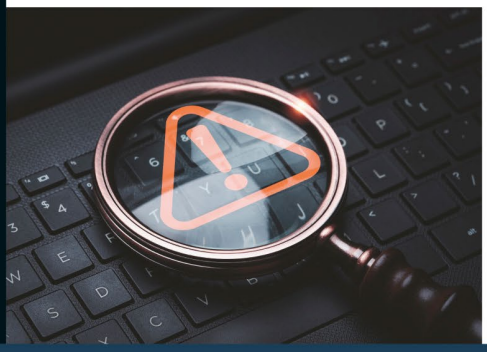
Derk Rembold



Safety Engineering

Das Praxisbuch für funktionale Sicherheit

- Methoden für sichere und robuste Softwareentwicklung
- Betriebssicherheit herstellen und Fehler verstehen
- Gefahrenanalysen, Risikographen, Fehlerbaumanalysen, Zuverlässigkeitsblockdiagramme, Markov-Decision-Prozesse

Rheinwerk Computing

Kapitel 1
Einführung

Das Buch befasst sich mit der Sicherheit von Systemen und Geräten. Im Englischen wird dieses Thema *Safety Engineering* genannt, und zwar abgegrenzt von der Sicherheit vor Angriffen, die von außen auf Systeme und Geräte wirken. Im Englischen wird dafür der Begriff *Security* verwendet. In diesem Buch stehen Gefahren, die selbst von Systemen und Geräten ausgehen, im Vordergrund. Die englische Bezeichnung dafür ist *Safety*. Abbildung 1.1 zeigt die Abgrenzung zwischen *Security* und *Safety*. Bezüglich einer detaillierteren Unterscheidung wird auf das Buch [1] hingewiesen.

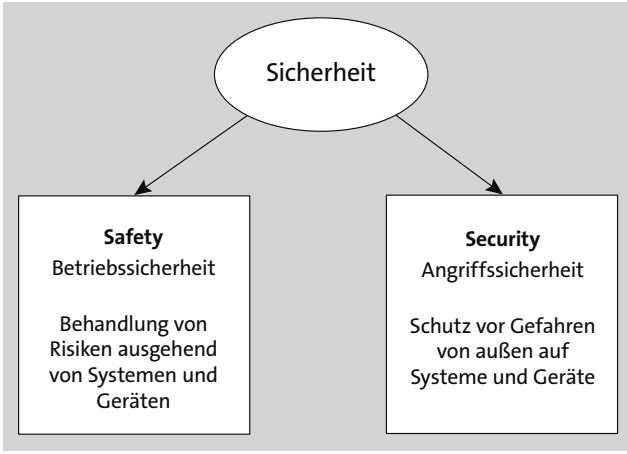


Abbildung 1.1 Abgrenzung zwischen Security und Safety (aus [1])

Gefahren und Risiken, die aus dem System oder Gerät hervorgehen, werden über Methoden bewertet. In den ersten Kapiteln werden qualitative Ansätze und in den folgenden Kapiteln quantitative (also rechnerische) Ansätze vorgestellt. In jedem Kapitel sind ein oder zwei Fallbeispiele beschrieben, um zu zeigen, welche Vorfälle zu teilweise katastrophalen Unglücken führten. So werden die vorgestellten Methoden an den Fallbeispielen durchgespielt bzw. diskutiert.

Viele der Methoden in diesem Buch sind bereits in Normen zusammengestellt. Für eine Übersicht werden in Kapitel 3 die wichtigsten Normen aufgezählt, und deren Inhalt wird kurz beschrieben. Auch Normen, die keine Anwendung mehr finden, sind darunter. Die Benennung der alten Normen ist wegen der Historie interessant, damit die Entstehung der aktuellen Normen aufgezeigt wird.

Da dieses Buch im Rahmen einer Vorlesung in der Informatik entstanden ist, enthält die Softwareentwicklung ein entsprechendes Gewicht, ihr ist ein ganzes Kapitel gewidmet. Software ist heutzutage ein fester Bestandteil von Systemen und Geräten. Im Gegensatz zu Hardwarefehlern lassen sich Softwarefehler (vermeintlich) sehr leicht und kostengünstig beheben. Bei neueren Systemen oder Geräten kann die Software mit Orchestrierungswerkzeugen (z. B. *Kubernetes*) direkt von der Entwicklungsabteilung auf eine technische Anlage des Kunden aufgespielt werden. Dies verleitet dazu, Sicherheitsbetrachtungen bei der Software zu vernachlässigen, da die Behebung der Probleme unter Umständen leicht sein kann. Diese Leichtigkeit ist aber ein Trugschluss, denn der Entwickler von Software hat nicht in allen Fällen Zugang zur technischen Anlage des Kunden. Debug-Daten zur Fehleranalyse, notwendig zur Fehlerbehebung, sind oft umständlich zu erhalten.

Unglücke aus der Vergangenheit sind aufgrund von Fehlern oder Ausfällen unterschiedlicher Art entstanden. So gibt es z. B. unterschiedliche technische Fehler, hervorgerufen durch Hardwarefehler, Softwarefehler oder menschliche Fehler. Bei laufenden Systemen und Geräten entstehen immer Fehler und Ausfälle bei den Komponenten. Dabei gibt es ungefährliche und gefährliche Fehler und Ausfälle. Des Weiteren kann bei der Detektierbarkeit von Fehlern und Ausfällen unterschieden werden. In den Fallbeispielen dieses Buchs wird beschrieben, wie auftretende Fehler und Ausfälle nicht bemerkt wurden und so zu einem katastrophalen Unglück führten.

Vor allem bei der Hardware, aber auch in vielen Fällen bei der Software, folgen Fehler und Ausfälle statistischen Mustern. Deswegen können statistische Methoden eingesetzt werden, um die Häufigkeiten und ihre zeitlichen Verläufe zu beschreiben. Durch den Einsatz der Wahrscheinlichkeitstheorie können Vorhersagen bezüglich der Zuverlässigkeit und der Verfügbarkeit von Systemen und Geräten getroffen werden.

Bereits in der zweiten Hälfte des 20. Jahrhunderts wurden Verfahren entwickelt, um Gefahren zu ermitteln, die von Systemen und Geräten ausgehen. Das Studium der Fallbeispiele aus der Vergangenheit, aber auch das Sammeln eigener Erfahrungen hilft bei der Entwicklung von Systemen und Geräten mit hohen sicherheitstechnischen Anforderungen. So gibt es Methoden, die sowohl am Anfang als auch am Ende der Entwicklungsphase eingesetzt werden. Idealerweise liefern Methoden der unterschiedlichen Projektphasen auch ähnliche Ergebnisse. Es ist dabei wichtig, dass die Methoden von Experten angewendet werden, die Erfahrung darin haben, sie richtig einzusetzen. Dennoch ist es von Vorteil, junge Entwickler und Entwicklerinnen mit ihren Ideen und kreativen Denkansätzen einzubeziehen.

Ergebnisse der Gefahrenanalyse aus teamorientierten Methoden (qualitativ) und mathematischen Beschreibungen von Systemen und Geräten (quantitativ) mit Methoden der Statistik können miteinander kombiniert werden. Ziel ist unter anderem

die Ermittlung von Kenngrößen, die eine Aussage über den sicherheitstechnischen Zustand liefern können. Kenngrößen lassen sich bestimmen, indem Systeme und Geräte über längere Zeit beobachtet werden. Publikationen und Normen helfen, aus Daten und über Berechnungsvorschriften Kenngrößen zu ermitteln.

Es gibt teamorientierte Methoden zur Gefahrenanalyse, womit Ereignisse, die zum unerwünschten Ausfall führen können, herausgearbeitet werden können. So sind Ereignisse z. B. Bedienfehler, Verschleißausfälle etc. In der Regel führt aber nicht ein einziger Fehler oder Ausfall zum Unglück, sondern eine Kombination daraus. Es gibt Methoden, z. B. die Fehlerbaumanalyse, mit der sich diese Ereignisse mit Elementen aus der *booleschen Algebra* kombinieren lassen. Dieses Verfahren kann dazu verwendet werden, die Ergebnisse aus der teamorientierten Gefahrenanalyse zu validieren. Aber auch Berechnungen zur Wahrscheinlichkeitsbestimmung bei Kenntnis der Kenngrößen können angewendet werden.

Weitere Methoden zur Untersuchung der Gefahren ist die Anwendung von Methoden wie *Risikographen* und *Layer of Protection Analysis*. Es sind Methoden, die auf teamorientierten Gefahrenanalysen aufsetzen. So liefert die erste Methode eine Einschätzung für weitergehende Maßnahmen zur Risikominimierung von Gefahren. Mögliche Maßnahmen können in Normen vorgeschlagen sein, was den Vorteil hat, dass der *Safety Engineer* sich darauf beziehen kann. Die zweite oben genannte Methode ist eine Erweiterung der Methode aus der Gefahrenanalyse. So werden während der Entwicklung Maßnahmen zur Risikominimierung aufgezählt und dokumentiert. Über Berechnungen lässt sich durch den Einsatz von Schätzwerten eine Wahrscheinlichkeit für das Eintreten von Gefahrsituationen bestimmen. Das Ergebnis der Analyse kann zur Bestimmung der notwendigen Maßnahmen herangezogen werden. Normen stehen dafür als Entscheidungsgrundlage zu Verfügung.

Normen schlagen in vielen Fällen die Redundanz als technische Maßnahme zur Gefahrenreduzierung vor. So gibt es Zuverlässigkeitsdiagramme, mit denen sich Systeme und Geräte mit ihren Sicherheitssystemen zur Risikoreduzierung grafisch beschreiben lassen. Bei Einsatz der Zuverlässigkeitskenngröße der einzelnen Komponenten kann auch die Wahrscheinlichkeit des Ausfalls für ein ganzes System oder Gerät bestimmt werden. Der Einsatz dieser Diagramme geht also über die Dokumentation von System und Gerät und die Darstellung von konstruktiven Maßnahmen zur Gefahrenreduzierung hinaus.

Nachteilig bei Modellen mit Fehlerbäumen und Zuverlässigkeitsdiagrammen ist, dass der Verlauf der Zeit nicht ausreichend berücksichtigt wird. Es ist die Regel, dass ein Fahrzeug alle zwei Jahre zur Inspektion in die Werkstatt gebracht wird. Das Ziel ist es, mögliche Fehler und Ausfälle zu erkennen und beheben, sodass ein weiterer Fehler oder Ausfall in den folgenden zwei Jahren vermieden werden kann. Die beiden oben genannten Methoden können zwar diese rudimentären Instandsetzungsmaßnahmen abbilden, aber der *Markov*-Prozess bietet eine wesentlich elegantere Methode, die die

Zeit berücksichtigt. Sie zeichnet sich durch höhere Flexibilität in der Modellierung aus. Mathematische Methoden können eingesetzt werden, um zeitliche Simulationen durchzuführen, und Wahrscheinlichkeiten für Ausfälle unter Berücksichtigung von Betriebsarten von Systemen und Geräten (Betrieb, Instandsetzung, Prüfung) zu ermitteln.

Eine Erweiterung des *Markov*-Prozesses ist der *Markov Decision*-Prozess. Damit können verschiedene Szenarien von verschiedenen ineinandergreifenden Markov-Modellen modelliert werden. So kann der Betrieb des Systems mit dem ersten Markov-Modell und die Wartung mit einem zweiten Markov-Modell beschrieben werden. Das Überführen von einem Modell zu einem anderem erfolgt über Aktionen, die der Betreiber einleiten kann. Belohnungen können Aktionen zugeordnet werden, genannt Strategie, um die die Gesamtbelohnung zu optimieren.

Zuverlässigkeitsdiagramme und *Markov-Diagramme* sind Methoden, um Systeme und Geräte, die regelmäßig geprüft werden, zu modellieren. Normen stellen dafür Formeln für die Zuverlässigkeitsberechnung zur Verfügung. Die Formeln sind zwar leicht anzuwenden, aber die Herleitung ist nicht immer offensichtlich. So sollen mithilfe von Zuverlässigkeitsdiagrammen und einem Modell, bei dem das System in regelmäßigen Prüfintervallen instandgesetzt wird, die Formeln aus den Normen hergeleitet werden. *Markov*-Prozesse sind zwar hervorragend für die Beschreibung von Systemen mit Prüfintervallen geeignet, dennoch kann die analytische Berechnung kompliziert sein. Bei Betrachtung von langen Zeiträumen kann der zeitliche Verlauf mit Grenzwerten vereinfacht werden.

Schlussendlich soll in diesem Buch noch eine letzte Methode zur Modellierung von Fehlern und Ausfällen beschrieben werden. Das *Binary Decision Diagram* ist eine der letzten wissenschaftlich relevanten Datenstrukturen aus der Informatik der letzten Jahrzehnte. Diese wurde entwickelt, um binäre Funktionen mit einer großen Anzahl von Eingängen auf eine übersichtliche Struktur zu reduzieren. Bei Einsatz der Wahrscheinlichkeitstheorie kann die *Binary Decision Diagram*-Methode auch für die Berechnung von Zuverlässigkeits- und Verfügbarkeitswahrscheinlichkeiten von Systemen und Geräten eingesetzt werden.

