgemeine Akzeptanz kann ebenso ein wichtiger Faktor für den Betrieb von Sicherheitsrelevanten Systemen werden. Risiko und Wahrscheinlichkeit für Ausfälle können stark reduziert werden, jedoch kann ein Unfall nie völlig ausgeschlossen werden.

Falls es bei einem System, das nach entsprechender SIL gebaut und erfolgreich zugelassen wurde, bei der Einführung oder kurz danach doch zu einem fatalen Unfall kommen sollte, wird eine Argumentation für den Weiterbetrieb des Systems schwierig.

Ein gutes Beispiel hierfür wäre etwa eine Bahnsicherungsanlage, die nach SIL 3 errichtet wird und bei der Einweihung prominente Teilnehmer zu Tode kommen. Falls dieses System korrekt nach EN 61508 errichtet wurde und die Sicherheitsfunktionen SIL 3 genügen, würde statistisch gesehen der nächste gefahrbringende Ausfall erst wieder in ca. 1100 Jahren auftreten. Diese Argumentation ist zwar schlüssig und das System gilt weiterhin als sehr sicher, aber eine Akzeptanz durch Kunden oder Bevölkerung wird trotzdem problematisch.

Ein Maß für die Zuverlässigkeit des Gesamtsystems bzw. eine quantifizierbare Messgröße oder Einheit für Sicherheit ist sehr wünschenswert, existiert aber leider (noch?) nicht.

6. FMECA

Die Abkürzung FMECA steht für Failure Modes, Effects, and Criticality Analysis. Eine mögliche deutsche Übersetzung ist Fehlzustandsart-, -auswirkungs- und -kritizitätsanalyse [19]. Sie ist eine vorbeugende Methode zur Identifikation von Problemen, deren Risiken und Auswirkungen.
FMECA basiert auf FMEA (Failure Modes and Effects Analysis), jedoch speziell für sicherheitsrelevante Systeme vorgesehen. Sie erweitert die FMEA um eine quantitative Analyse, sie isoliert Teilbereiche eines Systems, die besonders kritisch sind und betrachtet besonders schwerwiegende Fehler.
Die Ziele der FMECA sind die Erkennung von Risiken und Problembereichen, Erkennen von Risikopotentialen, Quantifizierung von Risiken sowie die Findung von Abhilfemaßnahmen. FMECA bietet ein systematisches Vorgehen bei der Analyse eines Systems, um mögliche Fehlzustandsarten, ihre Ursachen und ihre Auswirkungen auf das Systemverhalten zu ermitteln. Sie beinhaltet außerdem die Klassifizierung der Schwere der Ausfallarten, um die Einstufung der Dringlichkeit von Abhilfemaßnahmen zu ermöglichen.
Das Verfahren der FMECA ist weitgehend formalisiert und daher vollständig und systematisch durchführbar. Die Sicherheit der Aussage ist von der gründlichen Durchführung und anschließender kritischer Bewertung abhängig.

Historische Entwicklung
FMECA wurde in den 1940ern vom U.S. Militär entwickelt. Im Jahr 1949 wurde erstmals die FMEA-Methoden als United States Military Procedure unter MIL-P 1629 publiziert. 1974 wurde das Papier nochmals überarbeitet und als MIL-STD-1629 (SHIPS) veröffentlicht und sechs Jahre später durch MIL-STD-1629A ersetzt [14], bevor es 1998 ersatzlos widerrufen wurde. Trotz dieser Annullierung ist MIL-STD-1629A für militärische und raumfahrttechnische Anwendungen noch heute weit verbreitet und in Verwendung [77].
Die NASA setzte erstmals in den 1960ern die FMECA Methoden für ihr Apollo Programm ein [84], anschließend auch für das Viking, Voyager, Magellan, Galileo und Skylab Programm. FMEA und FMECA fanden wenig später auch

Verbreitung in der Automobilindustrie und in der zivilen Luftfahrt[11]. 1967 veröffentlichte die SAE (Society for Automotive Engineers) erstmals eine Publikation [78], die sich mit FMECA befasst.
Nachdem Ford Motor Company in den 1970ern große Probleme mit Ihrem Modell Ford Pinto hatten, begannen sie FMECA einzusetzen. Später zogen auch die beiden anderen US-amerikanischen Automobilunternehmen General Motors und Chrysler nach. Seit den 1980ern ist FMEA und FMECA weltweit bei allen großen Automobilherstellern in Verwendung[84].

FMECA Verwendungszweck und Einsatzzeitpunkt

Zweck der Ausfallbedeutungsanalyse ist es, die relative Bedeutung jeder Ausfallauswirkung als eine Hilfe für die Entscheidungsfindung zu quantifizieren, so dass mit einer Kombination aus Kritizität und Schwere Prioritäten für Aktionen zur Milderung oder Minimierung der Auswirkungen bestimmter Ausfälle gesetzt werden können.[12]

Es ist empfehlenswert, eine FMECA möglichst früh im Designprozess zu starten, da hier der Einfluss auf die Ausfallsicherheit am größten ist. Auch aus wirtschaftlicher Sicht ist eine möglichst frühe Fehlervermeidung wünschenswert, denn je später ein Fehler entdeckt wird, desto schwieriger und kostenintensiver wird seine Korrektur sein.
Die Durchführung der FMECA soll nicht nur bei der Auswahl von Design Alternativen mit hoher Zuverlässigkeit und hoher Sicherheit in frühen Design Phasen helfen, sondern auch möglichst früh Kriterien für Testpläne und Anforderungen für Testgeräte entwickeln. Es bildet die Grundlage für qualitative Zuverlässigkeits- und Verfügbarkeitsanalysen und schafft eine Basis für die Planung der Instandhaltung.

[11] Die zivile Luftfahrtindustrie verwendet heutzutage anstatt FMECA eine Kombination vom FMEA und Fault Tree Analysis, da diese Methode von der Federal Aviation Administration (FAA - die Bundesluftfahrtbehörde der Vereinigten Staaten) bevorzugt wurde.

[12] EN 60812 Analysetechniken für die Funktionsfähigkeit von Systemen – Verfahren für die Fehlzustandsart- und -auswirkungsanalyse (FMEA), 2006, Abschnitt 5.3.1, S. 17f

Vorgehensweise FMECA

Grundsätzlich unterscheidet man zwischen zwei Vorgehensweisen, den Bottom-up und den Top-Down Approach. Bottom-up Approach wird verwendet, wenn das Systemkonzept bereits feststeht. In der niedrigsten Gliederungsebene wird jede Komponente nach einander studiert. Da alle Komponenten berücksichtigt werden, wird davon ausgegangen, dass die Analyse vollständig ist.
Top-down Approach wird hauptsächlich in frühen Design Phasen verwendet, bevor die gesamte Systemstruktur feststeht. Die Analyse ist normalerweise funktionsorientiert und startet bei den Hauptfunktionen des Systems – und wie diese ausfallen könnten. Funktionsausfälle mit wesentlichen Auswirkungen werden normalerweise priorisiert, weshalb die Analyse nicht notwendigerweise vollständig ist. Top-Down kann auch auf existierende Systeme angewendet werden, um auf Problemfelder zu fokussieren.

Voraussetzungen für FMECA

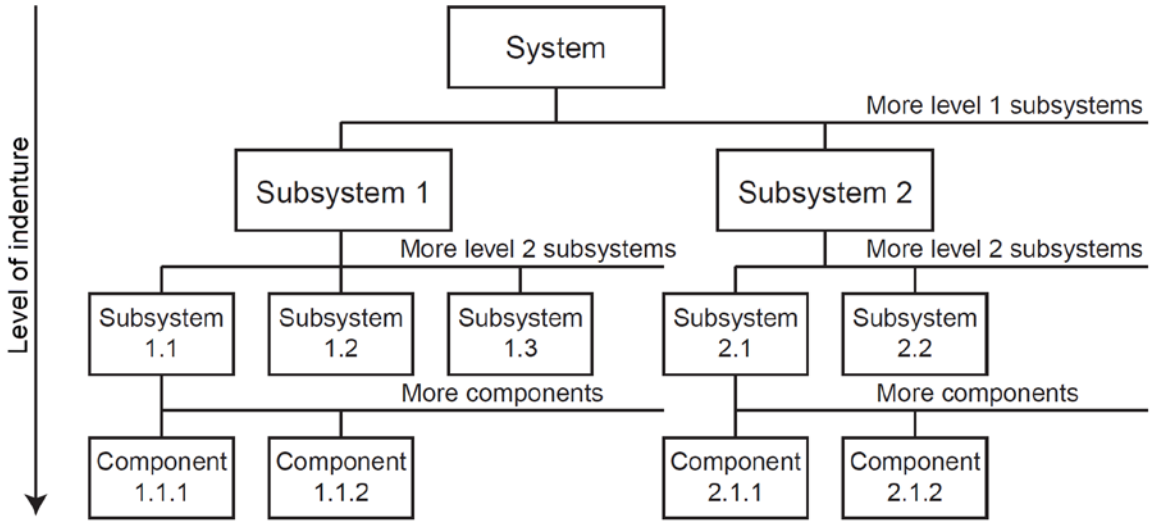
Das zu analysierende System soll möglichst genau definiert werden. In dieser Definition werden die Systemgrenzen festgelegt. Der Einsatzzweck und die Funktionen des Systems werden beschrieben und die zu beachtende Betriebs- und Umgebungsbedingungen werden festgehalten.
Der nächste Schritt ist die Sammlung von verfügbaren Informationen, die das zu untersuchende System beschreiben. Dazu gehören Abbildungen, Spezifikationen, Schaltbilder, Komponentenliste, Interface Beschreibung, Funktionale Darstellung, usw. Den Abschluss bildet die Recherche und Zusammentragung von verfügbaren Informationen zu vorhergehenden und ähnlichen Designs sowohl von internen als auch externen Quellen. Von Interesse sind auch FRACAS Daten (failure reporting, analysis and corrective action system data), Befragung von Entwurfspersonal, Belegschaft und Wartungsmannschaft, Zulieferer, usw.

Systemstrukturanalyse

In Systemstrukturanalyse wird das zu untersuchende System in überschaubare Einheiten, typischerweise Funktionselemente, zerlegt. Der Detailgrad der Zerlegung hängt von der Zielvorgabe der Analyse ab. Idealerweise passiert dies auf möglichst hoher hierarchischer Stufe. Erst beim Auftreten von inakzeptablen Konsequenzen müssen Elemente weiter zerlegt werden. Auf einem niedrigeren

Level ist es einfacher, Fehlzustandsart und Ausfallursache geeignet zu identifizieren. Startet man anfangs bereits auf zu niedrigem hierarchischen Level, erhält man zwar eine komplette Analyse, jedoch kann das auch gleichzeitig eine unnötige Verschwendung von Zeit und Geld sein. Die Systemstruktur kann als hierarchisches Baumdiagramm dargestellt werden.

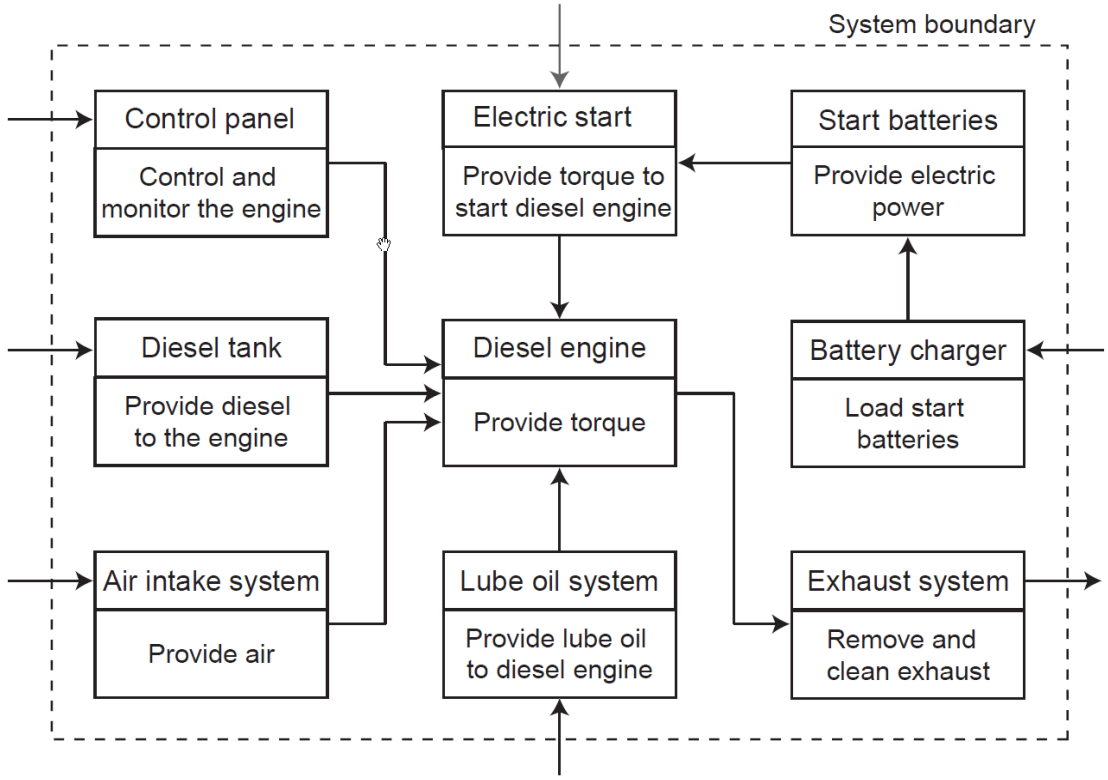
Abbildung 14: Systemstruktur als hierarchisches Baumdiagramm



[Quelle: M. Rausand, A. Høyland: *System Reliability Theory. Models, Statistical Methods, and Applications*. Wiley & Sons, 2.Auflage, Hoboken, NJ, 2004, S. 91]

In frühen Design Phasen existiert normalerweise noch keine Entscheidung über eine physikalische bzw. technische Realisierung eines Systems. In der Regel wird mit einem Set von gewünschten Funktionen begonnen, das Entwicklungsziel ist dann ein System, das die gewünschten Funktionen erfüllt. Ein solches System kann dann als Funktionsblockdiagramme dargestellt werden.

Abbildung 15: Beispiel eines Funktionsdiagramms eines Dieselmotors



[**Quelle: M. Rausand, A. Høyland: *System Reliability Theory. Models, Statistical Methods, and Applications*. Wiley & Sons, 2.Auflage, Hoboken, NJ, 2004, S. 77**]

Definition Fehlzustand

Zustand einer Einheit, in dem sie unfähig ist, eine geforderte Funktion zu erfüllen, wobei die durch Wartung oder andere geplante Handlungen bzw. durch das Fehlen äußerer Mittel verursachte Funktionsunfähigkeit ausgeschlossen ist.[13]

Bestimmung Fehlzustand und Ausfallursache

Ein System oder eine Komponente kann viele verschiedene Fehlzustände annehmen. Jeder (vermutlich) unabhängiger Fehlzustand sollte für jedes Element bestimmt werden. Typische Fehlzustände sind z.B. verzögerte oder verfrühte Ausgabe, Aussetzbetrieb, Leistungsverlust, fehlerhafte Ausgabe (unter bestimmten Bedingungen), ungültige Ausgabe (unter allen Bedingungen), usw.

[13] EN 60812: a.a.O., Abschnitt 3.3, S. 6

Für jeden Fehlzustand müssen die wahrscheinlichsten Ursachen bestimmt und beschrieben werden. Ausfallursachen können durch die Analyse von Feldausfällen und Ausfällen in Prüfeinrichtungen ermittelt werden, auch das Einholen von Expertenmeinungen ist wichtig und erwünscht.

Ausfallwirkungen und Ausfallerkennungsmethoden
Eine Ausfallauswirkung ist die Folge einer Ausfallart hinsichtlich des Betriebs, der Funktion oder des Zustands eines Systems. Eine Ausfallauswirkung kann eine oder mehrere Ausfallarten einer oder mehrerer Einheiten zur Ursache haben. Diese Auswirkungen müssen ermittelt, beurteilt und aufgezeichnet werden. Sofern zutreffend, sollten Instandhaltungsaktivitäten und Zielvorgaben für das System ebenfalls beachtet werden. Eine Ausfallauswirkung kann ebenfalls die nächsthöhere Ebene beeinflussen und sich schließlich bis zur höchsten analysierten Ebene auswirken. Daher sollten in jeder Ebene die Auswirkungen von Ausfällen auf die nächsthöhere Ebene beurteilt werden.
Für jede Ausfallart muss festgelegt werden, wie sie erkannt wird. Die Notwendigkeit der getrennten Ausfallerkennung redundanter Einheiten während des Betriebs soll in Erwägung gezogen werden. Außerdem muss in jedem Fall gewährleistet sein, dass die Ausfallerkennung und ihre Anzeige gefährliche Betriebszustände ausschließen.