Kapitel 2
Der Weg durch das Buch

In diesem Kapitel wird der Weg durch das Buch beschrieben. Es besteht im Wesentlichen aus drei Teilen. Der erste (siehe auch Abbildung 2.1) enthält die einleitenden Kapitel mit der Beschreibung von Normen, der Definition von Begriffen und der Benennung von Methoden zur Identifikation von Gefahren. Im zweiten Teil, siehe Abbildung 2.2, werden qualitative Methoden zur Modellierung vorgestellt, um Gefahren mit dem Ziel zu beschreiben, Kenngrößen zur Beschreibung der Sicherheitsanforderungen zu ermitteln. Manche dieser Methoden sind auch für die halbquantitative bzw. quantitative Analyse einsetzbar. Abbildung 2.3 zeigt die Methoden des letzten Teils. Dort werden ausschließlich quantitative Methoden zur Analyse von Systemen und Geräten bezüglich der Sicherheit beschrieben.

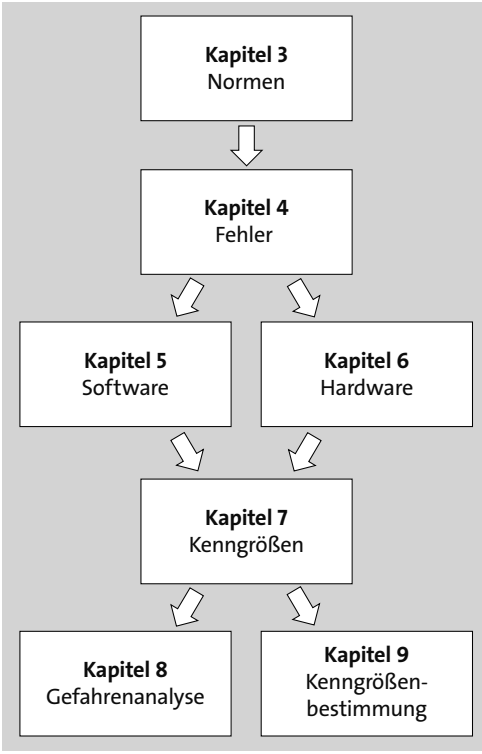


Abbildung 2.1 Einleitende Kapitel

2.1 Einleitende Kapitel

Wenn der angehende *Safety Engineer* durch einen Kundenwunsch die Aufgabe erhält, ein System oder Gerät sicher zu machen, wird er sich ohne Hilfe schwertun, den Anfang zu finden. Kapitel 3 soll ihm helfen, zunächst einen groben Überblick über Normen zu bekommen. So sind nämlich geltende Gesetze bei zu entwickelnden Systemen und Geräten einzuhalten, die vom Gesetzgeber nicht notwendigerweise kontrolliert werden. Wenn es aber zu einem Zwischenfall kommt, steht der Hersteller des Systems oder Geräts in der Beweispflicht. Er muss nachweisen, dass das System oder Gerät nach aktuellen Sicherheitsstandards entwickelt wurde. Es ist dann vorprogrammiert, dass es innerhalb der Firma zu gegenseitigen Schuldzuweisungen kommt, was das Betriebsklima stört. Zur Veranschaulichung wird das Fallbeispiel der Ölförderplattform *Deep Horizon* vorgestellt, die an der Ölkatastrophe im Golf vom Mexiko maßgeblich beteiligt war. Es soll anhand des Fallbeispiels klargemacht werden, dass nicht nur der Safety Engineer Verantwortung trägt, sondern auch Betriebspersonal, Manager und Geschäftsführung. Kapitel 3 beschreibt als Einstieg die Norm *IEC-61508*, die dem Safety Engineer als Grundlage dient. Es wird unter anderem in diesem Kapitel eine Tabelle aus der Norm übernommen, aus der ein Kennwert (das sogenannte *SIL*) entnommen wird, der die Anforderungen in Abhängigkeit von der Anzahl der auftretenden Fehler für ein Sicherheitssystem beschreibt. Die Tabelle ist zentral, und ich werde mich in mehreren Kapiteln immer wieder auf diese Tabelle beziehen.

Weiterführende Normen basieren auf der Norm *IEC-61508*. Sie übernehmen zum Teil Ausschnitte aus ihr, beziehen diese aber dann wieder auf ihre Teilgebiete, wie z. B. Medizintechnik, Maschinenbau, Prozesstechnik etc. Ihnen soll in diesem Kapitel klargemacht werden, dass Sicherheitssysteme von Systemen und Geräten nicht nur in der Entwicklungsphase konstruiert werden, sondern dass Sicherheit in den ganzen Lebenszyklus – also Planungsphase, Entwicklungsphase, Produktion, Betrieb und Außerbetriebsetzung – einbezogen werden muss. So umfasst das Thema *Safety Engineering* deutlich mehr als nur die Entwicklung von robuster Software und Hardware.

Auch andere Normen, z. B. *ISO-26262* (Automobilindustrie), *DIN-19250* (ausgediente Norm) etc., spielen beim *Safety Engineering* eine Rolle und werden in diesem Kapitel vorgestellt. Es wird nicht zu sehr auf die Details eingegangen. Es soll reichen, dass Sie einen groben Überblick erhalten, der Ihnen bei weiteren Recherchen wertvolle Hilfe leisten kann.

Kapitel 4 geht sehr grundlegend auf die Beschreibung von Fehlern, Ausfällen und Raten ein. Hier sind nur wenige mathematische Kenntnisse für die Berechnung von Kenngrößen zum Verständnis notwendig. Zunächst werden die Begriffe Sicherheit, Zuverlässigkeit, Verfügbarkeit und Risiko definiert, da sie in diesem Buch immer wieder genutzt werden. In dem Kapitel werden zwei historische Fallbeispiele beschrie-

ben, um Fehler- und Ausfallursachen aufzuzeigen, die entweder vom Betriebspersonal ausgehen oder eine technische Ursache haben. So bestand z. B. beim *Seveso-Unglück* das Problem darin, dass die Dauer einer chemischen Reaktion durch ein überarbeitetes Personal falsch eingeschätzt wurde. Bei Verwendung einer Zeittoleranz, also dem Warten über den tatsächlichen Zeitbedarf hinaus, hätte das Unglück vermieden werden können. Das zweite Fallbeispiel zeigt, dass Instandsetzungsmaßnahmen Fehler oder Ausfälle hervorrufen, wenn defekte Geräte in ein System eingebaut werden, so wie es bei einem Metrounglück in New York geschah.

Es wird unterschieden zwischen Ausfall, was ein Ereignis ist, und Fehler, was ein Zustand ist. Ausfälle und Fehler können sicherheitsrelevant sein, müssen aber nicht. Deswegen werden Ausfälle bzw. Fehler gruppiert. In vielen Fällen können sie durch Voranalysen identifiziert und somit bei der Entwicklung des Systems berücksichtigt werden. In anderen Fällen sind mögliche Fehler und Ausfälle unbekannt. Wenn sie dabei sicherheitsrelevant sind, kann der Fehler bzw. Ausfall zu einem unerwünschten Ereignis führen, was eine Katastrophe für Mensch und Umwelt bedeuten kann.

Aus den gruppierten Fehlern lassen sich Kenngrößen ermitteln, wie zum Beispiel der *Diagnostic Coverage*-Faktor oder die *Safe Failure Fraction*. Aus denen lassen sich wiederum Kenngrößen für die Sicherheitsanforderungen ableiten. Diese dienen als Richtlinien bei der Entwicklung von Systemen und Geräten. Um eine Verbesserung der Zuverlässigkeit oder Verfügbarkeit etc. zu erreichen, wird oftmals Redundanz eingesetzt. Das kann bedeuten, dass ein Gerät oder eine Komponente mehrfach ausgelegt wird. Bei einem Ausfall eines Geräts, oder auch einer Komponente, übernimmt ein zweites, das sich z. B. im Stand-by befindet. Hier wird ein Beispiel aus dem Bereich der Mikrosystemtechnik gezeigt, bei dem ein *ASIC* aus mehreren redundanten Komponenten aufgebaut wird.

Redundanz bringt aber eine neue Klasse von Fehlern hervor. Denn ein Ereignis kann eine Auswirkung auf alle redundanten Komponenten vom gleichen Typ haben. Diese Klasse wird *Fehler mit gemeinsamer Ursache* genannt.

Jeder Softwareentwickler weiß (oder sollte es wissen), dass Software ohne Fehler kaum herzustellen ist. Dies gilt insbesondere dann, wenn sie einen hohen Grad an Komplexität erreicht. Das Fallbeispiel in Kapitel 5 erzählt von einem Flug, geführt durch einen Autopiloten, mit einem Problem der Benutzerschnittstelle der Software. Durch eine Falscheingabe einer Anweisung führte er das Flugzeug in ein fatales Unglück. Die Ursache konnte auf ein Missverständnis zwischen den Anforderungen, beschrieben in den Entwicklungsdokumenten des Autopiloten, und dem Softwareentwickler zurückgeführt werden. Um Fehler bei der Softwareentwicklung einzudämmen, entwickelten sich im Laufe der Zeit Spielregeln zwischen den Softwareentwicklern, Architekten, Testern etc. Diese Spielregeln, vorgestellt in diesem Buch, werden als Softwareentwicklungsprozess bezeichnet.

In diesen Prozess weiß der Softwareentwickler genau, wo sein Quellcode abgelegt wird und wie andere Entwickler mit ihm umgehen . Viele Entwickler, die sich an der System- und Geräteentwicklung beteiligen, kommen aber nicht aus dem Softwarebereich. So kann es sein, dass ihnen nicht bewusst ist, welcher Aufwand dahintersteckt, wenn eine Infrastruktur für die Softwareentwicklung aufgebaut werden soll. Kapitel 5 soll deswegen allen Entwicklern (Software und Hardware) helfen, die Prozesshintergründe besser zu verstehen. Für sicherheitsrelevante Software ist auch dieser Softwareentwicklungsprozess fundamental wichtig. Ein Merkmal ist unter anderem, dass z. B. die Norm *IEC-61508* dem Entwickler Checklisten mitgibt, um Entscheidungen zu erleichtern, bestimmte Programmiermethoden bzw. Prozessschritte in den Softwareentwicklungsprozess aufzunehmen.

Kapitel 5 beginnt mit der Erklärung der Projektmanagementmethode *V-Modell* (es handelt sich dabei nicht um das erweiterte *V-Modell XT*). Angefangen wird hier mit den Sicherheitsanforderungen und der Entwicklung der Architektur. Danach geht der Prozess in die Projektabschnitte Entwurf, Test und Integration über. Gegen Ende erfolgt die Abnahme. In jedem dieser Abschnitte müssen sicherheitsrelevante Dokumente erstellt und strukturiert abgelegt werden. Dies ist unter anderem wichtig, damit bei einer Weiterentwicklung des Systems oder im Fall einer Beweisumkehr die Dokumente verfügbar sind. Die Norm *IEC-61508* schlägt in den einzelnen Phasen Maßnahmen (bzw. Checklisten) vor, die für bestimmte Sicherheitsanforderungen (auch genannt Sicherheitsintegritätslevels) empfohlen werden. Beispiele sind Coderichtlinien, lineare und modul-orientierte Programmierung etc. Werkzeuge können durch automatische Erzeugung von Programmiervorschlägen unterstützend wirken. Analysatoren untersuchen den Quellcode und finden problematische Softwarekonstruktionen.

Kooperative Werkzeuge sind weitere wichtige Bausteine im Softwareentwicklungsprozess. Es gibt Werkzeuge, die das Review von Quellcode zwischen Softwareentwicklern erleichtern. Weitere Werkzeuge dokumentieren Probleme im Quellcode und ordnen sie einem Verantwortlichen zu. Insbesondere in der Testphase des Softwareentwicklungsprozesses tritt eine sehr große Zahl von Fehlern auf. Die Koordination kann dann nur durch ein Ticketmanagementsystem erfolgen. Weitere Werkzeuge, wie ein Konfigurationsmanagementsystem, hilft dem Entwickler, alte Softwarezustände und Entwicklungsumgebungen wiederherzustellen.

Das Ticketmanagementsystem und der Entwicklungsprozess gehen vor allem in Testphase und Integrationsphase Hand in Hand. Bei Software mit hoher Komplexität muss vor allem die Integration gut durchdacht und vorgeplant werden. Dabei sind die sicherheitsrelevanten Komponenten bei der Integration hervorzuheben. Am Ende des Kapitels wird der Softwareentwicklungsprozess im Detail erklärt. Es wird dargestellt, welche Werkzeuge bei der Entwicklung eingesetzt werden sollten. Themen wie *Continuous Integration and Development* sowie *DevOps* werden angespro-

chen. Abgeschlossen wird das Kapitel mit einem vereinfachten Bauplan für eine Infrastruktur (Server für das Ticketmanagementsystem, Repository, Backup und ihre Interaktion) eines Softwareentwicklungsprozesses.

Systeme und Geräte sind geprägt durch ihren Hardwareanteil. In Kapitel 6 gehe ich deswegen, ähnlich wie in Kapitel 5 für Software, auf den Prozessablauf für die Hardware ein. So beginnt der Prozess mit den Anforderungen, geht weiter zu den Abschnitten Architektur und Entwicklung und schließt mit Test, Integration und Abnahme ab. Dieses sind Schritte, die in einem Hardwareentwicklungsprozess beschrieben werden. Dazu verwendet das Kapitel die vereinfachte *V-Modell*-Methode, die aber nicht mit der weiterführenden *V-Modell-XT*-Methode zu verwechseln ist. Aus den Ablauf des Hardwareentwicklungsprozesses gehen dann Dokumente hervor, die für die folgenden Schritte Grundlagen sind. Auch bei einem Audit (wegen der Beweisumkehr) ist es wichtig, die Dokumente zugreifbar zu haben. Bei der Entwicklung von Hardware kann sehr häufig auf Softwarewerkzeuge und -sprachen zurückgegriffen werden. So gibt es Werkzeuge auf verschiedenen Detaillierungsebenen. Zum Beispiel wird in diesem Kapitel kurz auf die Hardwarebeschreibungssprachen für die Entwicklung von *ASIC* auf sehr niedriger Ebene eingegangen. Auf mittlerer Entwicklungsebene können Geräte zur Steuerung von Systemen und Geräten gekauft werden. Die Möglichkeiten zur Konfiguration der Geräte sind nahezu unbegrenzt. Auf abstrakter Ebene ist Hardware und sind technische Prozesse oftmals mit *Petri-Netzen* modellierbar – insbesondere dann, wenn Ereignisse wesentliche Eigenschaften des Systems oder Geräts sind. Auf Petri-Netze wird in diesem Buch in mehreren Kapiteln eingegangen, denn sie eignen sich für die Modellierung von Zustandswechseln. Ein Zustandswechsel kann der Übergang eines Systems vom Betriebszustand in den Instandsetzungszustand sein.

Das Fallbeispiel in diesem Kapitel nimmt das Unglück während des Starts der *Challenger*-Raumfähre auf. Hier bestand wahrscheinlich die Unglücksursache unter anderem in einem spröden Dichtungsring der Feststoffrakete. Zur Verbesserung der Sicherheit von Systemen werden hier einfache Beispiele mit schaltungstechnischen Methoden gezeigt. Sie beschränken sich auf die Funktionsweise von Schaltern oder Relais und Abfragen der Zustände durch Steuerrechner. Tatsächlich kann die Sicherheitstechnik viele Bücher füllen. Eine Rolle bei der Bestimmung der Sicherheitsanforderungen spielt die Kenntnis der eingesetzten Geräte und Komponenten innerhalb des Systems. Geräte und Komponenten werden deshalb in Typen klassifiziert, abhängig davon, ob der Hersteller selbst die Fehler und Zustände kennt und ob diese durch Daten nachgewiesen werden können. Die *Safe Failure Fraction*-Kenngröße und der Gerätetyp sind dann eine Möglichkeit, die Kenngröße für die Sicherheitsanforderung herzuleiten. Wird die Anforderung des Sicherheitssystems nicht erreicht, lässt sie sich durch Einsatz von Redundanz verbessern.

In diesem Kapitel lernen Sie, wie die Kenngröße der Sicherheitsanforderungen eines gesamten Systems, bestehend aus Teilsystemen, Geräten und Komponenten, durch einfache Methoden bestimmt werden können. Es wird eine weitere Möglichkeit gezeigt, bei der die Kenngröße für die Sicherheitsanforderung mit Daten und Expertenmeinungen ermittelt wird.

Alle Komponenten fallen nach einer gewissen Zeit aus. Wann aber diese ausfallen, kann einem Zufallsprozess unterliegen. So kann der Komponenten eine Wahrscheinlichkeit zugeordnet werden. Sie verändert sich nach der Zeit, sodass die Wahrscheinlichkeit eine Funktion abhängig von der Zeit ist. Die Wahrscheinlichkeit, dass ein neues Gerät funktioniert, sollte hoch sein. Aber sie nimmt ab, je länger das Gerät in Betrieb ist. In Kapitel 7 geht es um die mathematische Bestimmung der Zuverlässigkeits- bzw. Ausfallwahrscheinlichkeit.

Dafür sind Kenntnisse aus der Wahrscheinlichkeitstheorie notwendig, die in dem ersten Teilkapiteln als Grundlagen wiederholt werden. Mit dieser Theorie kann die mathematische Zuverlässigkeits- und Ausfallwahrscheinlichkeit definiert werden. Zum Teil basieren die Formeln auf Dichtefunktionen, die die Häufigkeit von Ausfällen zu einen Zeitpunkt bzw. Abschnitt beschreiben. Da die Ausfallcharakteristik nicht bei jedem Gerät gleich ist, unterliegt sie unterschiedlichen Dichtefunktionen. Es werden in diesem Kapitel diejenigen vorgestellt, die bei Zuverlässigkeitsberechnungen häufig eingesetzt werden.

Um sich ein Bild über die Charakteristik bezüglich der Zuverlässigkeit und Verfügbarkeit machen zu können, sind nicht immer mathematische Vorkenntnisse notwendig. Zur Vereinfachung können zusammenfassende Kenngrößen verwendet werden, wie z. B. die mittlere Ausfallzeit. Die Kenngrößen werden über mathematische Methoden hergeleitet. Wichtige Parameter bei Zuverlässigkeitsberechnungen sind Ausfallraten und Ausfallhäufigkeiten (deren Verteilung eine Dichtefunktion ist). Diese können sich zwar nach einiger Zeit ändern, aber zur Vereinfachung wird oft angenommen, dass sie konstant sind. So wird gezeigt, wie durch diese Vereinfachungen Zuverlässigkeitsfunktionen durch Exponentialfunktionen beschrieben werden können.

Das in diesem Kapitel aufgeführte Fallbeispiel beschreibt die Reihe der Abstürze von *Starfighter*-Kampfflugzeugen, deren Ursache in vielen Fällen der frühzeitige Ausfall von Komponenten war. Es wird unter anderem beschrieben, wie ein Reengineering der Komponenten die Ausfallraten und die Zuverlässigkeit des Flugzeugs verbesserte.

Systeme oder Geräte mit Sicherheitsfunktionen, die den Konsequenzen der Fehler und Ausfälle entgegenwirken, werden aufgeteilt in Systeme mit niedriger und hoher Anforderungsrate. Die Anforderungsrate beschreibt die Häufigkeit des Einsatzes der Sicherheitsfunktionen des Systems innerhalb eines Jahres. So werden gegen Ende des Kapitels Formeln hergeleitet, die die Wahrscheinlichkeiten ausdrücken, dass innerhalb eines Zeitintervalls das Sicherheitssystem ausfällt. Mit dieser Wahrscheinlichkeit

kann der zentralen Tabelle aus Kapitel 3 entnommen werden, welche Kenngröße der Sicherheitsanforderung (*SIL*) sich für das zu betrachtende System oder Gerät ergibt.

In Kapitel 7 wird gezeigt, wie die Zuverlässigkeit von einfachen Systemen mit mathematischen Funktionen beschrieben werden können. Die Realität zeigt aber, dass die Herleitung schwierig sein kann. Oftmals sind Daten, die die Fehler beschreiben, nicht vorhanden. Menschliche Faktoren haben dabei einen großen Einfluss, sodass eine quantitative Beschreibung nur durch Abschätzung möglich ist. Bei Systemen oder Geräten ohne Erfahrungswerte müssen die Gefahren, die aus ihnen hervorgehen, zunächst identifiziert werden, bevor eine mathematische Beschreibung möglich ist. Kapitel 8 befasst sich deshalb mit Methoden, die Fehlermöglichkeiten und Gefahren identifizieren. Im Fallbeispiel wird das Unglück von *Bhopal* beschrieben. Primär waren menschliche Faktoren die Ursache des Unglücks. Technische Faktoren leisteten aber auch ihren Beitrag, da wichtige Geräte ausgefallen bzw. abgebaut wurden, die das Unglück hätten verhindern können. Eine der Methoden zur Analyse der Fehlermöglichkeiten ist die *Failure Mode Effect Analysis*, sie wird am Anfang des Kapitels beschrieben. Die einzelnen Schritte werden erklärt, und als Ergebnis entsteht ein Dokument mit einer Reihe von Fehlermöglichkeiten, die das Entwicklungsteam bei der Entwicklung berücksichtigen sollte. Anhand des Fallbeispiels wird die Methode *Failure Mode Effect Analysis* angewendet.

Beim Einsatz von Analysetechniken entsteht für das Planungs- und Entwicklungsteam eine Liste von Gefahren, sortiert nach Prioritäten. In vielen Fällen ergibt es Sinn, auch Gefahren und die daraus resultierenden Konsequenzen mit niedriger Priorität zu berücksichtigen. Denn das Eintreten einer Gefahr kann Kosten verursachen. Demgegenüber steht das mit finanziellem Aufwand errichtete Sicherheitssystem, das genau diese Gefahr abwendet. Sie sollen aus diesem Kapitel mitnehmen, dass eine Kosten-Nutzen-Analyse durchaus sinnvoll sein kann. Diese sollte gegebenenfalls zur Entscheidung führen, auch in niedrig prioritäre Sicherheitssysteme zu investieren. Das Vorgehen wird als das *As Low As Reasonably Practical*-Prinzip bezeichnet.

Die *Failure Mode Effect Analysis* betrachtet beinahe auf statische Weise die System- und Gerätestruktur, um Fehlermöglichkeiten zu identifizieren. Die *Hazard and Operability*-Methode hat tendenziell einen dynamischen Ansatz. So wird das System in *Design Intents* aufgebrochen. Durch Anwendung von *Guide Words* werden in einer Teamarbeit die Konsequenzen untersucht. Am Ende soll eine Liste von identifizierten Gefahren und Verbesserungsvorschlägen entstehen. Ihnen werden die einzelnen *Hazard and Operability*-Schritte vorgestellt, am Fallbeispiel werden diese angewendet.

In Kapitel 7 werden Formeln angegeben, die Ausfälle von Systemen und Geräten mithilfe von Ausfallraten beschreiben. Zusammen mit dem Fallbeispiel *Fords Pinto*-Memo aus Kapitel 9 soll es Ihnen einen Einblick darin geben, wie Ausfallraten ermittelt werden können. Eine einfache Möglichkeit ist es, Ausfallraten einzelner Bauteile aus Handbüchern zu entnehmen und dann die Ausfallrate von Komponenten zu ermitteln.

Sind aber Daten über das Ausfallverhalten der Betrachtungseinheiten des Systems und Geräts vorhanden, kann mit statistischen parameterfreien Methoden die Zuverlässigkeitsfunktion abhängig von der Zeit ermittelt werden. Dazu benötigen Sie Kenntnisse über die Wahrscheinlichkeitstheorie. Konkret wird in diesem Kapitel der *Kaplan-Meier-Schätzer* hergeleitet, wofür die *Maximum-Likelihood*-Methode angewendet wird. Dazu wird die Differenzialrechnung zur Extremwertbestimmung aus der Mathematik benötigt.

Oft gibt es aber auch implizites Wissen über die Daten. So weiß möglicherweise der *Safety Engineer*, dass die Dichtefunktion der zu untersuchenden Daten einer Exponentialfunktion ähnlich ist. Dann muss nur ein einziger unbekannter Parameter geschätzt werden. Auch hier kann die *Maximum-Likelihood*-Methode zur Herleitung der Formel zur Schätzung des einzelnen Parameters angewendet werden. Um die Daten zur Analyse von Systemen und Geräten zu beschaffen, kann sehr viel Zeit vergehen. So muss eine Anzahl von Geräten beobachtet und bei Ausfällen müssen Zeitpunkte und Betriebsdauer dokumentiert werden. Diese Zeit kann sich der *Safety Engineer* nicht nehmen, sondern wird innerhalb eines kürzeren Zeitraums Schätzungen durchführen. Wenn zum Zeitpunkt der Schätzung nicht alle Daten der Betrachtungseinheiten zur Verfügung stehen, spricht er von *zensierten Daten*. Auch Systeme und Geräte, die nach wie vor in Funktion sind, werden bei der Schätzung berücksichtigt. Hier wird der Schätzer des Parameters ebenfalls mit der *Maximum-Likelihood*-Methode hergeleitet.

Bei der Anwendung des Schätzers werden Daten benötigt, die normalerweise in Datenbanken gespeichert sind. Im letzten Abschnitt des Kapitels wird gezeigt, wie Daten methodisch abgelegt werden können. Dabei werden die Grenzen des zu betrachtenden Systems, des Teilsystems und seine Komponenten definiert, um Überschneidungen der Datenaufnahme zu vermeiden und um doppelte Einträge in Datenbanken zu verhindern. So erhalten die Datenbanktabellen eine Struktur ähnlich der Struktur der Systemhierarchie.

Es werden Ihnen Datenbanktabellen zur Strukturierung des Ford-Pinto-Tanksystems vorgestellt, um Ausfälle der Komponenten in der Datenbank abzulegen.