FMECA worksheets
Die Ergebnisse der Analyse sowie Detailinformationen des Systems werden in FMECA Arbeitsblättern (worksheets) dargestellt. Diese FMECA worksheets enthalten die wesentlichen Informationen für das gesamte System, sie zeigen die Art und Weise, wie das System möglicherweise ausfallen kann, die Bauteile und ihre Ausfallarten, die den Systemausfall verursachen können, und den Grund bzw. die Gründe für das Auftreten jeder einzelnen Ausfallart.
In den Arbeitsblättern werden die Einzelheiten der Analyse in tabellarischer Form festgehalten. Abbildung 17 und Abbildung 18 sind Beispiele für FMECA worksheets. Während die Tabelle in Abbildung 17 ein universelles Arbeitsblatt zeigt, ist die Tabelle in Abbildung 18 einem Herstellungsprozess für Aluminumstrangpressen entnommen [19].

In der Regel werden spezielle Arbeitsblätter mit besonderen Spalteneingängen entworfen. Die genaue Form, Anzahl und Beschriftung der Spalten richtet sich nach Anwendungs- und Projektanforderungen.

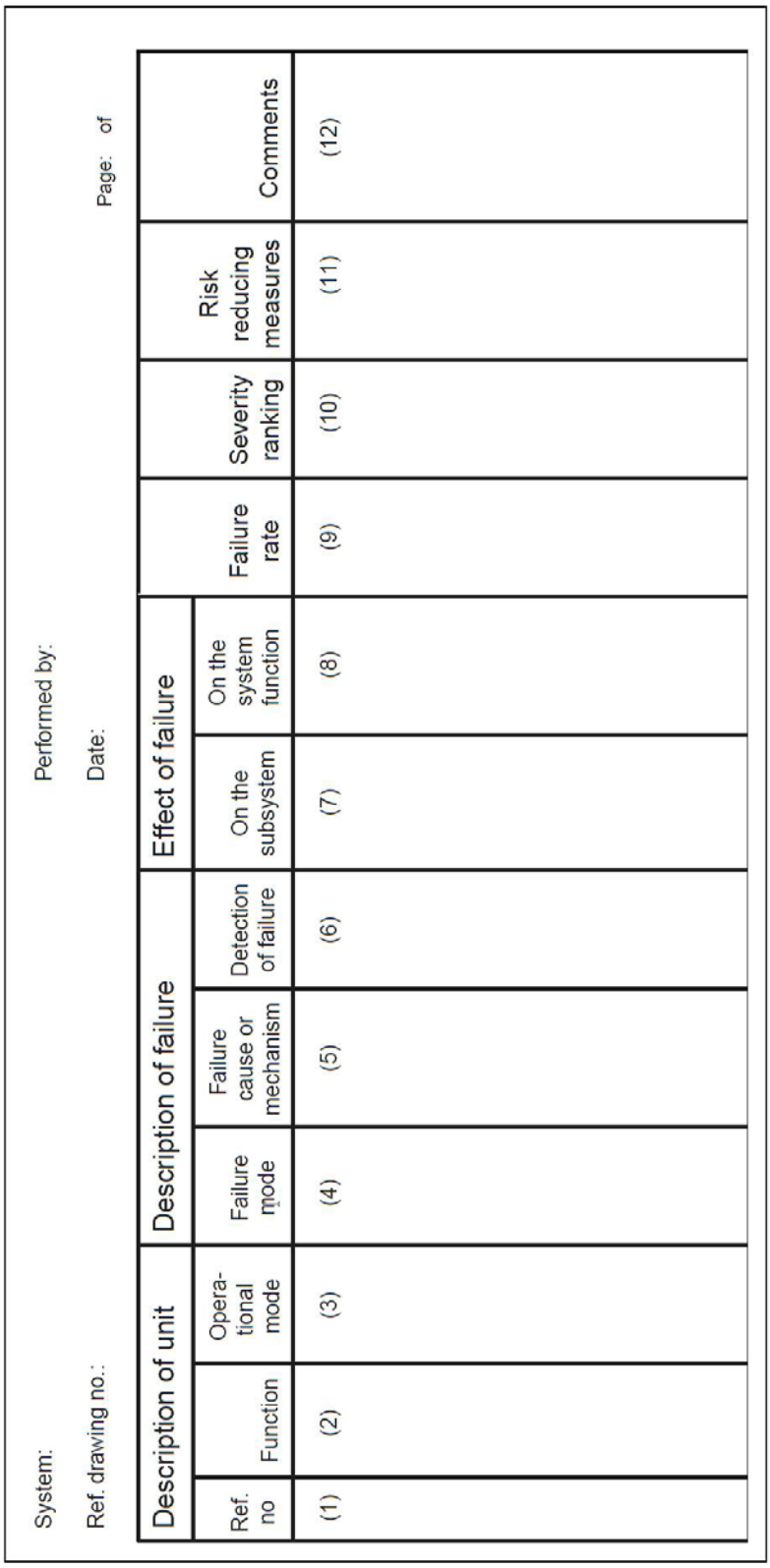
Der Kopfteil des Arbeitsblattes sollte jedenfalls folgendes enthalten:

- *das System als Ziel-Einheit bezeichnet die Einheit, für die die Endauswirkungen ermittelt werden sollen. Diese Bezeichnung sollte im Einklang stehen mit der in Blockdiagrammen, Schaltbildern und anderen Darstellungen verwendeten Terminologie*
- *die für die Analyse unterstellte Betriebsart*
- *Einheit bezieht sich auf die Einheit (Modul, Baugruppe oder Bauteil), die in diesem Arbeitsblatt untersucht wird*
- *Überarbeitungsebene, Datum und Name des Bearbeiters, sowie die Namen der Kernteammitarbeiter [19]*

Die Spalten des Arbeitsblatts selbst sollten u.a. folgende Informationen enthalten:

- Bezeichnung der Einheit und Funktionsbeschreibung
- Ausfallart
- Mögliche Ausfallursachen
- Auswirkung des Ausfalls
- Erkennungsmethode
- Vorkehrungen zur Ausfallvermeidung
- Schwereklasse und Eintrittshäufigkeit
- Bemerkungen

Abbildung 16: Beispiel für ein generisches FMECA worksheet



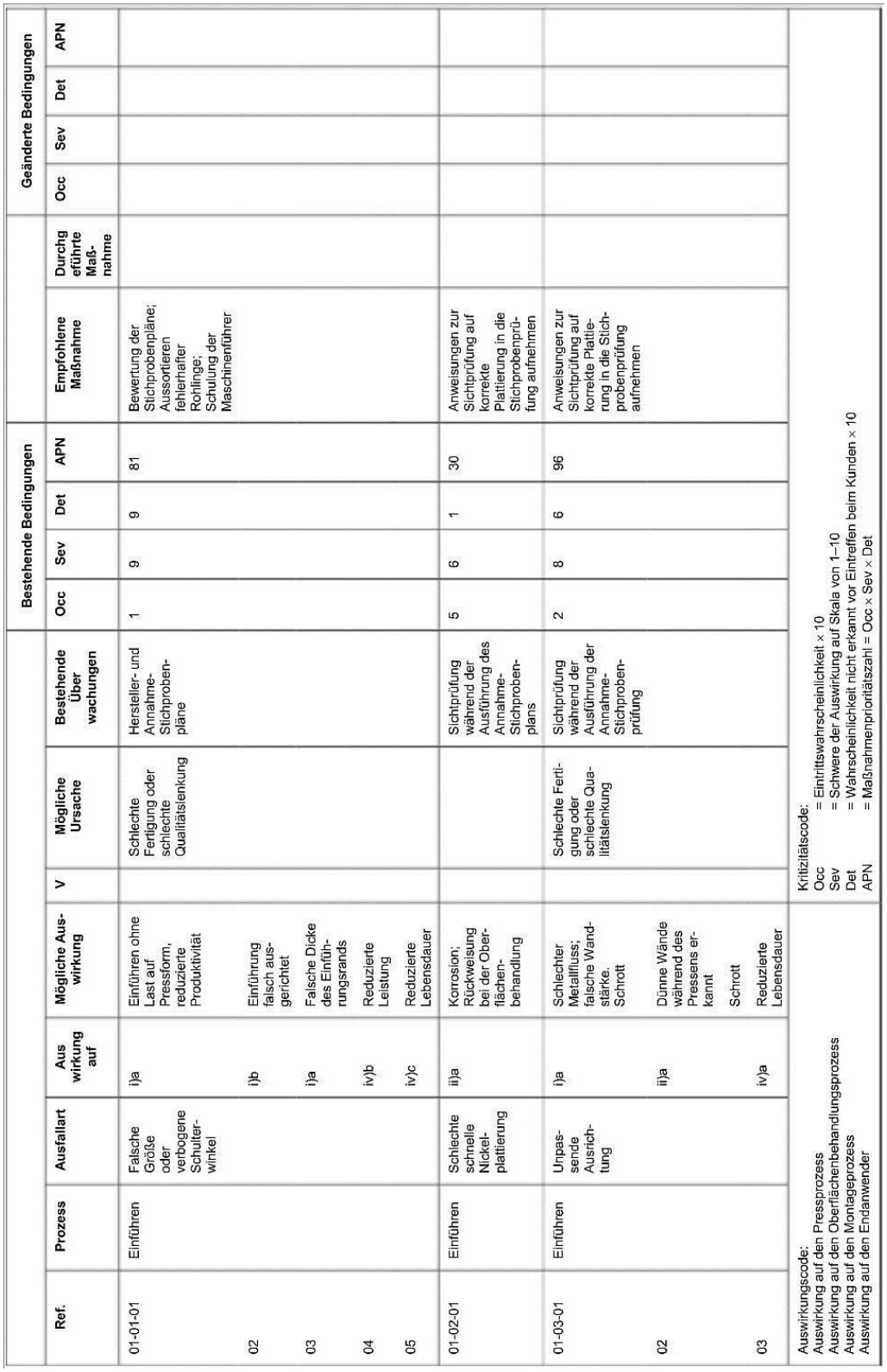
System: Performed by:

Ref. drawing no.: Date: Page: of

Description of unit			Description of failure			Effect of failure		Failure rate	Severity ranking	Risk reducing measures	Comments
Ref. no	Function	Operational mode	Failure mode	Failure cause or mechanism	Detection of failure	On the subsystem	On the system function				
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)

[Quelle: M. Rausand, A. Høyland: ***System Reliability Theory. Models, Statistical Methods, and Applications.*** Wiley & Sons, 2.Auflage, Hoboken, NJ, 2004, S. 89]

Abbildung 17: Beispiel FMECA worksheet Aluminiumstrangpressen



							Bestehende Bedingungen					Empfohlene Maßnahme	Durchgeführte Maßnahme	Geänderte Bedingungen			
Ref.	Prozess	Ausfallart	Auswirkung auf	Mögliche Auswirkung	V	Mögliche Ursache	Bestehende Überwachungen	Occ	Sev	Det	APN			Occ	Sev	Det	APN
01-01-01	Einführen	Falsche Größe oder verbogene Schulterwinkel	i)a	Einführen ohne Last auf Pressform, reduzierte Produktivität		Schlechte Fertigung oder schlechte Qualitätslenkung	Hersteller- und Annahme-Stichprobenpläne	1	9	9	81	Bewertung der Stichprobenpläne; Aussortieren fehlerhafter Rohlinge; Schulung der Maschinenführer					
02			i)b	Einführung falsch ausgerichtet													
03			i)a	Falsche Dicke des Einführungsrands													
04			iv)b	Reduzierte Leistung													
05			iv)c	Reduzierte Lebensdauer													
01-02-01	Einführen	Schlechte schnelle Nickelplattierung	ii)a	Korrosion; Rückweisung bei der Oberflächenbehandlung			Sichtprüfung während der Ausführung des Annahme-Stichprobenplans	5	6	1	30	Anweisungen zur Sichtprüfung auf korrekte Plattierung in die Stichprobenprüfung aufnehmen					
01-03-01	Einführen	Unpassende Ausrichtung	i)a	Schlechter Metallfluss; falsche Wandstärke. Schrott		Schlechte Fertigung oder schlechte Qualitätslenkung	Sichtprüfung während der Ausführung der Annahme-Stichprobenprüfung	2	8	6	96	Anweisungen zur Sichtprüfung auf korrekte Plattierung in die Stichprobenprüfung aufnehmen					
02			ii)a	Dünne Wände während des Pressens erkannt Schrott													
03			iv)a	Reduzierte Lebensdauer													

Auswirkungscode:
Auswirkung auf den Pressprozess
Auswirkung auf den Oberflächenbehandlungsprozess
Auswirkung auf den Montageprozess
Auswirkung auf den Endanwender

Kritizitätscode:
Occ = Eintrittswahrscheinlichkeit × 10
Sev = Schwere der Auswirkung auf Skala von 1–10
Det = Wahrscheinlichkeit nicht erkannt vor Eintreffen beim Kunden × 10
APN = Maßnahmenprioritätszahl = Occ × Sev × Det

[**Quelle: European Committee for Electrotechnical Standardization (CENELEC):** ***EN 60812 Analysetechniken für die Funktionsfähigkeit von Systemen. Verfahren für die Fehlzustandsart- und -auswirkungsanalyse (FMEA).*** **2006, S. 44**]

Risikoprioritätszahl (Risk Priority Number RPN)

Die Risikoprioritätszahl (engl. Risk Priority Number oder RPN) ist eine ganze Zahl zwischen 1 und 1000 und dient der Risikobewertung. Die RPN ist das Produkt der Einflußfaktoren Severity S, Occurrence O und Detection D. Diese drei haben einen Wert zwischen 1 und 10 [64].

Tabelle 7: Allgemeine Richtwerte der Einflussfaktoren

Bewertung	Severity S (Schwere)	Occurrence O (Eintrittswahrscheinlichkeit)	Detection D (Entdeckungswahrscheinlichkeit)	
1	Kunde bemerkt Auswirkung nicht	Fehler unwahrscheinlich; ähnliche Konstruktionen bisher ohne Fehler	Nahezu sichere Entdeckung	99,99%
2	Auswirkung sehr gering, Kunde kaum berührt	Fehler selten	Sehr hohe Wahrscheinlichkeit der Entdeckung	99,9%
3	Gebrauchsnutzen geringfügig eingeschränkt, Kunde leicht verdrossen	Sehr wenige Fehler wahrscheinlich	Hohe Wahrscheinlichkeit der Entdeckung	
4	Gebrauchsnutzen wenig eingeschränkt, Kunde verdrossen	Wenige Fehler wahrscheinlich	Mäßig hohe Wahrscheinlichkeit der Entdeckung	99,7%
5	Mäßiger Einschränkung des Gebrauchs-nutzens, Kunde etwas verärgert	Gelegentliche Fehler wahrscheinlich	Mittlere Wahrscheinlichkeit der Entdeckung	
6	Ausfall einzelner Hauptfunktionen, Kunde ziemlich verärgert	Mittlere Zahl von Fehlern wahrscheinlich	Nahezu mittlere Wahrscheinlichkeit der Entdeckung	
7	Funktionen stark eingeschränkt, Kunde verärgert	Mäßig große Zahl von Fehler wahrscheinlich	Geringe Wahrscheinlichkeit einer Entdeckung	
8	Totaler Funktionsausfall, Kunde sehr verärgert	Sehr große Zahl von Fehlern wahrscheinlich	Sehr geringe Wahrscheinlichkeit der Entdeckung	98%
9	Gefährdung, Verstoß gegen Gesetze möglich	Sehr große Anzahl von Fehlern wahrscheinlich	Entdeckung möglich, aber unsicher	90%
10	Gefährdung, Verstoß gegen Gesetze	Fehler nahezu sicher; zahlreiche Fehler mit gleichen oder ähnlichen Konstruktionen bekannt	Keine Entdeckungsmaßnahmen geplant	<90%

[**Quelle: eigene Darstellung in Anlehnung an P. Liggesmeyer: *Safety and Reliability of Embedded Systems. FMECA*. http://agde.informatik.uni-kl.de/teaching/suze/ws2008/material/folien/SRES_05_FMECA.pdf, 2008, Zugriff: 26.06.2013**]

Severity S:

Bedeutung oder Schwere; Schätzwert wie stark die Auswirkungen eines Ausfalls das System oder den Anwender beeinflussen können.