2.2 Methoden zur qualitativen Analyse und Mischformen

In Kapitel 10 wird die erste Methode zur Systemmodellierung mit einem Fehlerbaum vorgestellt. Dabei werden Ereignisse mithilfe von Gattern zusammengeführt, um dann als Ergebnis ein unerwünschtes Ereignis zu ermitteln. Dazu wird das Fallbeispiel eines Reaktorunglücks (*Three Miles Island*) beschrieben, das in den USA vorgefallen war. Das Zusammenstellen des Fehlerbaums ist eine deduktive Methode (*top-down*). Angefangen wird mit dem unerwünschten Ereignis (Reaktorunfall im Fallbei-

spiel), der *Safety Engineer* teilt dieses in Zwischenereignisse auf und führt sie mit Gattern zusammen. Er wiederholt dies, bis alle Basisereignisse erreicht sind. Die dazu benötigten Gatter können mit Operatoren aus Mengenlehre und Wahrscheinlichkeitstheorie mathematisch beschrieben werden. So lässt sich der Fehlerbaum auch dazu nutzen, Fehlerwahrscheinlichkeiten und Nichtverfügbarkeiten quantitativ zu bestimmen.

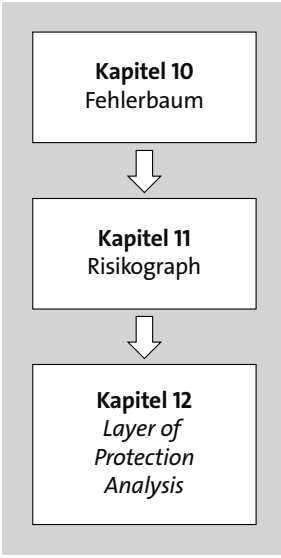


Abbildung 2.2 Methoden zur qualitativen Analyse sowie Mischformen

Nicht alle Systeme oder Geräte lassen sich durch Operatoren aus der Mengenlehre beschreiben, besonders wenn dynamische Situationen wie etwa ein Zustandswechsel auftreten. Zum Beispiel kann ein System oder Gerät von einem Betriebszustand in einen Instandsetzungszustand treten. Der Fehlerbaum würde sich unter Umständen verändern. Für die Modellierung komplexerer Systeme können dynamische Gatter eingesetzt werden, was aber die mathematischen Analysemöglichkeiten schwierig macht. *Petri-Netze*, beschrieben in Kapitel 6, eignen sich jedoch dafür, Zustandswechsel zu simulieren. Diese lassen sich dann in den Fehlerbaum integrieren.

Im Fallbeispiel des Reaktorunglücks wird ein Fehlerbaum demonstrativ konstruiert, und es wird gezeigt, wie qualitative Analysetechniken daran anzuwenden sind, um z. B. die Minimalschnitte zu bestimmen. Das sind die Ereignisse, die auftreten müssen, damit das unerwünschte Ereignis auftritt. Auch weitere qualitative Analysetechniken, wie die boolesche Reduktion und die disjunkte Zerlegung, werden vorgestellt.

Bei den quantitativen Analysetechniken geht es um die Bestimmung der Wahrscheinlichkeit des unerwünschten Ereignisses. Um diese zu bestimmen, werden Sie mit einfacher mathematischer Integration konfrontiert. Aus den berechneten Wahr-

scheinlichkeitswerten kann die Kenngröße der Sicherheitsanforderung für die Sicherheitstechnik des Systems oder Geräts aus der zentralen Tabelle aus Kapitel 3 bestimmt werden.

Basisereignisse sind die Eingänge des Fehlerbaums, sie haben unterschiedlichen Einfluss auf das unerwünschte Ereignis. Um zu bestimmen, wie groß der Einfluss einzelner Basisereignisse ist, kann die Sensitivitätsanalyse angewendet werden. Dafür werden Formeln präsentiert, die die Wichtigkeit der Basisereignisse bestimmen. Zur Veranschaulichung wird anhand des Fallbeispiels den Ereignissen Wahrscheinlichkeiten zugeordnet, und die Kenngrößen für die Wichtigkeit werden ermittelt. Diese werden dann grafisch dargestellt.

Zuletzt wird in dem Kapitel die *Monte Carlo*-Simulation als weitere Analysemethode vorgestellt. Sie ist eine einfache Methode, die genau dann ihre Anwendung findet, wenn Wahrscheinlichkeiten nicht genau bestimmt werden können. So wird ein Algorithmus vorgestellt, der Eingangsereignisse bestimmt, und dann wird die Verteilung des unerwünschte Ereignisses berechnet. *Monte Carlo*-Simulationen lassen sich auch bei dynamischen Gattern einsetzen und ersetzen so die analytische Berechnung, die ohnehin schwierig ist. Beispielhaft wird der Unterschied zwischen einem konventionellen und einem dynamischen Gatter gezeigt.

In Kapitel 10 wird die *Fehlerbaummethode* vorgestellt, die sowohl qualitative als auch quantitative Elemente hat. Anhand der zentralen Tabelle aus Kapitel 3 lässt sich mit der quantitativen Methode die Kenngröße der Sicherheitsanforderung bestimmen. Dafür werden aber konkrete Wahrscheinlichkeitswerte benötigt. Diese sind oftmals nicht vorhanden, vor allem dann nicht, wenn sich das Projekt in einer frühen Entwicklungsphase befindet.

Kapitel 11 behandelt den Risikographen, um mit qualitativen Methoden die Kenngrößen der Sicherheitsanforderung zu bestimmen. Das Fallbeispiel beschreibt das Zugunglück in *East Palastine, Ohio*, ein Ereignis aus der jüngster Vergangenheit. Hier wird das Konzept des Risikographen angewendet. *Frequency-N-Fatalities*-Diagramme werden eingesetzt, womit die Häufigkeiten für ein akzeptiertes Risiko nach dem *As Low As Reasonably Practical*-Prinzip bestimmt werden. Risikographen setzen Parameter wie Häufigkeit, Wahrscheinlichkeit, Möglichkeiten der Gefahrenabwehr und Konsequenzen ein. So werden in diesem Kapitel die Abstufungen der Parameter aufgezeigt, dann wird der Aufbau des Risikographen vorgestellt. Da nicht in jedem Fall alle Parameter benötigt werden, gehen auch daraus abgeleitete Risikographen hervor. Sie finden ihren Einsatz bei Sach- und Umweltschäden.

Die Anwendung von Risikographen ist nicht objektiv. So können bei unterschiedlichen Expertenteams auch unterschiedliche Ergebnisse erarbeitet werden. Aus diesem Grund gibt es die Möglichkeit, Risikographen zu kalibrieren. Am Fallbeispiel wird Ihnen das demonstriert.

Risikographen sind in sämtlichen Normen beschrieben, die der Vorgehensweise aus der Norm *IEC-61508* ähnlich ist (Norm *IEC-61508* übernahm wiederum den Risikographen aus einer älteren Norm). Die Norm *ISO-26262* beschreibt einen Risikographen, der für die Automobilindustrie zugeschnitten ist. Sie ist ähnlich aufgebaut, da aber im Autoverkehr katastrophale Auswirkungen mit vielen Opfern sehr selten sind, werden Kategorien ausgelassen und Parameter an die Bedürfnisse der Analysten angepasst.

Bereits in Kapitel 8 wurde die *Hazard and Operability*-Methode vorgestellt, um Gefahren zu ermitteln und technische Vorschläge zur Risikoreduzierung zu dokumentieren. Die in Kapitel 12 vorgestellte Methode *Layer of Protection Analysis* ist eine Erweiterung davon. Die technischen Vorschläge aus der *Hazard and Operability*-Methode können mit sogenannten Schutzebenen realisiert werden. Eine Vorgabe ist, dass sie unabhängig sind und nacheinander bei einer Gefahr in Aktion treten. So sind Schutzebenen z. B. ganz einfache Einrichtungen, wie Alarme, oder aber auch komplexere aktive Sicherheitssysteme, genannt *Independent Protection Layer*. Ihnen wird die Voraussetzung vorgestellt, wann ein Sicherheitssystem so bezeichnet werden darf. Einfache Formeln werden dazu verwendet, die Wahrscheinlichkeit des Ausfalls eines kompletten Sicherheitssystems mit allen Schutzebenen zu berechnen. Diese Formeln haben ihren Ursprung in den Grundlagen der Wahrscheinlichkeitstheorie. Die hier vorgestellte Methode ist allerdings ein halbquantitatives Verfahren. Zwar gibt es wie oben erwähnt Berechnungsmöglichkeiten, aber die Bestimmung der Häufigkeit des Eingangsereignisses erfolgt aus Tabellen. Somit ergeben die Ergebnisse der Berechnung nur eine Größenordnung. Sie sind dennoch sehr nützlich, insbesondere in der Anfangsphase eines Entwicklungsprojekts als Eingabe für das Entwicklungsteam.

Das beschriebene Fallbeispiel in diesem Kapitel ist das Brandunglück im *St.-Gotthard-Tunnel*. Hier wird die Anwendung der vorgestellten Methode demonstriert. So können die Wahrscheinlichkeiten für Ausfälle der Schutzebenen in die Tabelle *Layer of Protection Analysis* eingetragen werden, um daraus die Gesamtausfallwahrscheinlichkeit zu bestimmen. Die zentrale Tabelle aus Kapitel 3 liefert die Kenngröße für die Sicherheitsanforderung.

2.3 Methoden zur quantitativen Analyse

In vielen Fällen werden Sicherheitssysteme mit drei abstrakten Blöcken abgebildet. Die folgenden Komponenten stellen sie dar: Sensoren, Rechner (*Logic Solver*) und Aktoren (*Final Element*). Diese Blöcke sind in einer Reihe angeordnet, wie z. B. bei einer Reihenschaltung von Widerständen. Diese Architekturdiagramme werden Blockdiagramme genannt. Ihre Komponenten können sich dabei in einem Funktions- oder Fehlerzustand befinden. Ihnen wird in Kapitel 13 eine Erweiterung der Blockdia-

gramme vorgestellt, bei der nicht nur die Komponente selbst einem Block zugeordnet wird, sondern jede Fehlermöglichkeit der Komponente. Diese Art von Diagrammen wird *Zuverlässigkeitsblockdiagramm* genannt. Unterschiedliche Architekturen, z. B. die Redundanz, werden mit parallelen Blöcken dargestellt. Wenn Komponenten aber voneinander abhängig sind, werden sie seriell angeordnet. Das Fallbeispiel *Jakarta Incident* beschreibt in einem Beispiel die Ausfälle von vier Turbinen eines Flugzeugs. Sie sind redundant, und das Flugzeug fällt erst dann komplett aus, wenn alle Turbinen ausfallen.

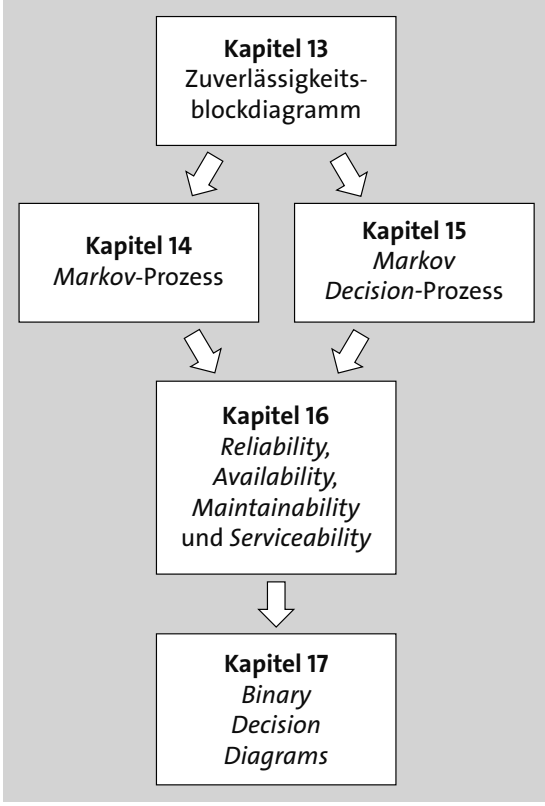


Abbildung 2.3 Methoden zur quantitativen Analyse

Im Gegensatz zum Fehlerbaum, bei dem das unerwünschte Ereignis bzw. der unerwünschte Fehler untersucht wird, steht beim Zuverlässigkeitsblockdiagramm die Funktion im Vordergrund. Dennoch kann der Fehlerbaum in ein Zuverlässigkeitsblockdiagramm überführt werden und umgekehrt.

Zur Berechnung der Zuverlässigkeiten von seriellen und parallelen Anordnungen von Blöcken werden Kenntnisse in Mengenlehre und Wahrscheinlichkeitstheorie benötigt. Es werden Formeln hergeleitet, die mit Beispielen für die Berechnung einhergehen. Die hergeleiteten Formeln aus Kapitel 13, zusammen mit den Formeln aus

Kapitel 7, können genutzt werden, um zeitliche Verläufe der Zuverlässigkeit von Systemen und Geräten zu beschreiben.