Occurrence O:

die Eintrittswahrscheinlichkeit für eine Ausfallart für einen angenommenen oder festgelegten Zeitraum

Detection D:

Schätzwert für die Chance, den Ausfall zu erkennen und zu beheben, bevor das System oder der Kunde betroffen wird.

Risikomatrix/Kritizitätsmatrix

Die Risikomatrix oder Kritizitätsmatrix ist eine weitere Form der Risikobewertung. Die Matrix kann aus den zugeordneten Werten für die Schwere und den Ereignishäufigkeiten erstellt werden. Risikoakzeptanz wird subjektiv definiert oder durch fachliche und finanzielle Entscheidungen gesteuert und unterscheidet sich je nach Branche [19].

Tabelle 8: Risikomatrix/Kritizitätsmatrix

Eintrittshäufigkeit der Auswirkung	Schwereniveau			
	1 unbedeutend	2 geringfügig	3 kritisch	4 katastrophal
5 häufig	unerwünscht	nicht akzeptabel	nicht akzeptabel	nicht akzeptabel
4 wahrscheinlich	akzeptabel	unerwünscht	nicht akzeptabel	nicht akzeptabel
3 gelegentlich	akzeptabel	unerwünscht	unerwünscht	nicht akzeptabel
2 gering	vernachlässigbar	akzeptabel	unerwünscht	unerwünscht
1 unwahrscheinlich	vernachlässigbar	vernachlässigbar	akzeptabel	akzeptabel

[Quelle: eigene Darstellung in Anlehnung an European Committee for Electrotechnical Standardization (CENELEC): *EN 60812 Analysetechniken für die Funktionsfähigkeit von Systemen. Verfahren für die Fehlzustandsart- und -auswirkungsanalyse (FMEA)*. 2006, S. 22]

Team Review

Die FMECA ist ein kontinuierlicher Prozess zwischen verschiedenen Personen aus unterschiedlichen Fachbereichen. Für die Durchführung einer FMECA ist in erster Linie der verantwortliche Projektleiter oder Entwicklungsleiter zuständig. Abhängig von Produkt und Branche ist auch die Mitarbeit von folgenden Personengruppen notwendig:

- Hardware/Software/System Designer
- Tester
- Zuverlässigkeitstechniker/Sicherheitstechniker
- Qualitätsprüfer
- Wartungstechniker
- Kundendiensttechniker/Außendiensttechniker
- Fertigungstechniker, usw.

Ein Team aus ausgewählten Personen aus allen betroffenen Fachbereichen erstellen anhand der FMECA worksheets, RPN und/oder der Risikomatrix/Kritizitätsmatrix Team Reviews. In diesen wird entschieden, ob das untersuchte System noch akzeptabel ist oder nicht. Es werden auch konkrete Vorschläge zur Verbesserungen oder Reduzierung des Risikos vorgeschlagen oder erarbeitet. Diese Vorschläge enthalten unter andern Maßnahmen zur Reduzierung der Wahrscheinlichkeit des Auftretens des Ausfalls, die Reduzierung der Auswirkung des Ausfalls oder auch die Steigerung der Wahrscheinlichkeit der Entdeckung des Ausfalls. Risiko kann ebenfalls durch Design Änderungen, Sicherheitseinrichtungen, Sicherheitsvorkehrungen, Warnanlagen oder mittels Anweisungen und Schulungen reduziert werden.

Tabelle 9: Korrekturmaßnahmen im RPN Vergleich

	Severity S	Occurrence O	Detection D	RPN
ursprünglich	8	7	5	280
verbessert	8	5	4	160
Reduzierung der RPN in %				43%

[Quelle: eigene Darstellung in Anlehnung an M. Rausand, A. Høyland: *System Reliability Theory. Models, Statistical Methods, and Applications.* http://frigg.ivt.ntnu.no/ross/slides/fmeca.pdf, 2004, Zugriff: 26.06.2013]

Ein Vergleich der RPN vor und nach den Korrekturmaßnahmen zeigt die Effektivität der Verbesserung (siehe Tabelle 9) [64].

Kritische Betrachtungen der RPN

Die Risikoprioritätszahl RPN basiert auf Schätzwerten und kann weder 100% objektiv noch absolut sein. Eine Vergleichbarkeit der Risikoprioritätszahl alleine ist außerdem nicht geben, verschiedene Faktoren führen nicht zwangsläufig zu verschiedenen Risikoprioritätszahl. Ebenso ist ein Vergleich der Risikoprioritätszahl bei verschiedenen Systemen nicht möglich. Obwohl die RPN eine ganze Zahl zwischen 1 und 1000 ist, können die drei Faktoren nur 120 Zahlen abbilden, d.h. 88% des Wertebereichs ist leer bzw. wird nicht genützt. Eine kleine Änderung in einem Faktor hat eine viel größere Auswirkung, wenn die anderen Faktoren größer sind, als wenn sie klein sind. Folgendes Beispiel erläutert diesen Umstand:

	9 x 9 x **3** = 243	3 x 4 x **3** = 36
	9 x 9 x **4** = 324	3 x 4 x **4** = 48
RPN Differenz:	**81**	**12**

Grenzen der FMECA

Die Durchführung einer FMECA bei komplexen Systemen kann sich als mühsam, zeitaufwendig und kostenintensiv erweisen. Die Ursache liegt an der Menge der zu berücksichtigenden detaillierten Informationen des Systems. Diese Schwierigkeiten können noch zusätzlich verstärkt werden, wenn das System mehrere mögliche Betriebsmodi kennt und sowohl die Reparatur – als auch Instandhaltungsvorschriften berücksichtig werden müssen. Alle Fehler (Fehlfunktionen) werden in der FMECA als voneinander unabhängig angesehen. Diese Betrachtung spiegelt die Realität aber nur bedingt wieder. FMECA ist deshalb auch wenig geeignet, um abhängige Fehler (*common cause failure*) zu behandeln. Menschliches Versagen wird gerne als Ursache vernachlässigt.

Eine weitere Schwäche der FMEA [und FMECA[14]] liegt in ihrer Unfähigkeit, ein Maß für die Zuverlässigkeit des Gesamtsystems zu liefern, und aus demselben Grund ist sie nicht in der Lage, Maßnahmen für Entwurfsverbesserungen und Kompromissstudien zu liefern [19].[15]

[14] Zum Zeitpunkt der Veröffentlichung der EN 6812 wurde nicht immer genau zwischen FMEA und FMECA unterschieden, hier ist aber speziell FMECA gemeint.

[15] EN 60812: a.a.O., Abschnitt 7.3, S. 31

7. Sicheres Mikrocomputersystem (SIMIS)

Sicherheitssysteme können einkanalig, zweikanalig oder mehrkanalig realisiert werden. Da einkanalige Systeme in der Regel auf einen Fehler mit einem Ausfall reagieren, muss sichergestellt sein, dass diese ein Fail-Safe-Verhalten aufweisen.

Sicherer Zustand und Fail-Safe-Verhalten
Ein sicherer Zustand ist nach EN 50129 definiert als „ein Zustand, der die Sicherheit weiterhin bewahrt." Sichere Zustände sind unverzichtbarer Bestandteil des Fail-Safe-Prinzips und jeder Sicherheitsarbeit.
Bei fehler- und ausfallfreier Funktion muss sich ein sicherheitsrelevantes System grundsätzlich in einem sicheren Zustand – im sicheren Ausgangszustand – befinden. Dem Fail-Safe-Verhalten entsprechend muss das System bei Eintreten eines Ausfalls einen sicheren Zustand – einen vom sicheren Ausgangszustand abweichenden Zustand – einnehmen; dabei hängt es von der Gestaltung des sicherungstechnischen Systems ab, ob dieser sichere Zustand die Funktionalität des Systems einschränkt oder nicht.[16][18][22]

Einkanalige Systeme als Hardwarelösung entsprechen nicht mehr dem Stand der Technik, stattdessen werden Mehrkanalige Systeme verwendet. Diese erhöhen nicht nur die Zuverlässigkeit, sondern auch die Verfügbarkeit.

SIMIS steht für *Sicheres Mikrocomputersystem* und wurde in den 80er Jahren von der Firma Siemens und der Firma SEL entwickelt. Mit SIMIS gesteuerten elektronischen Stellwerke wurden nach und nach die in die Jahre gekommenen Relaisstellwerke abgelöst [73].

SIMIS Konzept
Das SIMIS-Konzept besteht aus zwei identisch aufgebauten Mikrorechnersystemen, die taktsynchron betrieben werden und identisch programmiert sind.

[16] W. Fenner, P. Naumann, J. Trinckauf; Bahnsicherungstechnik: Steuern, Sichern und überwachen von Fahrwegen und Fahrgeschwindigkeiten im Schienenverkehr, Publicis Publishing, 2.Auflage, Erlangen, 2004, S. 147

Abbildung 14 zeigt das Blockschaltbild eines zweikanaligen Mikrorechnersystems mit jeweils unabhängig voneinander arbeitenden CPU, ROM-, RAM-Bausteinen sowie Ein- und Ausgabeeinheiten.
Der interne Datenaustausch erfolgt in Form von normierten Telegrammen, die zusätzlich einen Sicherheitsanhang beinhalten. Dieser Sicherheitsanhang ermöglicht eine Hamming-Distanz[17] von d=5, wodurch bis zu vier gleichzeitig auftretende Bitfehler pro Telegramm sicher erkannt werden.

Beide Rechner erhalten die gleichen Eingabedaten und werden mit einem sicheren Taktgeber (SITAG) synchronisiert. Ein vom Taktgeber ausgehendes Überwachungssignal kann nur bei Antivalenz der Bussignale diese Kette durchlaufen und wieder zum Taktgeber zurückkehren [31].

Die für den Vergleich verwendeten Signale werden invertiert, um die notwendige Antivalenz zur Erkennung von einander kompensierenden Doppelfehler zu beherrschen.

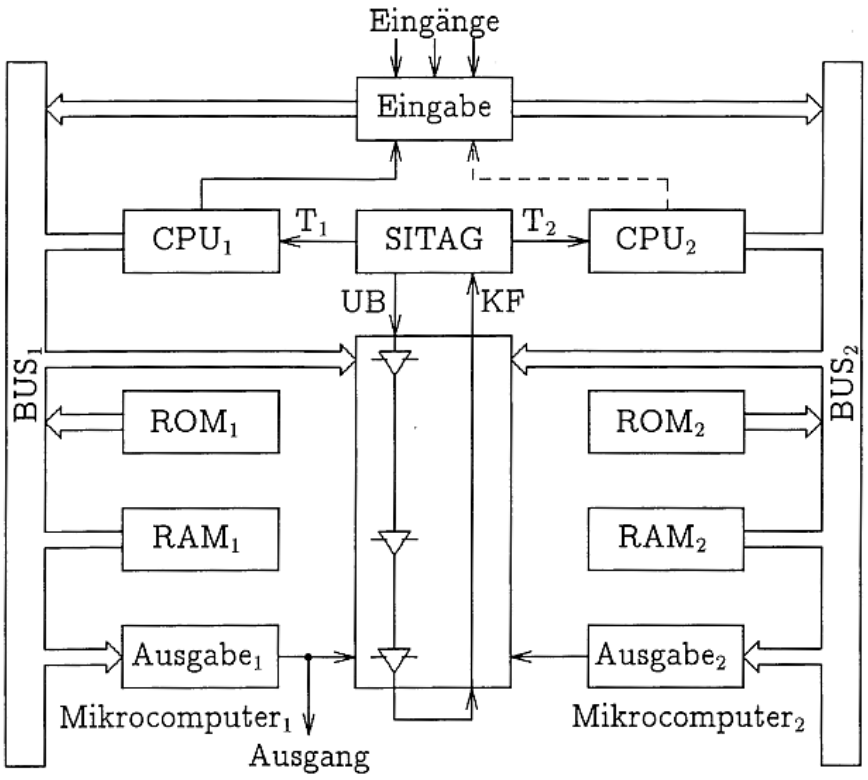
Während des Datenverarbeitungsprozesses vergleicht ein Antivalenz-Überprüfungs-Baugruppe mit Fail-Safe-Verhalten bei jedem Verarbeitungsschritt die Daten- und Adresssignale der Busse beider Systeme miteinander.

Werden Unterschiede in den Daten- und Adresssignalen der Busse festgestellt, wird der Taktgeber unterbrochen, sodass beide Zentraleinheiten stehen bleiben und kein fehlerhaftes Ergebnis weitergeleitet wird. Ein Funktionswächter (genaue Beschreibung siehe Abschnitt 7) aktiviert die Abschalteinrichtung, die das das System in einen sicheren Abschalt- bzw. Betriebszustand führt.

Zusätzlich läuft als Hintergrundprozess das SIMIS-Online Prüfprogramm (SOPP). Die Überwachung der Laufzeit des SOPP stellt sicher, dass es mindestens einmal innerhalb der maximal zulässigen Fehleroffenbarungszeit durch alle zu prüfenden Baugruppen gelaufen ist.

[17] Hamming-Distanz oder auch Hamming-Abstand ist ein Maß für die Unterschiedlichkeit von binären Zeichenketten.

Abbildung 18: Blockschaltbild Beispiel eines SIMIS



[Quelle: G.-H. Schildt, W. Kastner: *Prozeßautomatisierung.* Springer, Wien, 1998, S. 69]

Der Übergang in diesem sicheren Zustand ist irreversibel und kann von beiden Mikrocomputern nicht mehr rückgängig gemacht werden. Erst durch entsprechendes Wartungspersonal kann das System wieder neugestartet werden. Um eine höhere Verfügbarkeit zu erreichen, kann ein dritter Rechner entweder als Hot-Spare oder als Slave verwendet werden. Man spricht dann von einer 2-von-3-Redundanz. Ein Slave Rechner muss erst gestartet werden, es entsteht deshalb eine kurze Unterbrechung, wogegen ein Hot-Spare-Rechner sofort den ausgefallen Rechner ersetzt.

Common Mode Failure sind Fehler, die eine gemeinsame Ursache haben [9]. Um *Common Mode Failures* zu vermeiden, wird eine größtmögliche Unabhängigkeit des SIMIS, der Vergleichs- und Abschalteinrichtung angestrebt.

Folgende Maßnahmen sind vorgesehen:

Getrennte Stromversorgung
Werden verschiedene Systembestandteile aus einer gemeinsamen Stromquelle gespeist, kann eine Störung dieser auch die übrigen Komponenten beeinflussen. In SIMIS werden deshalb zwei getrennte Spannungsregler verwendet, die auf fehlerfreie Ausgangsspannung überwacht werden.

Galvanische Trennung
Eine direkte Verbindungen zwischen zwei (oder mehreren Geräten) kann zu gleichen, aber nicht erkennbar falschen Ergebnissen in beiden (oder mehreren) Kanälen führen.

Entkoppelung
Elektromagnetische Strahlung kann zu Störsignale führen, die von einer Baugruppe in eine andere gekoppelt werden kann. Es kann zu einer gemeinsamen Fehlfunktion kommen. Abhilfe schafft eine räumliche Trennung und andere Abschirmmaßnahmen.

8. Funktionswächter (Watchdog)

Ein Funktionswächter oder Watchdog ist eine robuste und unabhängige Hardware- und Softwarelösung, die im Allgemeinen dazu dient, ein System zu überwachen. Wird ein Fehlzustand, ein Ausfall oder Fehlfunktion durch den Funktionswächter entdeckt, werden Korrekturmaßnahmen eingeleitet. Oberste Priorität ist es, das System in einen sicheren Zustand zu bringen, danach kann versucht werden, einen normalen Betriebszustand wiederherzustellen. Der Funktionswächter ist ebenso ein wichtiger Schutz gegen böswilligen Code, Designfehler oder unvorhersehbare Ereignisse.

Watchdog Anwendungsbeispiel Clementine-Sonde

Clementine war eine US-Sonde, die am 25. Jänner 1994 von der Vandenberg Air Force Base in Kalifornien gestartet wurde [42][57]. Die Sonde war mit einer neuen Generation von Kameras, Solarzellen und Instrumenten ausgestattet [51]. Die Kosten für die Entwicklung, Konstruktion, Start und Durchführung der Mission wurde mit 80 Millionen US-Dollar beziffert [1]. Als Missionsziele wurde der Test von leichtgewichtigen Verteidigungstechnologien, Beobachtung von Langzeiteffekten der Weltraumbedingungen auf neue Technologien und die Testzielerfassung und Verfolgung von astronomischen Objekten, welche ähnliche Flugeigenschaften wie strategische Raketen haben, angegeben [50]. Als Sekundärziele wurde die Kartierung des Mondes in 11 Spektralkanälen und die Untersuchung des Asteroiden (1620) Geographos[18] definiert [50].
Ein Honeywell Dualprozessorsystem MIL-STD 1750A Subsystem kümmerte sich um die Telemetrie und andere verschiedene Funktionen der Raumsonde. Die Bedienung der Schubdüsen waren unter Kontrolle der Boden Station. Obwohl der MIL-STD 1750A auch die Kontrolle der Schubdüsen übernehmen konnte, war dies nur für den Notfall vorgesehen [27].
Die Clementine-Sonde erreichte planmäßig den Mond am 19. Februar und verließ den Mondorbit nach erfolgreicher Kartierung wieder am 22. April 1994 in Richtung Asteroid (1620) Geographos [50].

[18] (1620) Geographos ist ein erdnaher Asteroid, der 1951 entdeckt wurde und eine Abmessungen 5 × 2 × 1,5 km besitzt.

Am 7. Mai 1994 trat beim Honeywell MIL-STD 1750A ein Fließkomma Fehler auf. Davor waren schon 3000 solcher Fehler erkannt und richtig behandelt worden. Dieses Mal verursachte der Fehler aber einen Software-Crash, der den Prozessor blockierte. Die Bodenkontrolle versuchte 20 Minuten lang vergeblich, die Kontrolle des Systems mit Hilfe eines Software Resets wieder zu erlangen. Erst nach dem Senden eines Hardware Reset Befehls war der Prozessor wieder erreichbar. Während dieser unkontrollierten 20 Minuten zündete die Software mehrere Schubdüsen und verschwendete damit wertvollen Treibstoff. Die Sonde drehte sich daraufhin mit einer Geschwindigkeit von 80 Umdrehungen pro Minute um die eigene Achse [27][51].
Als Clementine wieder unter Bodenkontrolle gebracht wurde, musste die Mission dennoch aufgeben werden, da praktisch kein Treibstoff mehr vorhanden war. Obwohl der Honeywell MIL-STD 1750A Prozessor einen eingebauten internen Watchdog besaß, entschied das Software Team, ihn nicht zu benutzen [27]. Ein externer Watchdog oder ein einige Zeilen Programm Code für den internen Watchdog hätte mit hoher Wahrscheinlichkeit die 80 Million Dollar Mission verlängert.

Watchdog Anwendungsbeispiel Mars Pathfinder

Am 4. Dezember 1996 startete die NASA die Mars-Sonde Pathfinder von Cape Canaveral Air Force Station in Florida [49][81]. Die Sonde Pathfinder bestand aus drei Teilen, dem Cruise Stage, dem Lander und dem Rover Sojourner. Die Gesamtkosten für die Mission bezifferte die NASA mit 265 Millionen Dollar [50]. Wie geplant koppelte sich die Cruise Stage von Lander und Rover kurz vor dem Landungsmanöver ab. Am 4.Juli 1997 erreichten schließlich Lander samt Rover sicher die Marsoberfläche [49][50].
Der Lander verfügte über einen Bordcomputer mit einem RAD6000SC Prozessor und VMEbus. Dieser Prozessor war eine strahlungsgehärteten Singlechip Variante des multiprozessortauglichen RAD6000, hergestellt von Lockheed Martin[19]. Als Echtzeit Betriebssystem kam VxWorks der Firma Wind River Systems zum Einsatz, auf der die bestehende Flugsoftware, ca. 150.000 Zeilen Programm Code, lief [49][50].

[19] Heute BAE Systems

Das Betriebssystem VxWorks bot *Preemptive Priority Scheduling*[20] für Threads an [38]. Auf der Sonde wurden die Tasks somit als Threads mit Prioritäten abgearbeitet, wobei die Prioritäten gleichzeitig die relative Dringlichkeit dieser Tasks widerspiegelten[58][62]. Der Zugriff auf den VMEbus erledigte ein Bus Management Task, der mit hoher Priorität die Daten in und aus dem Bus bewegt. Der Buszugriff selbst wurde mit *Mutual Exclusion Locks*[21] synchronisiert [38].
Aufgrund eines Software-Fehlers trat jedoch manchmal eine *Prioritätsinversion*[22] beim Zugriff auf den VMEbus auf, sodass wichtige Tasks wie die Kommunikation nicht mehr ausgeführt werden konnte [65]. Glücklicherweise erkannte nach einer Zeit der Watchdog, dass einige wichtige Tasks nicht mehr ausgeführt wurden und folgerte richtigerweise, dass etwas schief gegangen sein muss und löste ein Hardware Reset aus. Durch den Systemneustart konnte die Kommunikation und Kontrolle der Mission Control wiederhergestellt werden. Die Techniker der Mission Control konnten schließlich das Problem nachvollziehen und durch einen Programm Upload beseitigen [2][38][65]. Ohne den Watchdog wäre diese Mission frühzeitig gescheitert.

Grundlegender Aufbau eines Watchdogs

Der Funktionswächter besteht aus einem digitalen Zähler, der mit einem bestimmten Tempo, abhängig von einem festen Taktgeber, von einem Anfangswert zu einem Zielwert zählt. Typischerweise wird von einem Initialisierungswert runter bis auf Null gezählt. Dieser Anfangswert ist programmierbar und die Zeitspanne, die der Counter benötigt, um von diesem Initialisierungsswert auf Null zu zählen, ist bekannt. Wenn der Zähler bei Null oder dem Zielwert angelangt ist, generiert er ein Timeout Signal. Das Timeout-Signal wiederum ist zu einem externen Schaltkreis verbunden.

[20] Beim Preemptive Priority Scheduling kann einem Task mit niedriger Priorität Ressourcen entzogen werden, um diese an einem anderen Task mit höherer Priorität zu übergeben.

[21] Wechselseitiger Ausschluss einer gemeinsam genutzten Ressource mit Hilfe einer Sperre

[22] Durch die Prioritätsinversion kann ein höher priorisierter Task beliebig lange durch niedriger priorisierte blockiert werden.

Der Watchdogs Timer (WDT) wird beim Systemboot gestartet. Ein Timeout des WDT erzeugt normalerweise ein Hardware Reset. Um dies zu verhindern, muss das System in regelmäßigen Abständen den Zählerstand des Watchdogs wieder auf den Anfangswert zurücksetzen. Der Prozess des Zählerneustarts wird im Englischen auch ‚kicking the dog' genannt [5].

Die entsprechende Metapher ist ein Mann, der von einem bösartigen Hund angegriffen wird. Solange er den Hund tritt (‚kicking the dog'), kann dieser ihn nicht beißen. Er darf aber niemals aufhören, da er sonst irgendwann gebissen wird. In ähnlicher Weise muss das System den Watchdog zyklisch zurücksetzen, um ein Reset zu vermeiden [5].

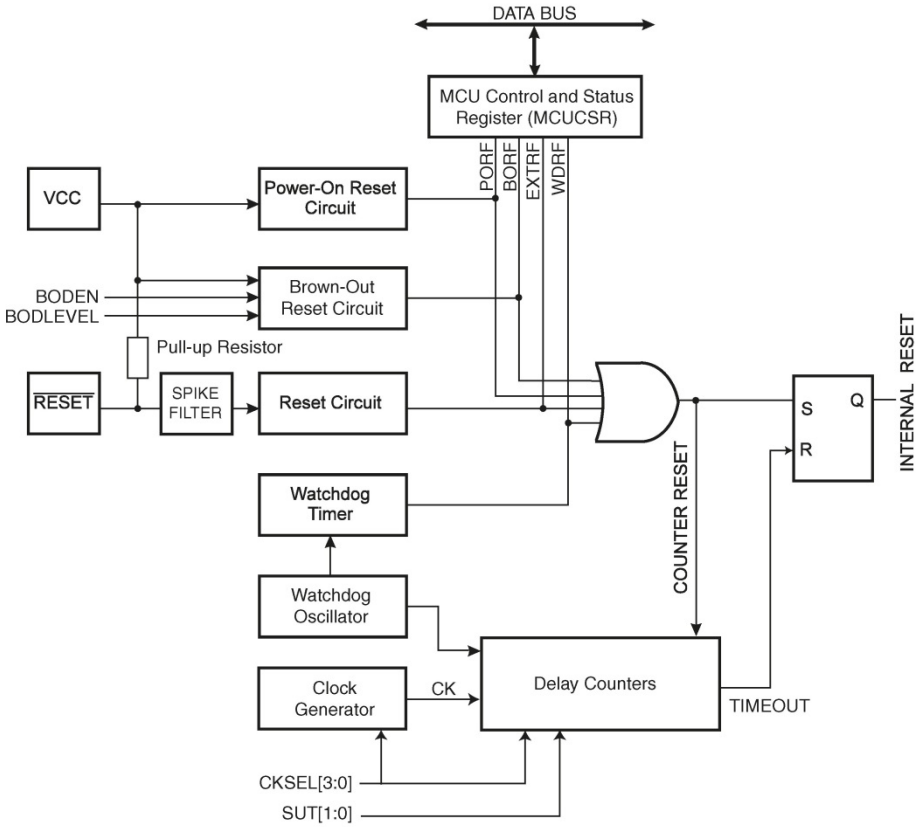
Watchdog Timer sind im Embedded Bereich weit verbreitet, sie sind ein wichtiger Bestandteil von Microcontrollern. Sie stellen einen automatischen Mechanismus zur Verfügung, um sowohl Softwarefehler als auch transiente Hardwarefehler zu behandeln [47]. Ein Funktionswächter kann viel schneller bei Fehlern eingreifen als menschliches Bedienpersonal und ist von unschätzbarem Wert, wenn menschliche Benutzer nicht schnell genug auf Fehlzuständige reagieren können. In manchen Fällen (z.B. unbemannte Raumfahrt) ist ein menschliches Eingreifen weder möglich noch vorgesehen. Auch hier leisten Watchdogs wertvolle Dienste.

Der Optimale Watchdog

Ein effektiver Watchdog ist unabhängig vom Hauptsystem und darf keine Annahmen bezüglich des Zustands der Software oder Hardware machen. Er muss das System wieder in einen normalen Betriebszustand bringen können, unabhängig von Fehlern wie Programmabsturz, Bit Flip, Glitch, Elektromagnetische Interferenz oder kosmischer Strahlung. Bei einem schwerwiegenden Hardwareausfall muss der Watchdog stets versuchen, das System in einen Fail-Safe-Zustand zu bringen [26].
Die Umstände, die zu einem Watchdog Ereignis geführt haben, soll idealerweise immer protokolliert und für eine spätere Analyse aufgezeichnet werden. Unter keinen Umständen darf die Betriebssoftware erforderliche Komponenten des WDT wie dessen Reset Vektor oder Input/Output Pins neuprogrammieren

dürfen. Damit wird verhindert, dass fehlerhafter oder bösartiger Programm Code den Schutzmechanismus aushebeln [28].

Abbildung 19: ATmega8 Rest Logik



[Quelle: Atmel Corporation: ATmega8(L) datasheet. http://www.atmel.com/devices/atmega8.aspx?tab=documents, 2013, S. 38]

Die Abbildung 19 zeigt die Reset Logik des Atmel ATmega8 Mikrocontrollers. Der hier abgebildete Watchdog Timer besitzt einen eigenen Oszillator und ist mit dem internen Reset verbunden.

Ein Watchdog darf nicht für die normale Fehlerbehandlung missbraucht werden. Erst wenn ein Fehler nicht mehr durch normales Exception Handling beherrschbar ist bzw. auch diese versagt, soll der Watchdog Gegenmaßnahmen einleiten.

Der Funktionswächter muss das System als Ganzes überwachen können. Eine vernünftige Auswahl von verschiedenen Parametern und Funktionen soll beobachtet werden. Erst wenn alle Parameter und Funktionen auf ein fehlerfreies System verweisen, soll der Zähler des WDT wieder zurückgesetzt werden.

Es ist schwierig bis unmöglich, den perfekten Watchdog Timer zu konstruieren. In der realen Welt müssen Abstriche in Bezug auf Unabhängigkeit, Kosten, Gewicht, Platz, Temperaturempfindlichkeit, etc. gemacht werden. Ein unabhängiger Oszillator ist vergleichsweise einfach zu realisieren, eine unabhängige Stromversorgung ist dagegen schon eine größere Herausforderung [27]. Eine möglichst effektive Realisierung mit robustem Design wird aber angestrebt.

Interner Watchdog Timer
Die meisten Embedded Prozessoren, die eine hohes Maß an Integration besitzen, haben auch interne WDT. Ein hoher Kostendruck führt oft dazu, deren Verwendung in Betracht zu ziehen. Diese bieten jedoch nur minimalen Schutz, da die geforderte Unabhängigkeit nicht gegeben ist. Internet WDT Register sind hier nicht ausreichend gegenüber ungewolltem Zugriff geschützt. Ebenso ist eine gemeinsame Verwendung des Taktgebers sowohl für das Embedded System als auch für den zugehörigen internen Watchdog Timer bei einem sicherheitskritischen System nicht ratsam [26].

Externer Watchdog Timer
Externe WDT sind idealerweise weder vom Systemhardware noch der darauf laufenden Software abhängig, die sie überwachen soll. Sie verwenden keine gemeinsamen Ressourcen, haben einen unabhängigen Oszillator, integrierten Speicher und eigene I/O Pins [26].

Windowed Watchdog Timer
Windowed Watchdog Timer erlauben das Zurücksetzen des Zählers nur zu bestimmten Zeitperioden. Ein Beispiel für eine solche Zeitperiode oder Zeitfenster wären etwa die ersten 10 Takte einer Sekunde.

Wird der Counter eines Windowed Watchdogs aufgrund eines bösartigen Programms oder eines fehlerhaften Codes zu oft gekickt, wird ebenfalls ein Timeout ausgelöst. Um die Zuverlässigkeit und Sicherheit weiter zu erhöhen, kann noch zusätzlich verlangt werden, dass das Zurücksetzen nur mit zwei verschiedene I/O-Befehlen in bestimmter Reihenfolge erfolgen darf. Mit diesen einfachen Maßnahmen wird ein zufälliges und ungewolltes Zurücksetzen effektiv erschwert [47][52].

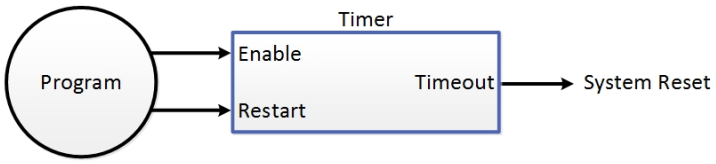
Watchdog Architekturen

Es werden nacheinander drei Watchdog Architekturen vorgestellt, die stufenweise an Leistungsfähigkeit und Komplexität gewinnen. Der Simple Watchdog ist die minimalste Architektur eines Funktionswächters [47].

Simple Watchdog

Die einfachste Form einer Watchdog Schaltung besteht aus einem einfachen Zähler, der bei einem Timeout sofort ein System Neustart aktiviert. Das Timeout-Signal wird dabei entweder direkt oder über eine Signalaufbereitungsschaltung zum System Reset Input verbunden. Diese Architektur ist darauf angewiesen, dass bei einem System Reset auch die Control Outputs in einen sicheren Zustand gebracht werden [47].