Der *Safety Engineer* kann bei der Verbesserung der Zuverlässigkeit einer Komponente vor der Entscheidung stehen, die Einzelkomponente robuster auszulegen oder eine Redundanz einzusetzen. Bei Verwendung von Formeln aus Kapitel 13 und Kapitel 7 kann über die Kurvenverläufe gezeigt werden, was die richtige Methode sein könnte. Dies soll dem *Safety Engineer* die Entscheidung erleichtern. Auch kann der *Safety Engineer* vor einer Entscheidung stehen, gleichartige oder unterschiedliche Komponenten für eine redundante Struktur einzusetzen. Anhand des Fallbeispiels und der Kurvenverläufe soll gezeigt werden, ob Vor- oder Nachteile dadurch entstehen.

In Kapitel 10 und Kapitel 13 wurden Methoden beschrieben, wie Fehler- und Zuverlässigkeitswahrscheinlichkeiten mit Fehlerbäumen und Zuverlässigkeitsblockdiagrammen quantitativ bestimmt werden können. Bei diesen Methoden wurde die Instandsetzung nur am Rande beachtet. Sie wird aber dafür gebraucht, die Verfügbarkeits- und Nichtverfügbarkeitswahrscheinlichkeit zu bestimmen. Instandsetzung ist ein Zustand, der sich durch die oben genannten Methoden nur auf eine sehr einfache Art modellieren lässt. Anhand des Fallbeispiels vom Seilbahnunglück am *Monte Mottarone* wird gezeigt, wie wichtig die Instandsetzung ist und dass sich diese auch auf die Zuverlässigkeit auswirken kann.

In Kapitel 14 wird eine Methode vorgestellt, um sowohl den normalen Betrieb als auch die Instandsetzung in einem Modell unterzubringen. Die Modellierungsmethode ist der *Markov*-Prozess. Das Kapitel beginnt mit der Definition des Prozesses und der Idee, wie ein Wechsel von einem Zustand in den nächsten erfolgen kann. An dieser Stelle werden Kenntnisse in der Wahrscheinlichkeitstheorie aus Kapitel 7 benötigen. Die Übergangswahrscheinlichkeiten zwischen den Zuständen können in kompakter Weise mit einer sogenannten Übergangsmatrix dargestellt werden. Diese wird von einem einfachen Modell mit drei Zuständen hergeleitet. Dann wird die Übergangsmatrix für beliebig viele Zustände verallgemeinert. Ziel ist es hier, ein Modell zu erhalten, das stets die Wahrscheinlichkeiten für alle Zustände und die Veränderung der Wahrscheinlichkeiten bei Zustandswechsel bestimmen kann. Vorteilhaft sind Kenntnisse über Matrizen und ihre rechnerische Handhabung. Bei der Betrachtung eines längeren Zeitraums kann der dynamische Verlauf der Zustandswahrscheinlichkeiten über die Zeit mithilfe von Differenzialgleichungen ausgedrückt werden. Ihnen sollten Differenzialgleichungen bekannt sein, wobei sie in diesem Kapitel bewusst einfach gehalten werden.

Gegen Ende des Kapitels werden drei Beispiele beschrieben, wie aus unterschiedlichen Zuverlässigkeitsdiagrammen (Einblock-, redundante Blöcke und redundante Blöcke mit unterschiedlichen Fehlermöglichkeiten mit entdeckbaren und nicht-ent-

deckbaren Fehlern) Markov-Prozesse und ihre Übergangsmatrizen erstellt werden können.

Eine Erweiterung des *Markov*-Prozesses ist der *Markov Decision*-Prozess, der in Kapitel 15 behandelt wird. So spielen die Entscheidungen eines Akteurs eine Rolle, die einen Zustandswechsel beim *Markov*-Prozess hervorrufen. Der Akteur (Fahrer eines Fahrzeugs) kann beispielsweise die Entscheidung treffen, das Fahrzeug zu fahren (Zustand *Betrieb*) oder es in die Instandsetzung zu bringen (Zustand *Instandsetzung*). Dabei stellt sich die Frage, welche Aktionen der Akteur wählen sollte, um eine optimale Entscheidung bezüglich der Kosten zu treffen. Dies wird dadurch erreicht, dass bei den Übergängen zwischen den Zuständen Belohnungen definiert werden. Sie erfahren, dass die Wahl einer Aktion für einen Zustandswechsel einer Strategie zugeordnet wird. Die Strategie ist dann optimal, wenn die Summe der Belohnungen ein Maximum erreicht. Zur Berechnung werden Zustandsbelohnungs- und Aktionsbelohnungsfunktionen hergeleitet. Wenn diese bezüglich der Belohnung optimiert sind, werden sie *Bellmann*-Gleichungen genannt. Zum Verständnis wird der Inhalt von Kapitel 14 voraussetzt.

Für die rechnerische Berechnung der Zustandswahrscheinlichkeiten können die *Bellmann*-Gleichungen in eine iterative Form gebracht werden. Somit vereinfacht sich die Lösung der Gleichungen. Optimierte Strategien können über die Simulation mit einem Rechner ermittelt werden.

Im Fallbeispiel des Kapitels wird ein Autounfall durch falsche Nutzung des Autopilotsystems der Automobilmarke *Tesla* beschrieben. Die Forschung auf dem Gebiet *autonomes Fahren und künstliche Intelligenz* hat derzeit eine große Aufmerksamkeit. *Markov Decision*-Prozesse spielen dabei eine Rolle, und deswegen soll es einen kurzen Ausflug in die künstliche Intelligenz geben, passend zum Fallbeispiel. Der aktuelle Zustand eines Systems (Fahrzeug und seine Umgebung) kann mit Kamerasystemen erfasst werden. Da aber das Fahrzeug und seine Umgebung viel zu kompliziert sind, um als *Markov*-Prozess modelliert zu werden, wird ein neuronales Netz dafür verwendet. Die Aktionen des Akteurs (Fahrers) sind die Lenkbewegungen. Belohnungen werden durch das neuronale Netz modelliert und an die Ausgänge ausgegeben. Ihnen wird ein Algorithmus vorgestellt, mit dem ein Fahrzeug selbstlernend auf Fahrsituationen reagieren kann. Dieser Teilbereich der künstlichen Intelligenz wird *Reinforcement Learning* genannt. Besondere Kenntnisse bezüglich der künstlichen Intelligenz brauchen Sie nicht, da das Thema in dem Kapitel einfach gehalten wird.

Das *Kursk*-Unglück ist das Fallbeispiel aus Kapitel 16. Als mögliche Unglücksursache gilt die unsachgemäße Handhabung von Sprengstoff. Normalerweise sollte Sprengstoff in regelmäßigen Abständen ausgetauscht werden, da Zerfalls- und Oxidierungsprozesse auftreten können. So ist dies bei seiner Lagerung eine wichtige Instandsetzungsmaßnahme. Kapitel 14 bringt Ihnen näher, wie der *Markov*-Prozess genutzt werden kann, um die Zuverlässigkeit von Systemen oder Geräten zu modellieren.

Durch kleine Veränderungen in der Modellierung kann aber auch die Verfügbarkeit eines Systems oder Geräts mit Markov bestimmt werden. Dieses Kapitel zeigt Ihnen, wie ein einfaches, ein serielles und ein redundantes (paralleles) System modelliert wird, um sowohl die Zuverlässigkeit als auch die Verfügbarkeit zu bestimmen. Das Ergebnis sind immer Übergangsmatrizen und Übergangsratenmatrizen, die in ein Gleichungssystem von Differenzialgleichungen erster Ordnung überführt werden können. Zur Bestimmung der Verfügbarkeit ist die Lösung von Differenzialgleichungen nicht immer trivial. Deswegen kann auch auf die stationäre Betrachtung ausgewichen werden. Schreitet nämlich die Zeit voran, können die Wahrscheinlichkeiten der Zustände konstante Werte annehmen. Es ist in diesem Kapitel von Vorteil, wenn Sie sich bereits mit der Lösung von einfachen Differenzialgleichungen beschäftigt haben. Die Lösung des Gleichungssystems von komplexeren Systemen erfordert weitergehende Kenntnisse, z. B. die *Laplace*-Transformation. Dabei wird das Gleichungssystem in einen sogenannten Bildbereich transformiert. Die Lösung des Gleichungssystems vereinfacht sich zwar dadurch, aber dafür ist sie arbeitsaufwendig. In diesem Kapitel werden die zeitlichen Verläufe der Wahrscheinlichkeiten von Zuständen eines redundanten Systems oder Geräts als Formeln hergeleitet und in einem Diagramm zur Verifizierung gezeigt.

Alternativ zum *Markov*-Prozess kann ein Sicherheitssystem als Sensor-Rechner-Aktor-Blockdiagramm mit zwei Fehlermöglichkeiten (entdeckbaren und nicht-entdeckbaren Fehlern) ähnlich wie beim Zuverlässigkeitsblockdiagramm modelliert werden. Die Wahrscheinlichkeit für den Ausfall eines Sicherheitssystems mit Berücksichtigung der Instandsetzung können so berechnet werden. Im Gegensatz zu Zuverlässigkeitsblockdiagrammen ist hier die Idee, Prüfintervalle einzuführen. Das Sicherheitssystem wird regelmäßig überprüft und danach als neuwertig angesehen. Das System erhält somit in regelmäßigen Abständen ein *Reset*, sodass sich die Wahrscheinlichkeitsverläufe nach der Prüfung wie bei einem neuen System verhalten. Es wird gezeigt, wie Wahrscheinlichkeiten für den Ausfall von einfachen und beliebigen Arten von redundanten Sicherheitssystemen berechnet werden können. Die Kenngröße der Sicherheitsanforderungen kann auch hier aus der zentralen Tabelle von Kapitel 3 bestimmt werden.

Fehlerbäume und Wahrheitstabellen dienen beide der Darstellung von booleschen Ausdrücken. Fehlerbäume stellen die Ausdrücke oftmals optimiert dar, während Wahrheitstabellen diese für alle Kombinationen aus Eingangsereignissen zeilenweise angeben. Ich zeige Ihnen in Kapitel 17, wie sich Fehlerbäume und Wahrheitstabellen in bereits bekannte Zuverlässigkeitsfunktionen aus Kapitel 7 mithilfe der booleschen Regeln und der Wahrscheinlichkeitstheorie überführen lassen. Bei Anwendung des *shannonschen* Zerlegungssatzes wird das Thema *Binary Decision Diagram* nähergebracht. Diese Darstellungsform ist ein Diagramm mit Knoten (Eingangsereignissen) und Kanten zur Modellierung von Systemen.

Es wird gezeigt, wie einzelne Gatter von Fehlerbäumen durch das *Binary Decision Diagram* modelliert werden. So kann aus einem kompletten Fehlerbaum ein *Binary Decision Diagram* aufgebaut werden, ohne den Umweg über die Wahrheitstabelle gehen zu müssen. Auch hier lassen sich somit Zuverlässigkeiten und Verfügbarkeiten ähnlich wie beim Fehlerbaum berechnen. Es soll aber darauf hingewiesen werden, dass die Modellierungsmöglichkeiten, insbesondere für die Verfügbarkeit, im Vergleich zum *Markov*-Prozess eingeschränkt sind.

Es wird gezeigt, wie das *Binary Decision Diagram* über Optimierungsmöglichkeiten verkleinert werden kann. Das Fallbeispiel, das das Sicherheitssystem *Permissive Action Link* bei Atombomben beschreibt, dient als Vorlage für den Aufbau einer Wahrheitstabelle und eines einfachen *Binary Decision Diagram*. Durch Anwendung der vorgestellten Optimierungsmöglichkeiten zeige ich Ihnen, wie das Diagramm verkleinert wird.

Kapitel 3
Normen

Menschengemachte Unglücke sind nicht neu und ereigneten sich in den letzten zwei Jahrhunderten immer wieder. Durch den technischen Fortschritt sind aber die Konsequenzen vieler Unglücke verheerend geworden. Beispiele sind die Unglücke in *Seveso* und *Bhopal*. Deswegen entwickelte sich gerade in den letzten Jahrzehnten ein größeres Bewusstsein bezüglich der Sicherheit, und deswegen sind Normen entstanden. Sie sind wichtig, damit sich Hersteller von Systemanlagen und Geräten bei der Entwicklung und Konstruktion darauf beziehen können. In diesem Kapitel werden einige der wichtigsten Normen aus der Sicherheitstechnik vorgestellt und beschrieben. So erhalten Sie einen Überblick über den aktuellen Stand bei Sicherheitsnormen.

Alle weiteren Kapitel werden sich auf die hier erwähnten Normen beziehen, sodass es sinnvoll ist, sich diesen Überblick zu schaffen. Gegen Ende des Kapitels sollten Sie die wichtigsten Normen für die restlichen Kapitel einordnen können. Insbesondere mache ich auf Tabelle 3.3 aus der Norm *IEC-61508* aufmerksam, da sie von zentraler Bedeutung für die Kenngrößenermittlung der Sicherheitsanforderungen ist und deswegen in sämtlichen Kapiteln verwendet wird.