Abbildung 20: Simple Watchdog

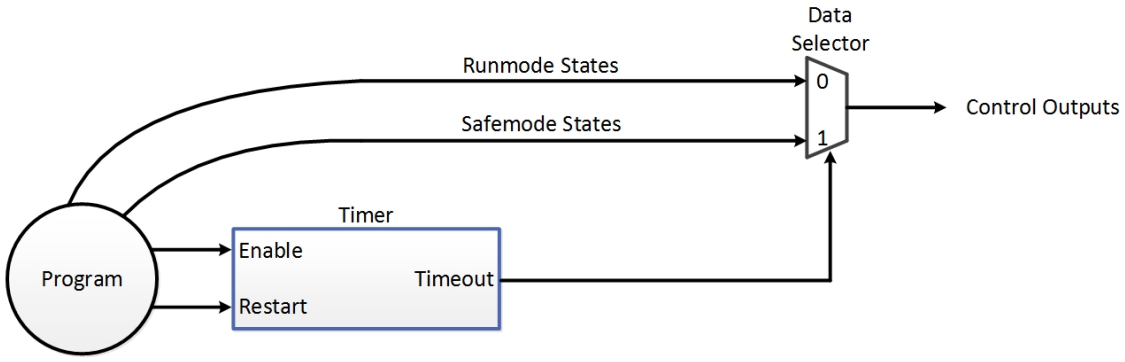


[Quelle: Eigene Darstellung in Anlehnung an J. Lamberson: *Single and Multistage Watchdog Timers.* http://www.sensoray.com/downloads/826_WatchdogWhitePaper.pdf, Sensoray, Tigard, OR, 2012, S. 2, Zugriff: 26.06.2013]

Fail-Safe Systems in Multistage Watchdogs

Bei einem Watchdog Ergebnis müssen die Control Outputs unverzüglich in einen sicheren Zustand gebracht werden, was bei einem Restart auch automatisch passiert [47].

Abbildung 21: Fail-Safe Systems in Multistage Watchdogs



[Quelle: Eigene Darstellung in Anlehnung an J. Lamberson: *Single and Multistage Watchdog Timers.* http://www.sensoray.com/downloads/826_WatchdogWhitePaper.pdf, Sensoray, Tigard, OR, 2012, S. 3, Zugriff: 26.06.2013]

Ein Multistage (Mehrphasen) Watchdog besteht aus mindestens zwei oder mehr Zähler, und das erste Timeout Signal führt nicht sofort zu einem Reset, sondern verzögert ihn nur. Multistage Watchdogs bestehen deshalb aus einer speziellen Schaltung und das zu überwachende System kennt zumindest zwei Betriebszustände, den normalen Betriebszustand (Runmode State) und den sicheren Betriebszustand (Safemode State). Beim Auslösen des ersten Timeout Signals wechselt das System in diesen Safemode und die Control Outputs werden durch einen eigenen *Data Selector* der Kontrolle des Watchdogs übergeben, damit dieser die Outputs in sicheren Zustand schalten kann [47].

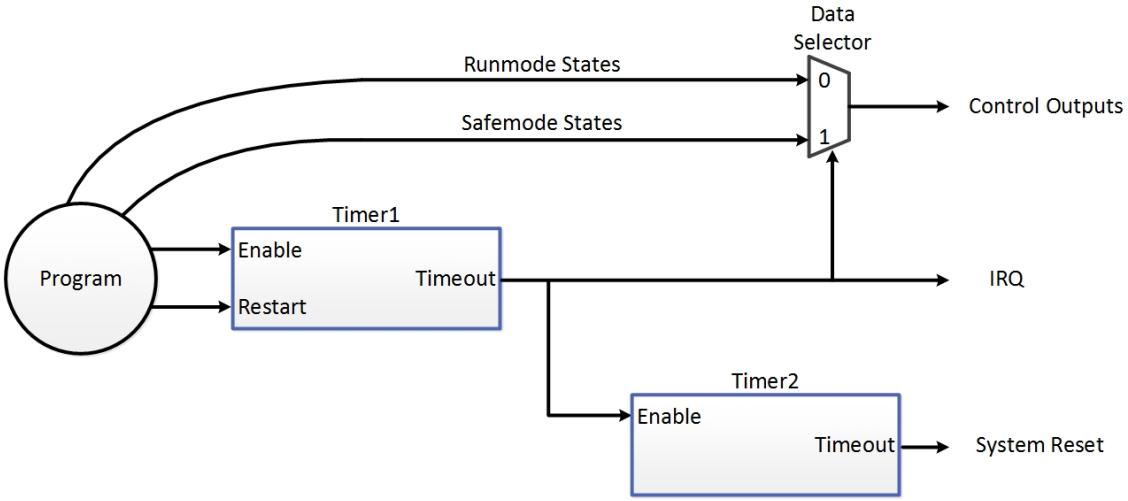
Two-stage Watchdog

Abhängig vom Anwendungsgebiet kann ein abrupter Systemneustart kostspielig auch im Sinne von Ausfallzeit und/oder Verlust von Zustandsinformationen sein. Mehrphasen Watchdogs reduzieren die Wahrscheinlichkeit für das Auftreten solcher Kosten. Der Two-stage Watchdog löst beim Timeout des

ersten Zählers noch nicht den System Reset aus, sondern verzögert ihn. Im Safemode werden die Control Outputs in sicheren Zustand gebracht und erst nach Ablauf eines weiteren Zählers wird ein Reset ausgelöst. Dadurch wird dem System noch Zeit eingeräumt, um den Fehler zu beseitigen und den kostspieligen Restart abzuwenden. Kann der Fehlerzustand nicht korrigiert werden, können aber trotzdem noch Zustands- und Fehlerinformationen gespeichert werden [47].

Die Abbildung 22 stellt die Architektur eines Zwei-Phasen Watchdog dar. Während des normalen Betriebszustands verhindert ein regelmäßiges Zurücksetzen von Timer1 ein Watchdog Ereignis. Timer2 bleibt deaktiviert und die Control Outputs stehen unter der Kontrolle des Programms.

Abbildung 22: Two-stage Watchdog



[Quelle: Eigene Darstellung in Anlehnung an J. Lamberson: *Single and Multistage Watchdog Timers.* http://www.sensoray.com/downloads/826_WatchdogWhitePaper.pdf, Sensoray, Tigard, OR, 2012, S. 3, Zugriff: 26.06.2013]

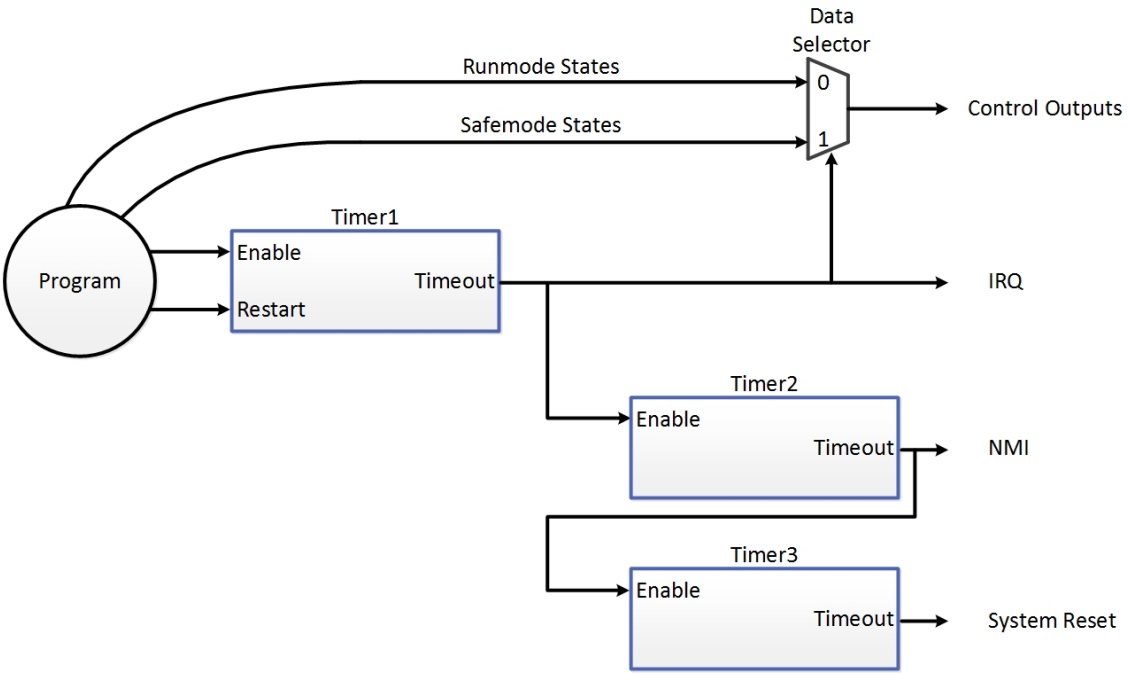
Tritt jedoch ein Fehler auf, und es kommt zu einem Überlauf bei Timer1, wird Timer2 gestartet und gleichzeitig in den Safemode State gewechselt sowie die Control Outputs in einen sicheren Zustand geschaltet. Über einen maskierbaren Interrupt (IRQ) wird eine Interrupt Service Routine (ISR) aufgerufen, die versucht, den Fehlzustand zu beheben. Kann das System auf die ISR reagieren und den Fehlzustand beseitigen, kann der Timer1 wiedergestartet und damit

auch Timer2 deaktiviert werden. Ist der Fehler nicht korrigierbar, kann ebenso in dieser ISR versucht werden, Zustands- und Fehlerinformationen aufzuzeichnen, bevor Timer2 ebenfalls überläuft und endgültig ein Reset ausführt[47].

Three-stage Watchdog

In Abbildung 23 ist die Architektur eines Drei-Phasen Watchdogs zu sehen. Er unterscheidet sich vom Zwei-Phasen Watchdog durch einen zusätzlichen Counter. Im Normalbetrieb, wo Timer1 stets rechtzeitig zurückgesetzt wird, sind Timer2 und Timer3 deaktiviert und die Control Outputs unter Programmkontrolle [47].

Abbildung 23: Three-stage Watchdog



[Quelle: Eigene Darstellung in Anlehnung an J. Lamberson: *Single and Multistage Watchdog Timers.* http://www.sensoray.com/downloads/826_WatchdogWhitePaper.pdf, Sensoray, Tigard, OR, 2012, S. 4, Zugriff: 26.06.2013]

Wie beim Two-stage Watchdog führt ein Timeout von Timer1 zum gleichzeitigen Start von Timer2, einer IRQ und dem Schalten der Control

Outputs in einen Fail-Safe Zustand. Falls das System auf die ISR reagieren kann bevor Timer2 zum Timeout kommt, wird Timer1 neugestartet, Timer2 deaktiviert und das System wechselt wieder in den Runmode State. Wenn das System nicht oder zu lange braucht, um auf die IRQ zu reagieren, löst ein Timeout von Timer2 einen nichtmaskierbaren Interrupt (NMI) aus und Timer3 wird gestartet. Der nichtmaskierbare Interrupt zeigt dem System an, dass ein Neustart unausweichlich und kurz bevor steht, und bei Ablauf von Timer3 jedenfalls durchgeführt wird. Falls das System entsprechend konfiguriert ist, reagiert es auf den NMI mit dem Speichern der wichtigsten Informationen [47].

Sicherheitsfahrschaltung

In Schienenfahrzeugen wird mit einer Totmanneinrichtung überprüft, ob der Triebfahrzeugführer noch handlungsfähig ist. Dieser muss während der Fahrt in kurzen Zeitintervallen die Totmanntaste, häufig ein Knopf oder ein Pedal, betätigen. Bleibt diese Handlung aus, erfolgt eine optischen und eine akustischen Warnung. Erfolg daraufhin immer noch keine Reaktion des Triebfahrzeugführers, wird der Zug automatisch zum Stillstand gebracht [23][37].

Diese Sicherheitsfunktion kann mit einem mehrphasigen Watchdog verglichen werden. Die Betätigung der Totmanntaste entspricht dabei dem Zurücksetzten des Watchdog Zählers.

9. Ruhestromprinzip

Die Bezeichnungen *Arbeitsstrom* und *Ruhestrom* werden meist im Zusammenhang mit Schaltungen verwendet, die neben einen Normalzustand („Ruhezustand“), auch einen außerordentlichen Zustand (z.B. Alarm, Störung, etc.) kennen [75]. Beim Arbeitsstrom wird nur ein Schaltvorgang ausgelöst, falls ein Kontakt betätigt wird, während beim Ruhestrom erst eine Unterbrechung des Stroms einen Schaltvorgang auslöst. Ruhestrom bedeutet also, dass im Normalzustand der elektrische Strom fließt, welcher dann im außerordentlichen Zustand unterbrochen wird.

Drahtbruchsicherheit
Ein Drahtbruch in einem geschlossenen elektrischen Stromkreis verursacht eine Unterbrechung desselben. Die Bezeichnung Drahtbruch stammt ursprünglich aus der Elektrotechnik. Er definiert einen Fehlerzustand, wo der Stromkreis unterbrochen und nicht betriebsbereit ist.
Der Begriff Drahtbruchsicherheit ist ein Sicherheitsmerkmal von elektrischer Steuerungstechnik und Steuerungen und ist gegeben, wenn keine Unterbrechung eines Leitungsdrahts zum Einschalten führt oder das Ausschalten verhindert. Drahtbruchsicherheit bedeutet auch, dass ein eventueller Drahtbruch von einer Steuerung als Fehler erkannt wird und er keine Fehlfunktion auslöst [61]. Besonders wichtig ist der drahtbruchsichere Anschluss bei allen Halt-Funktionen (Halt-Taster, Aus-Taster) und beim Not-Aus: Not-Aus und Halt-Funktionen müssen immer als Öffner angeschlossen werden. Bricht der Draht, gilt dies als Halt-Signal, die Maschine stoppt und der Fehler kann gesucht werden. Würde anders verdrahtet, könnte die Maschine bei einem Drahtbruch nicht mehr angehalten werden.

Arbeitsstromprinzip
Beim Arbeitsstromprinzip fließt der Strom nur, falls ein Signal aktiv ist oder ein Element betätigt wird. Ein solches Element kann zum Beispiel ein Schalter sein, der erst in Wirkstellung geht, wenn eine Ansprechspannung anliegt [22].
Für Sicherheitsanwendungen ist es sofort einleuchtend, dass das Arbeitsstromprinzip nicht geeignet ist. Eine Stromunterbrechung, ein

Energieverlust oder ein Drahtbruch (z.B. durch Unfall oder Sabotage) könnte die Funktionalität der Sicherheitsanwendung negativ beeinflussen.

Anwendung des Ruhestromprinzips

Das Ruhestromprinzip folgt dem Prinzip des Fail-Safe-Designs und geht über den Begriff der Drahtbruchsicherheit hinaus [4][9].
Grundlage bildet die Zuordnung der beiden Systemzustände. Der gefährliche Systemzustand ist stets dem energiereicheren Betriebszustand zugeordnet, der sichere Systemzustand dem energieärmeren bzw. energielosen Systemzustand. Bei einem möglichen Ausfall wird mit Hilfe einer unverlierbaren Eigenschaft des Systems automatisch der sichere Betriebszustand eingenommen [72].
Durch das Ruhestromprinzip kann gewährleistet werden, dass bei Störungen, insbesondere an sicherheitsrelevanten Komponenten, das System fail-safe reagiert.

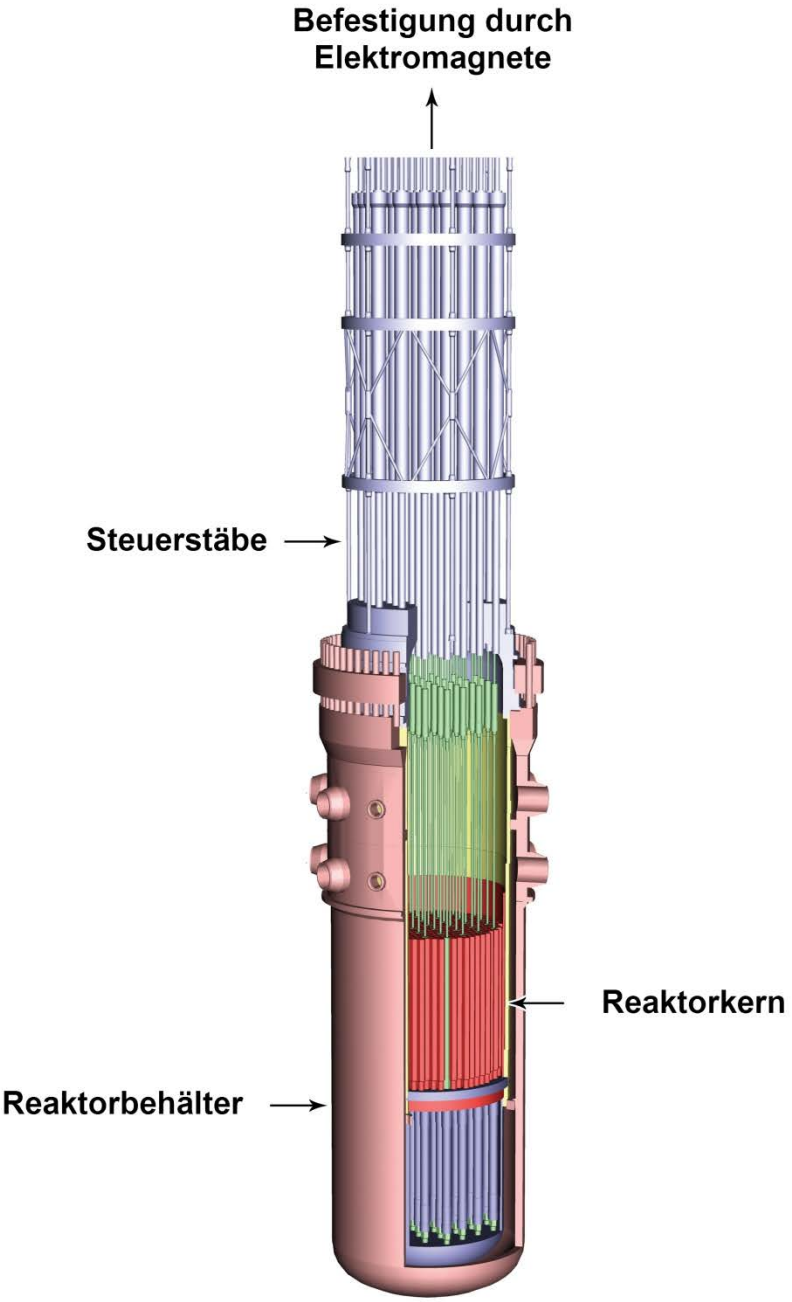
Nachteile des Ruhestromprinzips

Eine Implementierung nach dem Ruhestromprinzip bedeutet neben höherem Designaufwand meist auch höhere Kosten für Errichtung und Betrieb. Bei elektrischen Steuerungen und Schaltungen wird im Ruhezustand permanent Strom verbraucht. Alle Komponenten, die sich im Ruhestromkreis befinden, stehen unter ständiger Belastung (Überlastungsgefahr), was einen Mehraufwand bei Kühlung, Wartung und Instandhaltung bedeutet [75].

Anwendungsbeispiel Kernkraftwerk

In Kernkraftwerken werden Steuerstäbe zur Regelung und Abschaltung verwendet. Durch mehr oder weniger tiefes Einfahren der Steuerstäbe in den Reaktorkern kann die Reaktionsrate der Kettenreaktion und somit auch die Leistung des Atomreaktors reguliert werden. Das vollständige Einfahren der Steuerstäbe stoppt komplett die Kettenreaktion im Reaktorkern, und der Reaktor wird damit sicher abgeschaltet. Im Normalbetrieb befindet sich immer ein Teil der Steuerstäbe außerhalb des Reaktorkerns und kann im Notfall zur sichern Abschaltung verwendet werden [40][56].

Abbildung 24: Steuerstäbe in einem Kernkraftwerk



[**Quelle: modifiziert nach MVM Paks Nuclear Power Plant Ltd: *The nuclear reactor*. http://paksnuclearpowerplant.com/download/1132/VVER-440-213%20reactor.jpg, Zugriff: 27.06.2013**]

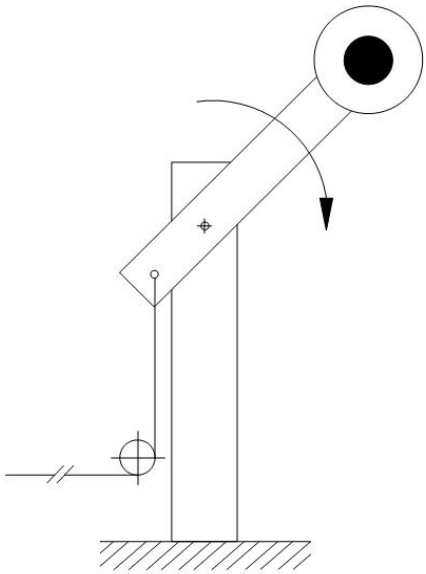
Das Ruhestromprinzip kommt bei der Befestigung dieser Steuerstäbe zur Anwendung. Diese sind mit Hilfe von Elektromagneten an der Decke des

Reaktors über dem Reaktorkern montiert. Kommt es zu einer Notabschaltung oder zu einem Stromausfall, schalten die Elektromagnete aus und die Steuerstäbe fallen durch die Schwerkraft in den Reaktorkern zwischen die Brennelemente [40]. Die Schwerkraft wird als unverlierbare Eigenschaft eingesetzt, um die Kernreaktion zu stoppen.

Anwendungsbeispiel Eisenbahnsignaltechnik

In der Eisenbahnsignaltechnik zeigen u.a. mechanische Formsignale entlang einer Eisenbahnstrecke dem Triebfahrzeugführer an, ob ein Streckenabschnitt frei und befahrbar ist.

Abbildung 25: Fahrtsignal (fail-safe)



[Quelle: eigene Darstellung in Anlehnung an G.-H. Schildt, W. Kastner: *Prozeßautomatisierung*. Springer, Wien, 1998, S. 56]

Abbildung 26: Formsignal bei Radevormwald

[Quelle: modifiziert nach F. Vincentz; Radevormwald. http://upload.wikimedia.org/wikipedia/commons/8/80/Radevormwald_Dahlhausen_-_Eisenbahnmuseum_02.jpg, Zugriff: 27.06.2013]

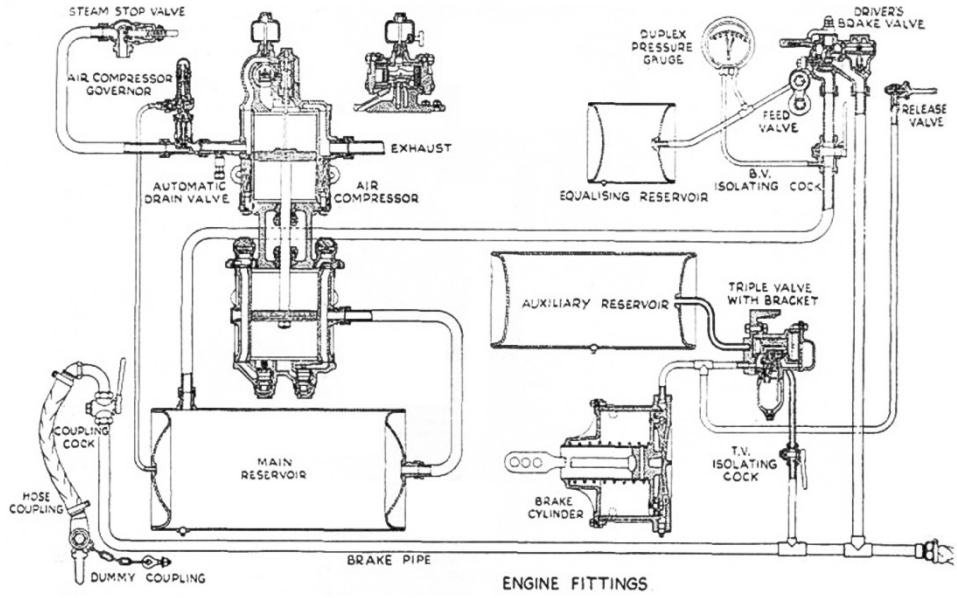
Diese mechanischen Eisenbahnsignale haben sicherheitstechnisch eine große Bedeutung und werden deshalb auch nach dem Ruhestromprinzip realisiert [22].

Die Abbildung 25 zeigt den prinzipiellen Aufbau eines Signalarms, Abbildung 26 zeigt ein Formsignal der Deutschen Bahn bei Radevormwald in Nordrhein-Westfalen, Deutschland.
Ein Signalarm, der schräg nach rechts oben zeigt, zeigt freie Fahrt an, wogegen ein waagrechter und nach unten zeigender Signalarm „Stopp“ bedeutet [43].
Über einen Seilzug wird der Signalarm in einen Zustand höherer potentieller Energie angehoben und gehalten. Sollte das Betätigungssignal reißen, fällt der Signalarm aufgrund seines Gewichtes und der unverlierbaren Eigenschaft der Schwerkraft in den energieärmeren Lagezustand, der gleichzeitig ein sichererer Betriebszustand[23] ist [72].

Anwendungsbeispiel Druckluftbremse (Westinghouse-Bremse)
Die Druckluftbremse wird vor allem zur Bremsung von modernen Eisenbahnfahrzeugen verwendet [25]. Moderne Druckluftbremsen beruhen auf dem von George Westinghouse 1872 patentierten Design [79].
Dieses Design verwendet das Ruhestromprinzip und besteht aus einem System von Druckluftbehältern, Bremszylindern und Druckluftleitungen [17].
Im energiereicheren Zustand sind alle Hilfsluftbehälter gefüllt und in der Hauptluftleitung herrscht ein Regelbetriebsdruck, die Bremsen sind gelöst und inaktiv. Durch Absenkung des Drucks in der Hauptluftleitung werden durch die Bremszylinder die Bremsscheiben ans Rad bzw. die Bremszange an die Scheibenbremsen gepresst und das Schienenfahrzeug gebremst [79]. Der drucklose und gleichzeitig energieärmere Zustand ist in jeden Fall fail-safe, da alle Bremsen angezogen bzw. aktiviert sind.

[23] Voraussetzung ist eine hinreichende Wartung und Schmierung des Gelenks.

Abbildung 27: Schematische Darstellung der Westinghouse-Bremse



[Quelle: EngRailHistory: Stopping the Train. *The Working of the Westinghouse Automatic Air-Brake.* http://www.engrailhistory.info/r136.html, Zugriff: 27.60.2013]

10. Dynamisierung von Steuersignalen

Signal

Signale sind in der Fachsprache die physikalische Darstellung von Nachrichten oder Daten.[24] Unter einem Signal versteht man allgemeinen ein Zeichen mit einer bestimmten Bedeutung. In der Regel dienen Signale dem Transport einer Information bzw. einer Nachricht zwischen einem Sender und Empfänger mit Hilfe eines Übertragungsmediums [63].
Ein elektrisches Signal ist die Darstellung einer Nachricht durch änderbare Parameter elektrischer Größen wie Strom, Spannung oder Widerstand. Beispiele für solche Parameter sind etwa Tastgrad, Phasenlage, Frequenz, Gleichwert und Scheitelwert.
Die Nachrichtentechnik ist eine Ingenieurwissenschaft, die sich mit der Gewinnung, Umwandlung, Übertragung, Vermittlung, Speicherung und Verarbeitung von informationstragenden Signalen beschäftigt [12].

Dynamisches Signal

In der Zuverlässigkeitstechnik wird unter einem dynamischen Signal ein Signal verstanden, das dauernd seinen Spannungs- oder Stromwert ändert und damit z.B. in Form einer Impulsfolge vorliegt [68].

Steuersignale

Nicht nur in technischen Systemen spielen Signale eine elementare Rolle. In sicherheitsrelevanter Elektronik sowie bei einkanaligen Sicherheitssystemen ist es notwendig, dass die logische Verarbeitung digitaler Signale nach dem Fail-Safe-Prinzip durchgeführt wird.
Dabei wird entsprechend dem Ruhestromprinzip einem binären Signal die Information so zugeordnet, dass die Information log. 1 dem energiereichen und log. 0 dem energiearmen Signal zugewiesen wird. Semantisch entspricht dem ‚GO'-Signal (Enable) log. 1 und dem ‚NO-GO'-Signal (Disable) log. 0.

[24] W. E. Proebster: Rechnernetze. Technik, Protokolle, Systeme, Anwendungen. Oldenbourg Verlag, 2.Auflage, München/Wien/Oldenbourg, 2002, S. 59

In Sicherheitssystemen darf kein falsches logisches 1-Signal auftreten, da dies fatale Folgen haben kann [72].

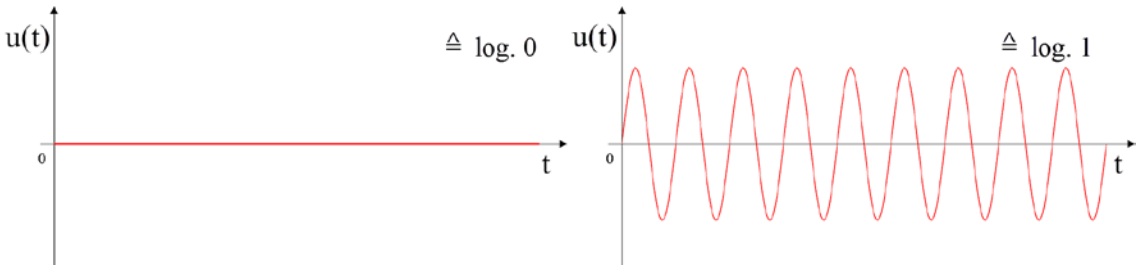
Problem der Bitsynchronisation

Bei der Übertragung von Steuersignalen stellt die Bitsynchronisation ein wesentliches Problem dar. Da der Empfänger wissen muss, in welchen Abständen Signale/Bits ankommen, muss er das verwendete Übertragungsverfahren und die zugehörige Übertragungsrate kennen. Zusätzlich benötigen Sender als auch Empfänger einen gemeinsamen Takt. In der Regel sind regelmäßige „1“/„0“-Folgen gut erkennbar. Bei langen „0“- und „1“-Folgen hingegen ist die Anzahl von Bits in einer solchen Folge nicht immer klar erkennbar [11].

Dynamisierung und Gleichrichtung

Die Verwendung von Wechselspannungssignalen mit mäanderförmigem oder sinusförmigen Verlauf zur Darstellung der logischen Werte „0“ und „1“ (sog. Dynamische Signale) bringt den Vorteil, dass die Ausfallart „Unterbrechung“ eindeutig vom logischen Wert „0“ unterscheidbar ist. Ebenso kann durch die Gleichrichtung von dynamischen Signalen jede Signalunterbrechung erkannt werden, wenn die so erzeugte Gleichspannung als Stromversorgung verwendet wird [48]. Sowohl ein Kurzschluss als auch eine Unterbrechung der Bauelemente macht sich im Signal bemerkbar [68].

Abbildung 28: Zuordnung der Signale



[Quelle: eigene Darstellung in Anlehnung an G.-H. Schildt, W. Kastner: *Prozeßautomatisierung*. Springer, Wien, 1998, S. 57]

Die Abbildung 28 zeigt die Zuordnung der Signale, wobei die Information log. 1 eindeutig mit einem Wechselspannungssignal verknüpft ist, während log. 0 durch keine Wechselspannung dargestellt wird.

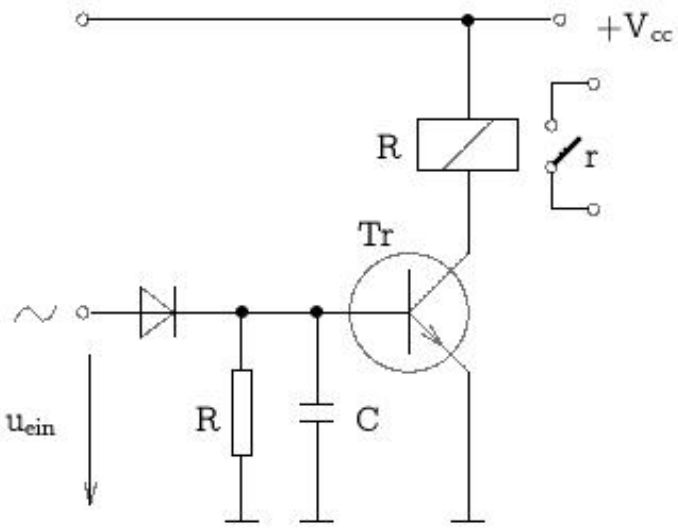
Diese Dynamisierung der Signale erfordert anspruchsvolle und durchdachte Elektronik. Herkömmliche Schaltungsfamilien erfüllen diese hohen Ansprüche in der Regel nicht.