3.1 Überblick

Normen sind Richtlinien für Industrie, Behörden, Betriebe etc. und dienen der Einhaltung von Gesetzen. Diese werden durch nationale Parlamente, also den Gesetzgeber, erlassen. Die Gesetze beziehen sich auf den Stand der Technik, ohne diese zu nennen. Industrie, Behörden, Betriebe etc. sollten sie sich bei der Entwicklung von Systemen und Geräten selbst aneignen. Der Gesetzgeber bezieht sich in den Gesetzen nicht einmal auf Normen, sondern verallgemeinert seine Richtlinien durch die Vorgabe, dass alle entwickelten Produkte dem Stand der Technik entsprechen müssen. Abbildung 3.1 zeigt vereinfacht das Verfahren der Richtliniengebung durch die Europäischen Gemeinschaft (*EU*), die für den Erlass der Richtlinien zuständig ist. Als Beispiel ist hier die Sicherheit von elektrischen Betriebsmitteln (Richtlinie 2014/35/EU [2]) zu nennen. Bei Beschluss werden die Richtlinien an die nationalen Regierungen und Parlamente weitergereicht, wo sie in nationalen Gesetze umgeschrieben werden. Danach werden die Gesetze durch die Parlamente der EU-Länder ratifiziert. Hersteller von Geräten und Systemen müssen sich also an nationale Gesetze halten.

7.5 Statistische Kennzahlen

Einzelfallbetrachtungen über das Ausfallverhalten eines Geräts sind wegen der Zufälligkeit nicht aussagekräftig. Aus diesem Grund wird die Mittelwertbildung eingesetzt, um einen Überblick über sämtliche Systeme und Geräte zu erhalten. Zuverlässigkeitsfunktionen sind aber für den schnellen Einblick ungeeignet. Deshalb werden die Kenngrößen *mittlere Betriebszeit*, *mittlere Reparaturzeit* und *mittlere Ausfallzeit* in den folgenden Abschnitten vorgestellt.

7.5.1 Mittlere Betriebszeit

Der Mittelwert der Betriebszeit von Systemen und Geräten bis zum Ausfall wird mittlere Betriebszeit genannt oder *Mean Time to Failure*, kurz *MTTF*. Formel [7.19] zeigt die Berechnung aus der Zuverlässigkeitsfunktion bzw. Ausfallfunktion.

$$MTTF = \int_0^\infty R\,(t)\,dx = \int_0^\infty \big(1 - F(t)\big)\,dx \qquad [7.19]$$

Zur Erklärung wird grafisch die *MTTF* in Abbildung 7.14 mit Pfeilen dargestellt. Sie zeigt die Betriebszeit von drei Geräten und deren Ausfall zu einem bestimmten Zeitpunkt. Der Mittelwert der Betriebszeiten (wenn eine unendliche Anzahl von Geräten betrachtet wird) ist dann die Kennzahl *MTTF*. In Abbildung 7.14 wird gezeigt, dass das erste Gerät repariert wird. Nach einer Reparatur wird zur Vereinfachung das Gerät als neuwertig betrachtet.

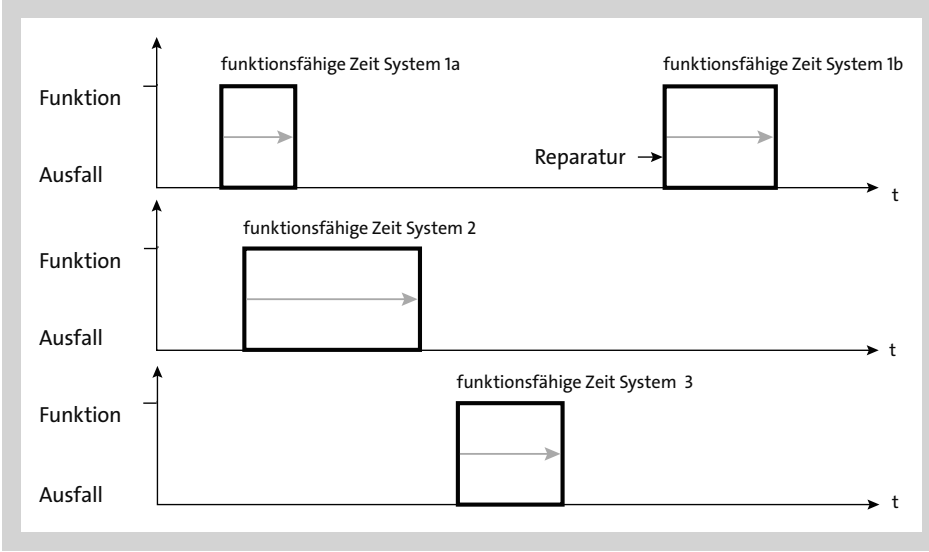


Abbildung 7.14 Betriebszeiten von drei Geräten

Wird nun angenommen, dass die Zuverlässigkeitsfunktion eine Exponentialfunktion ist (siehe Formel [7.15]), ergibt sich aus der *MTTF* von Formel [7.19] die Formel [7.20]. So zeigt sich, dass die mittlere Betriebsdauer bis zum Ausfall gleich dem Kehrwert der Ausfallrate $1/\lambda$ ist.

[7.20]
$$MTTF = \int_0^\infty R(t)\,dx = \int_0^\infty e^{-\lambda \cdot t}\,dx = \frac{1}{\lambda}$$

7.5.2 Mittlere Reparaturzeit

Die mittlere Reparaturzeit nach dem Ausfall eines Geräts wird *Mean Time to Repair* genannt, kurz *MTTR*. Abbildung 7.15 zeigt die Dauer t_0 und t_1 für die Reparatur eines Geräts. Es wird angenommen, dass das Gerät nach der Reparatur als neuwertig betrachtet werden kann.

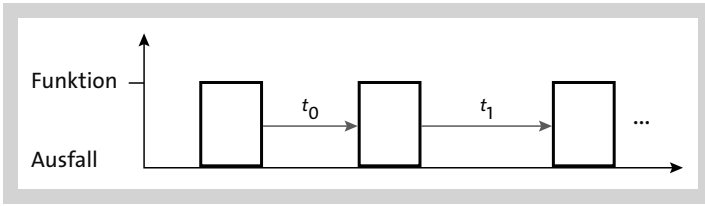


Abbildung 7.15 MTTR

Mithilfe des arithmetischen Mittelwerts kann *MTTR* durch den Quotienten aus der Summe der Reparaturzeiten und der Anzahl der reparierten Geräte ermittelt werden, siehe Formel [7.21]. So ist der arithmetische Mittelwert eine Approximation des tatsächlichen Mittelwerts.

[7.21]
$$MTTR = \frac{t_0 + t_1 + \ldots + t_n}{n}$$

Analytisch wird *MTTR* aus der Verteilungsfunktion $G(t)$ der Reparaturzeiten berechnet, siehe Formel [7.22]. Da $G(t)$ nicht leicht herzuleiten ist, hat sie einen begrenzten praktischen Einsatz.

[7.22]
$$MTTR = \int_0^\infty (1 - G(t))\,dx$$

Deswegen soll angenommen werden, dass die Reparaturrate konstant ist und demnach die Verteilungsfunktion der Reparaturzeiten einer Exponentialfunktion unterliegt. Eine konstante Reparaturrate bedeutet, dass unabhängig vom Ausfall die Reparatur des Geräts innerhalb einer konstanten Zeit durchgeführt werden kann. Dies ist zwar nicht allgemeingültig, aber dennoch eine realistische Annahme. Zum Beispiel

steht ein Fahrzeug, das morgens in die Werkstatt gebracht wird, abends nach der Reparatur zur Abholung bereit. In diesem Fall ist die Reparaturrate $1/8h$. Eine vereinfachte Verteilungsfunktion wird durch Formel [7.23] gegeben. Dabei ist μ die Reparaturrate.

$$G(t) = 1 - e^{-\mu t} \qquad [7.23]$$

Die Verteilungsfunktion lässt sich in Worten wie folgt interpretieren: Bei einem neuwertigen Gerät (kleines *t*) sind Reparaturen selten, sodass die Wahrscheinlichkeit für eine Reparatur gegen null geht. Nun verschleißen laufende Systeme und Geräte nach der Zeit. Bei älteren Systemen und Geräten (größeres *t*) kommen Probleme häufiger vor, sodass die Wahrscheinlichkeit für eine Reparatur stets größer wird. Dieses Denkmodell bezieht die Möglichkeit von häufigen Reparaturen wegen Frühausfällen natürlich nicht ein, dafür aber die Ausfälle durch Verschleiß.

Die *MTTR* lässt sich mit Formel [7.22] und Formel [7.23] umstellen, und es ergibt sich Formel [7.24].

$$MTTR = \int_0^\infty (1 - G(t))\, dx = \int_0^\infty e^{-\mu t}\, dx = \frac{1}{\mu} \qquad [7.24]$$

Die *MTTR* ist also der Kehrwert der Reparaturrate μ, wenn die Verteilungsfunktion der Reparaturzeiten eine Exponentialfunktion ist.

7.5.3 Mittlere Ausfallzeit

Die mittlere Ausfallzeit *Mean Time Between Failures*, kurz *MTBF*, ist die mittlere Zeit zwischen zwei Ausfällen eines Geräts. Die mittlere Betriebszeit wird durch *MTTF* ausgedrückt. Im Mittel wird nach einem Ausfall das System oder Gerät in der Zeit *MTTR* repariert. So kann *MTBF* durch die beiden Kenngrößen *MTTF* und *MTTR* ausgedrückt werden. Abbildung 7.16 veranschaulicht das Verhältnis zwischen *MTBF*, *MTTF* und *MTTR*.

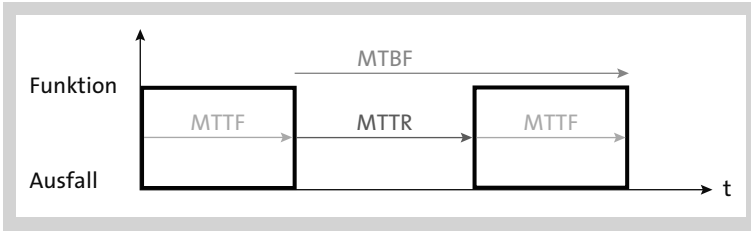


Abbildung 7.16 MTBF

Tatsächlich ist, entgegen der Darstellung in Abbildung 7.16, die mittlere Betriebsdauer bis zum Ausfall (also *MTTF*) in der Regel viel größer als die mittlere Reparaturzeit (*MTTR*) So kann in manchen Fällen *MTTR* vernachlässigt werden, wenn die Bedin-

gung *MTTF >> MTBF* eintritt. Formel [7.25] gibt das Verhältnis zwischen *MTBF*, *MTTF* und *MTTR* an:

[7.25] $$MTBF = MTTF + MTTR$$

Eine weitere Kenngröße, die sich aus *MTBF*, *MTTF* und *MTTR* ableiten lässt, ist die Verfügbarkeit zu einem unbestimmten Zeitpunkt (engl. *Point Availability*, kurz PA). Ist ein Gerät reparierbar, wird gern die Verfügbarkeit statt der Zuverlässigkeit verwendet. Die Kenngröße *PA* ist eine Wahrscheinlichkeit, dass zu einem unbestimmten Zeitpunkt ein Gerät funktionsfähig ist. Formel [7.26] zeigt dies:

[7.26] $$PA = \frac{MTTF}{MTTF + MTTR}$$

Die Kenngröße *PA* ist das Verhältnis zwischen der mittleren Betriebsdauer bis zu einem Ausfall (also *MTTF*) und der mittleren Betriebsdauer zwischen zwei Ausfällen (also *MTTR*). Unter der Annahme, dass es sich bei den Ausfallraten und den Reparaturraten um Exponentialverteilungen handelt, kann *PA* über Formel [7.27] ermittelt werden. Sie ergibt sich aus Formel [7.20] und Formel [7.24].

[7.27] $$PA = \frac{\frac{1}{\lambda}}{\frac{1}{\lambda} + \frac{1}{\mu}} = \frac{\mu}{\lambda + \mu}$$

7.6 Ausfallrate

Bereits im Fallbeispiel in Abschnitt 7.1 wurden Komponenten aufgeführt, die während der Betriebsdauer ausfallen, etwa die neue Turbine, die in Auftrag gegeben und nicht ausreichend am Flugzeug getestet wurde. Somit fiel eine hohe Anzahl der Turbinen im Betrieb aus, und es folgte in der Regel ein Absturz des Flugzeugs. Die Häufigkeit des Ausfalls wird über die Ausfallrate $\lambda(t)$ bestimmt. Im Allgemeinen ist sie abhängig von der Zeit. Die Kenngröße für die Ausfallrate $\lambda(t)$ gibt an, wie oft in einem Intervall $[t,t+\Delta t]$ ein Ausfall stattfindet. Es wird dabei vorausgesetzt, dass das Gerät zuvor in Betrieb war. Also gibt es im Intervall $[0, t]$ keinen Ausfall. Dies ist eine Bedingung, und es gilt Formel [7.6]. Ausgedrückt wird $\lambda(t)$ durch Formel [7.28], was die Ausfallwahrscheinlichkeit innerhalb eines Zeitabschnitts Δt ist.