Funktionsschaltung mit Transistorstufe

Steuerungssysteme, die aus Halbleitern aufgebaut sind, dienen meist dem Zweck, an Prozessperipherie ein Relais anzusteuern. Die elektrische Leistung, die notwendig ist, um dieses Relais zu schalten, wird dabei durch eine Transistor-Verstärkerschaltung bereitgestellt [72].

Die Abbildung 29 und Abbildung 30 zeigen solche Funktionsschaltungen mit einer Transistorstufe [69].

Abbildung 29: Funktionsschaltung einer Transistor-Verstärkerstufe

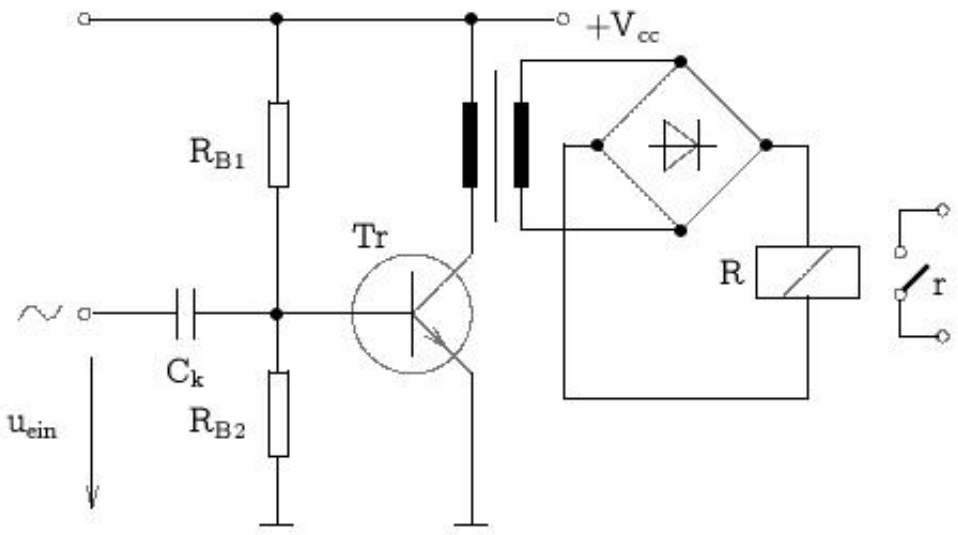


[Quelle: G.-H. Schildt, D. Kahn: *Fundamental principles of safety related system design.* In: Electrical and Computer Engineering Series - Recent Advances in Intelligent Systems and Signal Processing, World Scientific Engineering Academy Society Press, Corfu, 2003, S. 76]

Ein Kurzschluss zwischen Kollektor und Emitter in der ersten Schaltung würde ein unerwünschtes Einschalten des Relais verursachen [72]. Ein solches Verhalten kann in sicherheitsrelevanten Systemen möglicherweise zu einem gefährlichen Zustand führen.

Die Schaltung in Abbildung 30 arbeitet dagegen fail-safe [69]. Aufgrund des Wechselspannungssignals am Eingang wird der Transistor periodisch sperrend und leitend gesteuert. Hierdurch fließt im Kollektorzweig ein Wechselstrom, der mit Hilfe eines Transformators übertragen wird. Der umgewandelte Gleichstrom auf der Sekundärseite schaltet schließlich das Relais.

Abbildung 30: Fail-safe Schaltung



[**Quelle: G.-H. Schildt, D. Kahn: *Fundamental principles of safety related system design.* In: Electrical and Computer Engineering Series - Recent Advances in Intelligent Systems and Signal Processing, World Scientific Engineering Academy Society Press, Corfu, 2003, S. 76**]

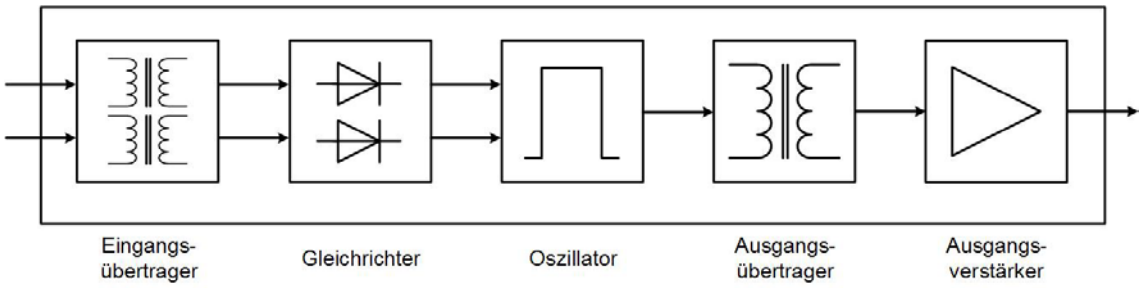
Der Ausfall einer Komponente dieser Schaltung hätte zur Folge, dass keine Energie mehr auf die Sekundärseite übertragen wird. Ein unerwünschtes Einschalten des Relais aufgrund eines Komponentendefekts kann somit ausgeschlossen werden.

LOGISAFE

Die Logikfamilie LOGISAFE von AEG ist ein Schaltkreissystem für höchste Sicherheitsanforderungen. Sie stellt modular aufgebaute Standardbaugruppen zur Verfügung, mit denen fehlersichere Schaltungen realisiert werden können [31].
Unter fehlersicherer Schaltungstechnik versteht man eine spezielle Schaltungsart, die die Eigenschaft besitzt, dass sich statische Bauteilfehler immer ungefährlich auswirken und sich deutlich bemerkbar machen [31].

Den prinzipiellen Aufbau eines LOGISAFE-Moduls zeigt Abbildung 31.

Abbildung 31: Aufbau eines LOGISAFE-Moduls



[Quelle: eigene Darstellung in Anlehnung an W. Halang, R. Konakovsky: *Sicherheitsgerichtete Echtzeitsysteme*. Oldenbourg, München, 1999, S. 36]

Das Wesentliche am LOGISAFE Systems sind die Ringkerntransformatoren (Übertrager), die in jedem Baustein das Fail-safe Verhalten garantieren. Nur fehlerfreie dynamische Signale können übertragen und im Baustein verarbeitet werden. Statische Signale, die den logisch Wert "0" darstellen oder durch Komponentenausfall entstanden sind, können ein fehlerfreies dynamisches Signal, das dem logischen Wert „1“ entspricht, nicht erzeugen [31].

11. Zusammenfassung

Die immer umfassendere Automatisierung komplexer technischer Systeme erhöht zugleich auch die Wahrscheinlichkeit von Fehlern und potentiell gefährlichen Situationen. Die Minimierung möglicher Gefahren und die Reduzierung des Risikos auf ein akzeptierbares Niveau stellen große Herausforderungen an die Entwicklung dar.

Die EN 61508 bietet hier normative und informative Richtlinien für alle sicherheitsgerichteten Systeme. Es werden alle Aspekte von der Planung bis zur Außerbetriebnahme betrachtet. Im Mittelpunkt dieser Europäischen Norm stehen die Entwicklungsprozesse, die Qualitätssicherung, die erforderliche Dokumentation, die Betriebsarten für Sicherheitsfunktionen und die Safety Integrity Levels.

Die ISO 26262 ist eine speziell für die Automobilindustrie entwickelte Norm und löst die EN 61508 in diesem Bereich ab. Eine Adaptierung war notwendig, da bei der Entwicklung von Serienfahrzeugen eine Validierung erst nach Fertigstellung, wie von der EN 61508 gefordert, ein großes Problem darstellte. Die Automotive Safety Integrity Level als Maß für die Sicherheitsrelevanz eines Bauteils wird hier vorgestellt. Die Einschätzung des Risikos basiert auf einer Kombination der Schwere des möglichen Schadens, der Häufigkeit der Fahrsituation und der Beherrschbarkeit durch den Fahrer.

Der Einsatz von FMECA ist ebenfalls eine Möglichkeit, um die Sicherheit von technischen Systemen zu erhöhen. Nach Erfüllung aller FMECA Voraussetzungen wird das zu untersuchende System in der Systemstrukturanalyse zerlegt. In FMECA Arbeitsblättern werden mögliche Fehlzustände, Ausfallursachen, -auswirkungen und -erkennungsmethoden festgehalten. Das Risiko wird mit der Risikoprioritätszahl oder einer Risikomatrix/Kritizitätsmatrix bestimmt. In Team Reviews werden schließlich Korrekturmaßnahmen und Verbesserungen erarbeitet.

Das Sichere Mikrocomputersystem SIMIS wird in elektronischen Stellwerken eingesetzt und ersetzt zunehmend elektrische Relais-Stellwerke in der Eisenbahnsignaltechnik. Parallel arbeitende Rechner, die unabhängig voneinander die gleichen Eingabedaten erhalten, bilden den grundlegenden Aufbau. Werden Abweichungen zwischen den Rechnern während des Datenverarbeitungsprozesses festgestellt, wird das System in einen sicheren Abschalt- bzw. Betriebszustand geführt. Fehler mit gemeinsamer Ursache (Common Cause Failure) werden mit unabhängiger Stromversorgung sowie galvanischer und räumlicher Trennung der Prozessrechner vermieden.

Der Funktionswächter oder Watchdog ist ein sicherheitstechnisches Prinzip zur Überwachung von technischen Systemen. Er besteht aus einem Counter, der beim Systemstart beginnt, zu einem festgelegten Zielwert zu zählen. Dieser Zähler muss vom System regelmäßig und rechtzeitig zurückgesetzt werden, sonst wird ein Hardware-Reset ausgelöst. Watchdogs sind ein wichtiger Schutz gegen unvorhergesehene Ereignisse und von unschätzbaren Wert, wenn menschliche Benutzer nicht schnell genug auf Fehlzuständige reagieren können.

Das Ruhestromprinzip gewährleistet, dass bei Störungen, insbesondere an sicherheitsrelevanten Komponenten, das System fail-safe reagiert. Es wird zwischen zwei Systemzuständen unterschieden, dem gefährlichen und dem sicheren Betriebszustand. Der gefährliche Systemzustand ist stets der energiereichere Zustand. Bei einem Ausfall oder einer Störung wird mit Hilfe einer unverlierbaren Eigenschaft der energieärmere und zugleich sichere Systemzustand eingenommen.

Die Dynamisierung von Steuersignalen verhindert, dass im Fehlerfall ein ‚NO-GO'-Signal mit einem ‚GO'-Signal verwechselt wird. Hierbei werden Wechselspannungssignalen mit mäanderförmigem oder sinusförmigen Verlauf zur Darstellung der logischen Werte „0" und „1" verwendet. Ein Kurzschluss ist eindeutig vom logischen Wert „0" unterscheidbar und das Problem der Bitsynchronisation von langen „0"- und „1"-Folgen wird damit gelöst.

Die Einhaltung und Verwendung diesen hier beschriebenen Normen, Richtlinien und Prinzipien kann die Sicherheit von technischen Systemen wesentlich erhöhen.

12. Conclusio und Ausblick

Neue Technologien entstehen stets vor einer Norm. Der Entstehungsprozess einer Norm kann sehr langwierig werden und die so entstandene Norm selbst hält nur den „Stand der Technik“ fest. Zukünftige Entwicklungen können nur sehr eingeschränkt vorhergesehen werden. Die Einhaltung von Normen und der Nachweis der funktionalen Sicherheit aufgrund dieser kann deshalb nur als zu erreichendes Minimum verstanden werden.

Der stetige Komplexitätsanstieg macht den Nachweis der funktionalen Sicherheit immer aufwendiger und umfangreicher. Sowohl für die Betriebszulassung, als auch aus versicherungstechnischer Sicht ist dieser Nachweis aber essentiell. Um diesem Umstand Rechnung zu tragen, wird dieser Nachweis in naher Zukunft verstärkt mit Hilfe von Software-Tools erbracht. Hier stellt sich aber die Frage, inwieweit diese Software-Tools selbst fehlerfrei sind und wie diese Fehlerfreiheit nachgewiesen werden kann.

Der Betrieb vieler Systeme ist immer mit einem (tolerierbaren) Restrisiko verbunden. Trotz aller Vorkehrungen darf auch der Mensch als Fehlerquelle nicht vernachlässigt werden. Möglicherweise ist das auch ein Grund, warum es so schwierig ist, eine Einheit für die Zuverlässigkeit des Gesamtsystems zu entwickeln.

13. Abkürzungsverzeichnis

AEG	Allgemeine Elektricitäts-Gesellschaft
AIS	Abbreviated Injury Scale
ALARP	As Low As Reasonably Practicable
ASIC	Application Specific Integrated
ASIL	Automotiv Safety Integrity Level
CENELEC	European Committee for Electrotechnical Standardization
CPU	Central Processing Unit
DIN	Deutsches Institut für Normung
E/E	Elektrisch/Elektronisch
E/E/PE	Elektrisch/Elektronisch/Programmierbar Elektronisch
EN	Europäische Norm
EUC	Equipment Under Control
FAA	Federal Aviation Administration
FMEA	Failure Mode And Effects analysis
FMECA	Failure Modes, Effects, and Criticality Analysis
FRACAS	Failure Reporting, Analysis And Corrective Action System
I/O	Input/Output
IC	Integrated Circuit
IEC	International Electrotechnical Commission
IRQ	Interrupt
ISO	International Organization For Standardization
ISR	Interrupt Service Routine
LOPA	Layer Of Protection Analysis
MIL-P	Military Procedure
MIL-STD	Military Standard
NASA	National Aeronautics And Space Administration
NMI	Non-Maskable Interrupt
PFD	Probability Of Failure On Demand
PFH	Probability Of Dangerous Failure Per Hour
QM	Qualitätsmanagement
RAM	Random-Access Memory

ROM	Read-Only Memory
RPN	Risk Priority Number
SAE	Society For Automotive Engineers
SIL	Safety Integrity Level
SIMIS	Sicheres Mikrocomputersystem
SITAG	Sicherer Taktgeber
SOPP	SIMIS-Online Prüfprogramm
TÜV	Technischer Überwachungsverein
V&V	Verifikation & Validierung
WDT	Watchdog Timer

14. Abbildungsverzeichnis

15. Tabellenverzeichnis

16. Literaturverzeichnis

[1] C. Ackermann: *Die Mond-Sonde Clementine.* http://www.raumfahrer.net/raumfahrt/raumsonden/clementine.shtml (zuletzt besucht 04.Juni 2013), Raumfahrer Net e.V., 2001

[2] Associated Press: *Pathfinder's computer reprogrammed to avoid glitches.* http://chronicle.augusta.com/stories/1997/07/14/tec_211345.shtml (zuletzt besucht 04.Juni 2013), The Augusta Chronicle, Augusta, GA, 1997

[3] Atmel Corporation: *ATmega8(L) datasheet.* http://www.atmel.com/devices/atmega8.aspx?tab=documents, 2013

[4] Aviation Glossary: *Fail-Safe Design Concept.* http://aviationglossary.com/fail-safe-design-concept/ (zuletzt besucht 03.Juni 2013)

[5] M. Barr: *Introduction to Watchdog Timers.* http://www.embedded.com/electronics-blogs/beginner-s-corner/4023849/Introduction-to-Watchdog-Timers (zuletzt besucht 03.Juni 2013), UBM Electronics, 2001

[6] M. Baumann, P. Fuhrmann, R. Mariani: *A single channel, fail-safe microcontroller to simplify SIL3 safety architectures in automotive applications.* http://home.arcor.de/mbaumeister/veroeff/BadenBaden2007-Final.pdf (zuletzt besucht 03.Juni 2013), 2007

[7] A. Beer, M. Rau: *Qualität und Zuverlässigkeit elektronischer Systeme am Beispiel von x-by-wire Systemen.* TÜV SÜD, 2003

[8] J. Börcsök: *Elektronische Sicherheitssysteme: Hardwarekonzepte, Modelle und Berechnung.* Hüthig, 2.Auflage, Heidelberg, 2007

[9] J. Börcsök: *Funktionale Sicherheit: Grundzüge sicherheitstechnischer Systeme.* VDE-Verlag, 3.Auflage, Berlin, 2011

[10] R. Borgovini, S. Pemberton, M. Rossi: *Failure Mode, Effects and Criticality Analysis (FMECA).* Reliability Information Analysis Center (RIAC), Rome, NY, 1993

[11] C. Bormann, J. Ott: *Konzepte der Internet Technik.* SPC TEIA Lehrbuch Verlag, o.O., 2002

[12] M. Bossert: *Einführung in die Nachrichtentechnik*. Oldenbourg Verlag, München, 2012

[13] T. A. Carbone, D. D. Tippett: *Project Risk Management Using the Project Risk FMEA*. In: Engineering Management Journal, Jg. 16, Nr. 4, Dezember 2004, S. 28-35

[14] Department of Defense: *MIL-STD-1629A Procedures for performing a Failure Mode, Effects, and Criticality Analysis*. United States Department of Defense (DoD), Washington, DC, 1980

[15] Department of the Army: *TM 5-698-4 Failure Modes, Effects and Criticality Analyses (FMECA) for Command, Control, Communications, Computer, Intelligence, Surveillance, and Reconnaissance (C4ISR) Facilities.* United States Department of the Army, Washington, DC, 2006.

[16] Endress+Hauser GmbH: *DIN EN IEC 61508/IEC 61511. Funktionale Sicherheit in der Prozess-Instrumentierung zur Risikoreduzierung.* http://www.endress.com/eh/productDBs/homeDBs/resourceadditional.nsf/imgref/Download_SIL_Broschuere_de.pdf/$FILE/SIL_Broschuere_de.pdf (zuletzt besucht 03.Juni 2013), 2004

[17] EngRailHistory: *Stopping the Train. The Working of the Westinghouse Automatic Air-Brake.* http://www.engrailhistory.info/r136.html (zuletzt besucht 04.Juni 2013), 2012

[18] European Committee for Electrotechnical Standardization (CENELEC): *EN 50129 Bahnanwendungen. Telekommunikationstechnik, Signaltechnik und Datenverarbeitungssysteme - Sicherheitsrelevante elektronische Systeme für Signaltechnik.* 2004

[19] European Committee for Electrotechnical Standardization (CENELEC): *EN 60812 Analysetechniken für die Funktionsfähigkeit von Systemen. Verfahren für die Fehlzustandsart- und -auswirkungsanalyse (FMEA).* 2006

[20] European Committee for Electrotechnical Standardization (CENELEC): *EN 61508 Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme.* 2010

[21] exida.com LLC: *IEC 61508 Overview Report. A Summary of the IEC 61508 Standard for Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems.* Sellersville, PA, 2006

[22] W. Fenner, P. Naumann, J. Trinckauf: *Bahnsicherungstechnik: Steuern, Sichern und überwachen von Fahrwegen und Fahrgeschwindigkeiten im Schienenverkehr.* Publicis Publishing, 2.Auflage, Erlangen, 2004

[23] Ž. Filipović: *Elektrische Bahnen. Grundlagen, Triebfahrzeuge, Stromversorgung.* Springer, 4.Auflage, Berlin/Heidelberg, New York, 2005

[24] S. Flod, H. Landquist: *Risk and Vulnerability Assessment of Expanding the Windhoek Drinking Water Supply.* Master's thesis, Chalmers University of Technology, Göteborg, 2010

[25] J. Gabriel: *Jans Eisenbahn-Homepage.* http://www.jan-steffen-gabriel.de/ (zuletzt besucht 03.Juni 2013), 2009

[26] J. Ganssle: *A Designer's Guide to Watchdog Timers.* http://www.digikey.com/us/en/techzone/microcontroller/resources/articles/a-designers-guide-to-watchdog-timers.html (zuletzt besucht 03.Juni 2013), 2012

[27] J. Ganssle: *Great Watchdog Timers For Embedded Systems.* http://www.ganssle.com/watchdogs.htm (zuletzt besucht 03.Juni 2013), The Ganssle Group, 2011

[28] J. Ganssle: *Li'l Bow Wow.* http://www.embedded.com/electronics-blogs/watchdog-timers/4024509/Li-l-Bow-Wow (zuletzt besucht 03.Juni 2013), UBM Electronics, 2003

[29] J.-T. Gayen, H. Schäbe: *(Miss-)Konzeptionen von Sicherheitsprinzipien.* In Signal+Draht, DVV Media Group, Hamburg, Jg. 7+8, 2008, S. 11-18

[30] R. Genser: *Safety of computer control systems. SAFECOMP '89: Proceedings of the Ifacifip Workshop, Vienna, Austria, 5 - 7 December 1989.* Pergamon Press, Oxford, 1989

[31] W. Halang, R. Konakovsky: *Sicherheitsgerichtete Echtzeitsysteme.* Oldenbourg, München, 1999

[32] M.Hillenbrand: *Funktionale Sicherheit nach ISO 26262 in der Konzeptphase der Entwicklung von Elektrik/Elektronik Architekturen von Fahrzeugen.* Dissertation, Karlsruher Institut für Technologie, KIT Scientific Publishing, Karlsruhe, 2012

[33] T. Huckle: *Collection of Software Bugs.* http://www5.in.tum.de/persons/huckle/bugse.html (zuletzt besucht 04.Juni 2013), 2011

[34] Institut für Planetenforschung: *Clementine.* http://www.dlr.de/pf/desktopdefault.aspx/tabid-4357/7096_read-10567/ (zuletzt besucht 04.Juni 2013), Institut für Planetenforschung

[35] International Organization for Standardization: *ISO 26262 Road vehicles. Functional safety.* 2011/2012

[36] Jet Propulsion Laboratory: *Mars Pathfinder.* http://mars.jpl.nasa.gov/MPF/ (zuletzt besucht 04.Juni 2013), Jet Propulsion Laboratory, California Institute, National Aeronautics and Space Administrationof Technology

[37] H. Jochim,F. Lademann: *Planung von Bahnanlagen. Grundlagen - Planung - Berechnung.* Fachbuchverlag Leipzig im Carl HanserVerlag, München, 2009

[38] M. Jones: *What really happened on Mars.* http://research.microsoft.com/en-us/um/people/mbj/Mars_Pathfinder/Mars_Pathfinder.html (zuletzt besucht 04.Juni 2013), 1997

[39] P. Kafka: *DIN EN 61508 – SIL Level – Risikograph, Zuverlässigkeit, Methoden.* http://www.relconsult.de/MAN61508.pdf (zuletzt besucht 03.Juni 2013), 2011

[40] Kernkraftwerk Gösgen-Däniken AG: *Steuerelemente.* http://www.kkg.ch/de/i/steuerelemente-_content---1--1073--307.html (zuletzt besucht 03.Juni 2013)

[41] O. Kettner: *Die Bremsenbude.* http://www.bremsenbude.de/ (zuletzt besucht 03.Juni 2013), 2012

[42] W. H. Kinard: *Clementine Deep Space Probe Science Experiment.* http://setas-www.larc.nasa.gov/CLEM/dspse.html (zuletzt besucht 04.Juni 2013), NASA Langley Research Center, Hampton, VA, 2001

[43] H. Kötting: *Signalsystem Deutschland. Haupt-/Vorsignal-System.* http://www.stellwerke.de/signal/deutsch/hv.html (zuletzt besucht 03.Juni 2013), 2006

[44] E. Kruschitz: *SIL versus SAL. Risiko-Analyse für IT-Systeme.* http://www.it-production.com/index.php?seite=einzel_artikel_ansicht&id=58963 (zuletzt besucht 03.Juni 2013), Technik-Dokumentations-Verlag GmbH, 2012

[45] J.L. LaChance, S.P. Nowlen, F.J. Wyant, et al.: *Circuit Analysis. Failure Mode and Likelihood Analysis.* National Laboratories, Office of Nuclear Regulatory Research, Washington, DC, 2003

[46] P. Ladkin, B. Sieker, J. Sanders: *Safety of Computer-Based Systems.* http://www.rvs.uni-bielefeld.de/publications/books/ComputerSafetyBook/index.html (zuletzt besucht 03.Juni 2013), unveröffentlicht, public draft version 1.0, 2011

[47] J. Lamberson: *Single and Multistage Watchdog Timers.* http://www.sensoray.com/downloads/826_WatchdogWhitePaper.pdf (zuletzt besucht 03.Juni 2013), Sensoray, Tigard, OR, 2012

[48] R. Lauber, P. Göhner: *Prozessautomatisierung 1.* Springer, 3.Auflage, Berlin, 1999

[49] R. Laufer: *Die Mars Pathfinder Mission.* http://berlinadmin.dlr.de/Missions/pathfinder/ (zuletzt besucht 04.Juni 2013), Institut für Planetenforschung Berlin-Adlershof, 2004

[50] B. Leitenberger: *Raumsonden.* http://www.bernd-leitenberger.de/raumsonden.shtml (zuletzt besucht 04.Juni 2013)

[51] Lunar and Planetary Institute: *The Clementine Mission.* http://www.lpi.usra.edu/lunar/missions/clementine/ (zuletzt besucht 04.Juni 2013), 2013

[52] Maxim Integrated: *Application Note 1070: Windowed Watchdog Enhances µP Supervisors.* http://www.maximintegrated.com/app-notes/index.mvp/id/1070 (zuletzt besucht 03.Juni 2013), Maxim Integrated, 2002

[53] A. Meyna: *Zuverlässigkeitsbewertung zukunftsorientierter Technologien.* Vieweg, Braunschweig, 1994

[54] Mikrocontroller.net: *AVR-Tutorial. Watchdog.* http://www.mikrocontroller.net/articles/AVR-Tutorial:_Watchdog (zuletzt besucht 03.Juni 2013), 2010

[55] The MTL Instruments Group plc: *An introduction to Functional Safety and IEC 61508.* http://www.mtl-inst.com/images/uploads/datasheets/App_Notes/AN9025.pdf (zuletzt besucht 03.Juni 2013), 2002

[56] MVM Paks Nuclear Power Plant Ltd: *The nuclear reactor.* http://paksnuclearpowerplant.com/the-nuclear-reactor (zuletzt besucht 03.Juni 2013)

[57] Naval Research Lab: *Clementine Mission 1994.* http://www.nrl.navy.mil/clementine2/ (zuletzt besucht 04.Juni 2013), U.S. Naval Research Laboratory

[58] C. Özdogan: *Priority Scheduling.* Ceng 328 Operating Systems Lecture Notes, http://siber.cankaya.edu.tr/OperatingSystems/ceng328/node124.html (zuletzt besucht 04.Juni 2013), Department of Materials Science and Engineering, Çankaya University, Ankara, 2011

[59] I. Parry: *SIL Calculations Easy or Difficult.* http://www.61508.org/wp-content/uploads/2013/03/IMC-Teesside-2012-SIL-Calculations-Easy-or-Difficult.pdf (zuletzt besucht 03.Juni 2013), 2011

[60] Pepperl+Fuchs GmbH: *SIL-Handbuch. Funktionale Sicherheit.* http://www.pepperl-fuchs.de/cgi-bin/db/doci.pl?ShowDocByDocNo=TDOCT-0713&LanId=GER (zuletzt besucht 03.Juni 2013), 2007

[61] W. Plaßmann, D. Schulz: *Handbuch Elektrotechnik. Grundlagen und Anwendungen für Elektrotechniker.* Vieweg + Teubner, 5.Auflage, Wiesbaden, 2009

[62] M. Plümicke: *Tasks und Threads.* http://www.ba-horb.de/~pl/BS_Skript/node69.html (zuletzt besucht 04.Juni 2013), 1999

[63] W. E. Proebster: *Rechnernetze. Technik, Protokolle, Systeme, Anwendungen.* Oldenbourg Verlag, 2.Auflage, München/Wien/Oldenbourg, 2002

[64] M. Rausand, A. Høyland: *System Reliability Theory. Models, Statistical Methods, and Applications.* Wiley & Sons, 2.Auflage, Hoboken, NJ, 2004

[65] G. Reeves: *What really happened on Mars.* http://research.microsoft.com/en-us/um/people/mbj/Mars_Pathfinder/Authoritative_Account.html (zuletzt besucht 04.Juni 2013), 1997

[66] RN-Wissen: *Watchdog.* http://www.rn-wissen.de/index.php/Watchdog (zuletzt besucht 03.Juni 2013), Roboternetz, 2011

[67] Rockwell Automation: *Einführung zur Sicherheitszuhaltung.* http://www.ab.com/de/epub/catalogs/3377539/5866177/3377559/6388299/3378094/6537414/Einf-hrung.html (zuletzt besucht 03.Juni 2013)

[68] D. Sautter, H. Weinerth: *Lexikon Elektronik und Mikroelektronik.* VDI-Verlag, 2. Auflage, Düsseldorf, 1993

[69] G.-H. Schildt, D. Kahn: *Fundamental principles of safety related system design.* In: Electrical and Computer Engineering Series - Recent Advances in Intelligent Systems and Signal Processing, World Scientific Engineering Academy Society Press, Corfu, 2003, S. 72-77

[70] G.-H. Schildt: *Fail-safe comparators in fault-tolerant systems.* Vortrag: Simposio de Ingenieria electrica. In: Proc. SIE2003, Santa Clara, Kuba, 2003

[71] G.-H. Schildt, *Grundlagen für Vergleicher mit Sicherheitsverantwortung.* In: Siemens Forschungs- und Entwicklungsberichte, Band 9, Springer, Berlin, 1980

[72] G.-H. Schildt, W. Kastner: *Prozeßautomatisierung.* Springer, Wien, 1998

[73] S. Schneider: *Die Dekomposition der ISO 26262.* http://www.sebastian-schneider.eu/cms/iso-26262/die-dekomposition-der-iso-26262 (zuletzt besucht 20.Juni 2013), 2011

[74] O. Schulz: *Entwurf und Analyse sicherheitsrelevanter Kommunikationsarchitekturen*: Dissertation, Universität Bremen, Bremen, 2011

[75] SecuPedia: *Arbeitsstrom/Ruhestrom.* http://www.secupedia.info/wiki/Arbeitsstrom_/_Ruhestrom (zuletzt besucht 03.Juni 2013), SecuMedia Verlags GmbH

[76] G. Socher: *A Hardware watchdog and shutdown button.* http://www.linuxfocus.org/English/July2002/article239.shtml (zuletzt besucht 03.Juni 2013), LinuxFocus, 2002

[77] T. Tietjen, D. H. Müller: F*MEA-Praxis. Das Komplettpaket für Training und Anwendung.* Carl Hanser Verlag, 2.Auflage, München/Wien, 2003

[78] Society of Automotive Engineer (SAE) International: *ARP926 Fault/Failure Analysis Procedure.* Aerospace Recommended Practice, 1967

[79] A. Steimel: *Elektrische Triebfahrzeuge und ihre Energieversorgung. Grundlagen und Praxis.* Oldenbourg Industrieverlag, 2.Auflage, München, 2006

[80] F. Stajano, R. Anderson: *The Grenade Timer. Fortifying the Watchdog Timer Against Malicious Mobile Code.* In: Proceedings of 7th International Workshop on Mobile Multimedia Communications (MoMuC 2000), Waseda, Tokyo, Japan, October 2000

[81] M. Stein: *Mars Pathfinder. Die Generalprobe.* http://www.raumfahrer.net/raumfahrt/marsrover2003/pathfinder.shtml (zuletzt besucht 04.Juni 2013), Raumfahrer Net e.V., 2003

[82] S. Tetlow: *Formal risk identification in professional SCUBA (FRIPS).* HSE Books, Norwich, 2006

[83] VDMA (Verband Deutscher Maschinen und Anlagenbau e.V.): *Leitfaden Funktionale Sicherheit Safety Integrity Level.* http://www.vdma.org (zuletzt besucht 03.Juni 2013), 2008

[84] M. Werdich: *FMEA. Einführung und Moderation.* Vieweg + Teubner, Wiesbaden, 2011/2012

[85] O. C. Winne, M. Hafner: *Leitfaden für die Norm IEC 61508.* http://www.elektronikpraxis.vogel.de/themen/embeddedsoftwareengineering/management/articles/185862/ (zuletzt besucht 04.Juni 2013), Vogel Business Media, 2009

[86] P. Wratil, M. Kieviet: *Sicherheitstechnik für Komponenten und Systeme.* VDE-Verlag, Berlin, 2010

[87] P. Wratil, M. Kieviet, W. Röhrs: *Sicherheitstechnik für Maschinen und Anlagen. Mechanische Einheiten, elektronische Systeme und sicherheitsgerichtete Programmierung.* VDE-Verlag, Berlin, 2010