[7.28] $$\lambda(t) = \lim_{\Delta t \to 0} \frac{1}{\Delta t} Pr\{Ausfall\ zwischen\ t \ldots t + \Delta t \mid Funktion\ zwischen\ 0 \ldots t\}$$

Wird Formel [7.6] in Formel [7.28] eingesetzt, ergibt sich:

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{1}{\Delta t} \frac{Pr\{Ausfall\ zwischen\ t \ldots t + \Delta t, Funktion\ zwischen\ 0 \ldots t\}}{Pr\{Funktion\ zwischen\ 0 \ldots t\}}$$

Abbildung 7.17 zeigt den Zusammenhang zwischen dem Ereignis *A*: *Ausfall zwischen t … t + Δt* und der Bedingung *B*: *Funktion zwischen 0 … t*. Tatsächlich ist Funktion zwischen 0 … *t* und *Ausfall zwischen t und* ∞ die gleiche Menge, siehe auch Formel [7.7] und Formel [7.8]. Es ergibt sich Formel [7.29].

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{1}{\Delta t} \frac{Pr\{Ausfall\ zwischen\ t \ldots t + \Delta t, Ausfall\ zwischen\ t \ldots \infty\}}{Pr\{Funktion\ zwischen\ 0 \ldots t\}} \quad [7.29]$$

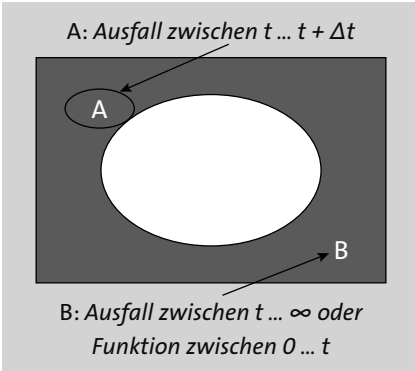


Abbildung 7.17 Ausfall innerhalb von dt

Nun ist es ersichtlich, dass die Menge *A* eine Teilmenge von *B* ist. Dadurch ergibt die Multiplikationsoperation aus den Mengen *A* und *B* die Menge *A*. Somit kann Formel [7.29] durch Formel [7.30] vereinfacht werden:

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{1}{\Delta t} \frac{Pr\{Ausfall\ zwischen\ t \ldots t + \Delta t\}}{Pr\{Funktion\ zwischen\ 0 \ldots t\}} \quad [7.30]$$

Die Wahrscheinlichkeit *Pr*{*Funktion zwischen* 0 … *t*} ist identisch mit *Pr*{τ > *t*}, siehe Formel [7.8]. Die Zufallsvariable τ beschreibt den Ausfall nach *t*. Eingesetzt, ergibt sich dieses:

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{1}{\Delta t} \frac{Pr\{Ausfall\ zwischen\ t \ldots t + \Delta t\}}{Pr\{\tau > t\}} \quad [7.31]$$

Es gilt nun die Definition der Zuverlässigkeit unter Verwendung der Formel:

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{1}{\Delta t} \frac{Pr\{Ausfall\ zwischen\ t \ldots t + \Delta t\}}{R(t)} \quad [7.32]$$

Die Wahrscheinlichkeit für einen *Ausfall zwischen t … t* + Δ*t* kann mithilfe der Fehlerwahrscheinlichkeit *F*(*t*) ausgedrückt werden. Abbildung 7.8 zeigt die Wahrscheinlichkeit in einem Bereich zwischen *t … t* + Δ*t* und kann durch *F*(*t … t* + Δ*t*) – *F*(*t*) ausgedrückt werden kann. Somit ergibt sich nach Umformung:

[7.33]
$$\lambda(t) = \frac{1}{R(t)} \lim_{\Delta t \to 0} \frac{F(t..t+\Delta t) - F(t)}{\Delta t}$$

In Formel [7.33] ist die Grenzwertbestimmung des Differenzquotienten von $F(t)$ dargestellt. Dies ist die mathematische Ableitung von $F(t)$. Damit ergibt sich:

[7.34]
$$\lambda(t) = \frac{1}{R(t)} \frac{dF(t)}{dt}$$

Letztendlich lässt sich $F(t)$ durch die Formel [7.9] ersetzen. Demnach wird nach $1 - R(t)$ abgeleitet. Formel [7.35] ist ein Ausdruck der Beziehung zwischen Ausfallrate und Zuverlässigkeitsfunktion.

[7.35]
$$\lambda(t) = -\frac{1}{R(t)} \frac{dR(t)}{dt}$$

Die Ausfallrate ist also der Quotient aus der Ableitung der Zuverlässigkeitsfunktion und der Zuverlässigkeitsfunktion selbst.

Abbildung 7.18 ist ein Beispiel für einen Verlauf der Ausfallrate als Funktion der Zeit t. Wegen ihrer Charakteristik wird diese Funktion als Badewannenkurve bezeichnet. Am Anfang stellt die Kurve erhöhte Werte von Ausfallraten dar, die nach der Zeit kleiner werden. Dieser zeitliche Bereich ist den Frühausfällen zuzuordnen. Diese entstehen oftmals wegen Produktionsfehlern, die sich dann im Feld anfangs bemerkbar machen. Dann geht der Verlauf der Kurve in eine Phase mit annähernd konstanter Ausfallrate über. Hier bietet es sich an, die Funktion der Ausfallrate mit einer Exponentialverteilung zu approximieren. Später steigen die Ausfälle stark an. Diesem zeitlichen Bereich werden Verschleißausfälle zugeordnet. Das Gerät altert zunehmend durch die Benutzung oder Umwelteinflüsse, und somit kommen Ausfälle häufiger vor.

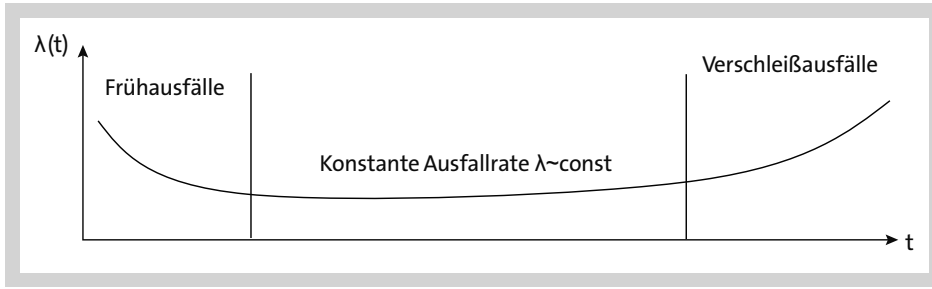


Abbildung 7.18 Badewannenkurve der Ausfallrate

Formel [7.35] lässt sich wie folgt umstellen:

[7.36]
$$-\lambda(t)dt = \frac{dR(t)}{R(t)}$$

Wird die Formel auf beiden Seiten integriert, folgt:

$$-\int_0^t \lambda\,(t)dt = \int_0^t \frac{dR(t)}{R(t)} \qquad [7.37]$$

Auf der rechten Seite der Gleichung wird die Integration angewendet, dann folgt:

$$-\int_0^t \lambda\,(t)dt = ln\big(R(t)\big)\Big]_0^t = ln\big(R(t)\big) - ln\big(R(0)\big) = ln\big(R(t)\big) \qquad [7.38]$$

Die Grenzwertbedingung $R(0) = 1$ ergibt Sinn, da ein Gerät direkt nach dem ersten Einsatz ($t = 0$) die Zuverlässigkeitswahrscheinlichkeit von eins haben sollte. Erst danach nimmt die Zuverlässigkeit ab. Nach Anwendung der Exponentialfunktion auf beiden Seiten der Gleichung und nach der Umstellung ergibt sich:

$$R(t) = e^{-\int_0^t \lambda(t)\,dt} \qquad [7.39]$$

Formel [7.39] ist die allgemeine Gleichung für die Zuverlässigkeit bei nicht konstanter Ausfallrate. Wird in dieser Formel $\lambda(t)$ durch einen konstanten Parameter λ ersetzt, ergibt sich Formel [7.15], da sich λ aus der Integration ziehen lässt. So lässt sich der mittlere Bereich von Abbildung 7.18 durch Formel [7.15] modellieren.

Die Einheit der Ausfallrate von Systemen und Geräten (aber auch Komponenten, Baugruppen und Bauelementen) wird oftmals in *Function in Time*, kurz *FIT*, angegeben. Dies ist die Anzahl der Ausfälle in einem Zeitintervall von eine Milliarde Stunden, siehe Formel [7.40].

$$[\lambda] = FIT = 1 \cdot 10^{-9}\frac{1}{h} \qquad [7.40]$$

7.7 Nichtverfügbarkeit und Ausfallrate des Sicherheitssystems

Bereits in Abschnitt 3.3.1 wurden die Kenngrößen *PFD* und *PFH* angesprochen, und eine Zuordnung der Sicherheitsintegritätslevels wurde durch Tabelle 3.3 angegeben. *PFD* ist dabei die mittlere Nichtverfügbarkeit des Sicherheitssystems. Diese wird angewendet bei einer niedrigen Anforderungsrate. *PFH* ist ein Ausdruck für die Ausfallrate des Sicherheitssystems. Bei hoher Anforderungsrate des Sicherheitssystems findet diese ihre Anwendung.

7.7.1 Probability for Dangerous Failure on Demand, PFD

Die Nichtverfügbarkeit eines sicherheitsgerichteten Systems bei einem notwendigen Einsatz (*On Demand*) ist ein wichtiger Kennwert. Diese wird für die *SIL*-Bestimmung

Kapitel 15
Markov Decision-Prozess

In Kapitel 14 wurde der Markov-Prozess als Alternative zu Fehlerbaum (siehe Kapitel 10) und Zuverlässigkeitsblockdiagramm (siehe Kapitel 13) zur Bestimmung der Zuverlässigkeit und Verfügbarkeit präsentiert. In diesem Kapitel wird der *Markov Decision*-Prozess vorgestellt. Dabei wird der Markov-Prozess derart erweitert, dass durch die Entscheidung des Akteurs ein Markov-Modell in ein anderes wechseln kann. Der Anwender hat somit eine weitere Möglichkeit zur Modellierung. Der Markov Decision-Prozess befasst sich unter anderem mit der Optimierung der Entscheidungen des Akteurs bezüglich der Belohnungen, die er bei seinen Entscheidungen erhält.

Bei dem Thema lade ich Sie dazu ein, einen Blick von einer anderen Seite auf das Thema zu werfen (selbst wenn der Zusammenhang mit Safety Engineering nicht sofort ersichtlich ist). Auch die künstliche Intelligenz befasst sich mit dem Markov Decision-Prozess. Er wird genutzt, um selbstständig neuronale Netze zu erlernen.

Sie werden auch eine Schwachstelle der Markov-Prozesse kennenlernen. Ihre Anwendung kann dazu führen, dass ein Modell zu viele Zustände erhält, und die Handhabung kann schwierig sein. Neuronale Netze können die Modellierung vereinfachen.

15.1 Fallbeispiel: Das Autopilotsystem des Tesla Model S

Im Jahr 2016 fuhr der Fahrer eines *Teslas* mit eingeschaltetem Autopiloten auf einer Autobahn durch Florida, siehe auch Artikel [84]. Der Autopilot ist eigentlich nur als Fahrhilfe ausgelegt, z. B. um die Spur zu halten und Kollisionen zu vermeiden. Der Blogpost [85] der Firma Tesla weist daraufhin, dass das Fahrzeug stets vom Fahrer unter Kontrolle gehalten werden muss und dass er immer verantwortlich für das Fahrzeug bleibt (der Autopilot von Tesla wird als *SAE-Level-2* klassifiziert).

Das Autopilotsystem steuert dabei die Fahrzeuggeschwindigkeit und behält die Spur des Fahrzeugs innerhalb der Fahrbahnmarkierungen. Dafür hat das Fahrzeug Ultraschallsensoren an der Seite, eine Kamera an der Windschutzscheibe und einen Radarsensor an der Front, siehe Bericht [86] der *National Highway Traffic Safety Administration*, kurz *NHTSA*. Die Ultraschallsensoren sind sehr ungenau und spielen bei der Steuerung eine untergeordnete Rolle. Die Kamera kann Objekte erkennen, hat aber

Einschränkungen bei starkem Lichteinfall, insbesondere bei direkter Sonneneinstrahlung. Der Radarsensor misst Entfernungen zwar sehr genau, kann jedoch keine Objekte unterscheiden. Bei dieser Fahrt schien die Sonne dem Fahrer und der Kameralinse entgegen, siehe Abbildung 15.1. So war das Kamerasystem nicht in der Lage, einen Lastwagen, der quer zur Autobahn fuhr, zu erkennen. Das Fahrzeug raste unter dem Lastwagen durch, und das Autodach riss ab.

Der Bericht [86] bestätigte den Blogpost von Tesla darin, dass der Autopilot den Anforderungen entsprechend funktionierte. Einen Kritikpunkt stellte dabei die *NHTSA* heraus: Der Autopilot hat durch die Namensgebung dem Fahrer ein falsches Gefühl der Sicherheit gegeben. Der Fahrer überschätzte also die Fähigkeiten des Autopiloten.

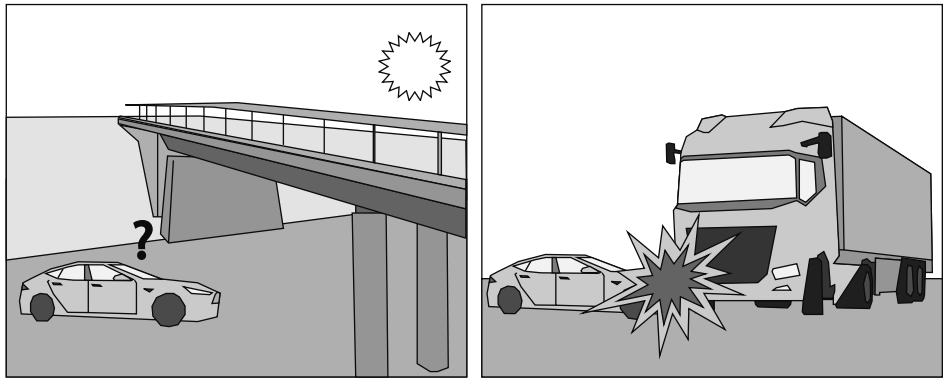


Abbildung 15.1 Das Autopilotsystem des Tesla Model S

Unabhängig von diesem Fallbeispiel stellte die *NHTSA* in dem Bericht [87] die aktuelle Unfallstatistik von autonomen Fahrzeugen vor, die nach *SAE-Level-3 bis -5* ausgelegt sind. Bei dem Unfall des Fallbeispiels handelte es sich jedoch um einen Fahrassistent, also ein *SAE-Level-2*-System. Es steuert das Fahrzeug nicht autonom, sondern assistiert nur dem Fahrer. Die Behörde hob hervor, dass es zwischen den Jahren 2021 und 2022 bereits 130 Unfälle mit autonomen Fahrzeugen gegeben hatte. Die große Mehrheit der Unfälle waren Sachschäden ohne Verletzungen. Lediglich ein Unfall hatte schwere Verletzungen zur Folge. Die geringe Anzahl der Unfälle kann aber auf die wenigen autonomen Fahrzeuge, die zurzeit auf der Straße getestet werden, zurückgeführt werden.

15.2 Einführung in den Markov Decision-Prozess

Ein Markov-Prozess bildet die Umwelt mit Zuständen und Zustandsvektoren ab, deren Elemente Wahrscheinlichkeiten sind. Eine Person oder eine Elektronik (im Folgenden Akteur genannt) kann einen Zustandswechsel veranlassen. So kann ein Mar-

kov-Prozess unter Umständen sehr komplex werden, wenn z. B. ein Szenario auf einer Autobahn als Modell abgebildet werden soll. Der Markov-Prozess kann mit Aktionen erweitert werden, die ein Akteur ausüben kann. Unter anderem kann er über diese Aktionen in einen weiteren Markov-Prozess überführt werden.

Abbildung 15.2 zeigt ein Zustandsmodell, bei dem der Akteur eine Aktion (*a* oder *b*) ausübt und so das Modell von einem Ausgangszustand in einen Folgezustand überführt. Die Abbildung zeigt im linken Zustand ein parkendes Auto. Über die Aktion *a* geht das Auto in einen fahrenden Zustand über. Bei Ausübung der Aktion *a* bleibt das Fahrzeug im fahrenden Zustand, oder der Prozess geht wieder in den parkenden Zustand. Die Aktion *a* kann als *Fahren* bezeichnet werden. Der Akteur hat aber die Wahl, im fahrenden Zustand die Aktion *b* zu wählen. Dies bewirkt, dass das Modell mit einer Wahrscheinlichkeit in den *Wartungs*-Zustand überführt wird oder mit der komplementären Wahrscheinlichkeit in dem fahrenden Zustand bleibt (möglicherweise war zu diesem Zeitpunkt keine Inspektionsstelle in der Nähe). Das Fahrzeug bleibt im *Wartungs*-Zustand, bis der Akteur die Aktion *b* wählt, damit es in den fahrenden Zustand kommt. Eine Aktion *a* hat dabei keine Auswirkung. Wenn das Inspektionspersonal zur Entscheidung kommt, das Fahrzeug auszumustern, geht der Prozess in den *Verschrottungs*-Zustand über.

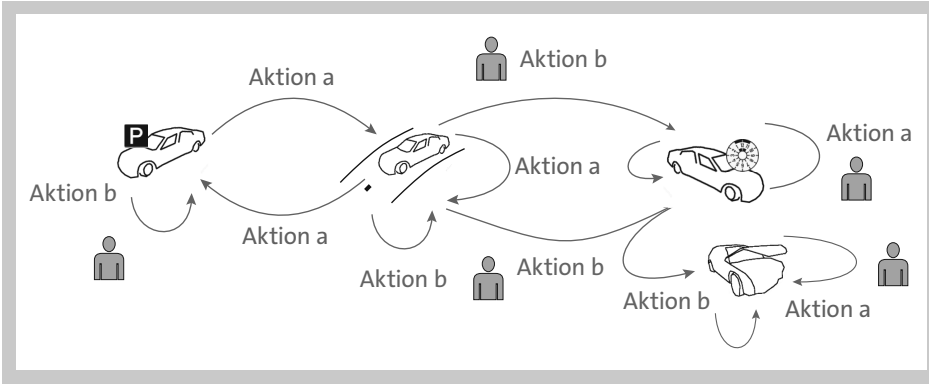


Abbildung 15.2 Zustandsmodell mit Aktionen

Sie werden sich die Frage stellen, wann sich der Akteur für Aktion *a* oder Aktion *b* entscheiden soll. Die Antwort ist: Belohnungen helfen dem Akteur, Entscheidungen zu treffen. Einer Aktion, die einen Wechsel vom einen Ausgangszustand zu einem Folgezustand führt, wird eine Belohnung ausgezahlt. Durch eine Sequenz von Aktionen können sich Belohnungen auf eine Gesamtbelohnung aufsummieren. Zum Beispiel hat in Abbildung 15.2 der Akteur stets die Motivation, das Fahrzeug zu fahren, denn nur dann hat er Einnahmen (z. B. durch den Transport von Waren). Ein parkendes Auto erzeugt beim Akteur Verluste, z. B. durch Parkgebühren und Abwertung des Fahrzeugs wegen Veraltung. Befindet sich das Modell im *Wartungs*-Zustand, hat der

Akteur eine Motivation, die Inspektion schnell abzuschließen, denn das Fahrzeug kann wieder als neuwertig angesehen werden (das Auto erfährt dadurch eine Wertsteigerung).

Abbildung 15.3 zeigt in abstrakter Weise die Beziehung zwischen dem Akteur und dem Zustandsmodell. Über Aktionen kann dieser Einfluss auf Zustandswechsel nehmen und erhält dafür eine Belohnung. Ziel für den Akteur ist die Optimierung der Gesamtbelohnung über eine Sequenz von Aktionen.

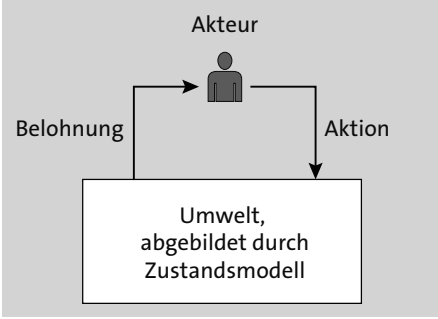


Abbildung 15.3 Akteur und Umwelt

Die hier vorgestellte Theorie ist eine Erweiterung des Markov-Prozesses und wird *Markov Decicion-Prozess* (kurz MDP) genannt. Die Theorie sind Grundlagen für einige Bereiche der künstlichen Intelligenz, z. B. *Reinforcement Learning*. In Artikel [88] wurde gezeigt, wie das Atari-Spiel *Breakout* durch einen Computer-Akteur erlernt wurde. Die Grundlagen dieses Kapitels werden unter anderem im Buch [89] beschrieben und sind Stoff in vielen Vorlesungen der künstlichen Intelligenz.

15.3 Grundlagen des MDP

In Abschnitt 14.1 wurde bereits die Definition in Formel [14.1] für den Markov-Prozess angegeben. Diese Formel lässt die Aktionen des Akteurs außer Acht. Formel [15.1] berücksichtigt die Aktionen. In Worten ist ein Prozess dann ein MPD, wenn der aktuelle Zustand und die Aktion den Folgezustand bestimmt. Zustände, die in der Vergangenheit liegen, haben keinen Einfluss auf zukünftige Entscheidungen.

[15.1]
$$Pr\{S(t_s) = s_j \mid S(t_{s-1}) = s_i, A(t_{s-1}) = u_i, \ldots, S(t_0) = s_l, A(t_0) = u_l\} = Pr\{S(t_s) = s_j \mid S(t_{s-1}) = s_i, A(t_{s-1}) = u_i\}$$

Sie stellen sich vielleicht die Frage, wie die Vergangenheit trotzdem modelliert werden kann, da diese in vielen Fällen einen Einfluss auf zukünftige Entscheidungen hat. Das kann durch einen Modellierungsansatz wie den in Abbildung 15.4 gelöst werden.

Auf einen Blick

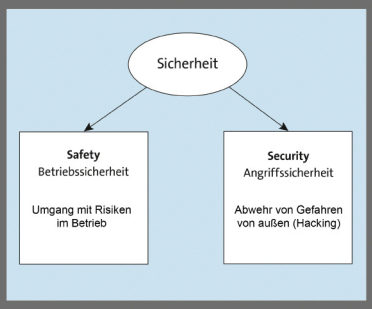
Inhalt

Safety Engineering

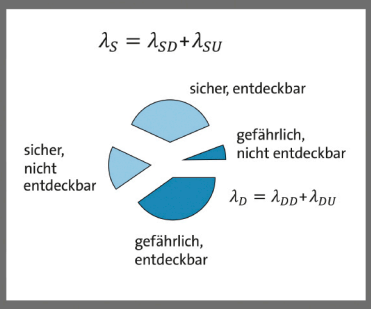
»Safety doesn't happen by accident«

Sichere und zuverlässige Systeme erstellen

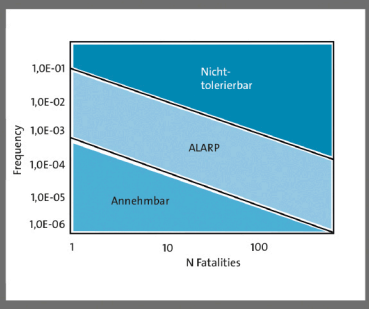
Dieses Lehrbuch vermittelt Ihnen die Grundlagen sicherer Softwareentwicklung und die Prinzipien der Betriebssicherheit in der Hardwareentwicklung. Sie lernen, wie Sie die Risiken komplexer Systeme einschätzen, Fehlerbaumanalysen durchführen, Risikographen gestalten und die essenziellen Methoden der sicheren Systementwicklung beherrschen.



Safety und Security



Fehler erkennen und verstehen



Modelle und Simulationen

Gute Programmierung

Unit-Tests, Code Reviews, defensive Programmierung: Schon mit einfachen Schritten können Sie die Qualität und Sicherheit Ihrer Software spürbar erhöhen. Hier erfahren Sie, was wirklich einen Mehrwert bietet und worauf Sie achten müssen.

Für Studium und Beruf

Von der Risikoidentifikation bis hin zu fortgeschrittenen Themen wie Fehlerbaum- und Markov-Analysen erhalten Sie einen Überblick über die Techniken der funktionalen Sicherheit. Fallbeispiele erläutern Sicherheitsprobleme, damit Sie aus den Fehlern beim Design und der Umsetzung sicherheitskritischer Systeme lernen können.

Sicher, robust und zuverlässig

Je komplexer Systeme werden, desto anfälliger sind sie für Ausfälle und Fehler. Dieses Lehrbuch zeigt Ihnen, mit welchen Methoden Sie systemrelevante Risiken qualitativ und quantitativ abschätzen können, um fehlerarme und wartbare Systeme zu entwickeln.

Aus dem Inhalt

- Normen und Sicherheitsrichtlinien
- Sicherheit in der Software- und Hardwareentwicklung
- Fehler analysieren und verstehen
- Verfügbarkeit, Schaden, Risiko, Fehlertoleranz
- Kenngrößen: Zuverlässigkeit, Ausfallrate, Lebensdauer
- Gefahrenanalyse
- Fehlerbaumanalyse
- Risikograph
- Layer of Protection Analysis
- Zuverlässigkeitsblockdiagramm
- Markov-Decision-Prozess
- Binary-Decision-Diagrams

Prof. Dr. Derk Rembold leitet das Institut für Echtzeitsysteme und Softwaretechnik an der Hochschule Albstadt-Sigmaringen. Er beschäftigt sich mit Betriebssicherheit, der qualitativen Überwachung von Produkten und Software in automatisierten Prozessen.

Rheinwerk Computing

Gedruckt in Deutschland
Papier aus nachhaltiger Waldwirtschaft
Mineralölfreie Druckfarben

ISBN 978-3-8362-XXXX-X

€ 59,90 [D] € 61,60 [A]
Informatik
ISBN 978-3-8362-9928-2