Der Einsatz quantitativer Sicherheitsanalysen für den risikobasierten Test eingebetteter Systeme[1]

Heiko Stallbaum, Andreas Metzger, Klaus Pohl

Software Systems Engineering
Universität Duisburg-Essen
Schützenbahn 70, 45127 Essen
{heiko.stallbaum|andreas.metzger|klaus.pohl}@sse.uni-due.de

Abstract: Um die oft hohen Sicherheitsanforderungen eingebetteter Systeme (im Sinne von Safety) zu gewährleisten, werden bei ihrer Entwicklung sowohl dynamische Testverfahren als auch statische Sicherheitsanalysen eingesetzt. Für den Test bieten sich insbesondere risikobasierte Ansätze an, da sie sicherstellen, dass solche Teile bzw. Anforderungen des Testobjektes früher und intensiver getestet werden, deren Versagen bzw. Nichterfüllung zu einem hohen Produktrisiko führen, also z.B. zu sehr hohen Schäden für Mensch und Umwelt. Der Einsatz eines risikobasierten Testansatzes erfordert jedoch aufgrund der notwendigen Produktrisikobewertungen hohen Zusatzaufwand. In diesem Beitrag wird ein Ansatz dargestellt, der risikobasierte Testtechniken mit quantitativen Sicherheitsanalysen kombiniert. Hierdurch werden Synergieeffekte erzielt, die eine separate Produktrisikobewertung obsolet machen und den entsprechenden Zusatzaufwand für den risikobasierten Test vermeiden. Die Anwendbarkeit des Ansatzes wird anhand eines Beispieles aus dem industriellen Umfeld demonstriert.

1 Einleitung und Motivation

Eingebettete Systeme unterliegen oftmals hohen Sicherheitsanforderungen (im Sinne von Safety). Solche sicherheitsgerichteten, eingebetteten Systeme, wie sie z.B. im Automobilbau oder der Luft- und Raumfahrt verwendet werden, bedürfen vor dem Einsatz der Zertifizierung, da sie zu Gefährdungen für Mensch und Umwelt führen können. Im Rahmen der Zertifizierung ist der Nachweis zu erbringen, dass schon im Entwicklungsprozess geeignete Maßnahmen ergriffen wurden, die sicherstellen, dass alle Sicherheitsanforderungen eingehalten werden. Zu diesen Maßnahmen zählen insbesondere dynamische Testverfahren und statische Sicherheitsanalysen (vgl. [Tr99], [Li02]).

Das primäre Ziel von Sicherheitsanalysen besteht darin, potenzielle Sicherheitsprobleme frühzeitig, vorausschauend aufzuspüren und zu gewichten (vgl. [Li02]). Für die Zertifizierung sicherheitsgerichteter, eingebetteter Systeme fordern Sicherheitsnormen die Anwendung ausgereifter, normierter Techniken der Sicherheitsanalyse (z.B. FMEA, FMECA, FTA, etc.) um die Sicherheit eines Systems nachzuweisen (vgl. z.B. [SAE96],

[1] Dieser Arbeit wurde mit Mitteln des Bundesministeriums für Bildung und Forschung (BMBF) unter dem Förderkennzeichen O1IS08045V (Softwareplattform Embedded Systems 2020, SPES 2020) gefördert.

[DIN03], [ISO09]). Hierbei wird zunächst ein Systemmodell erstellt und dieses dann systematisch hinsichtlich der Risiken analysiert. Dabei kommen sowohl qualitative als auch quantitative Analysen zum Einsatz. Sicherheitsanalysen bieten den Vorteil generelle Aussagen zur Sicherheit eines analysierten Systems zu liefern, da sie nicht wie dynamische Tests auf stichprobenartige Testfälle beschränkt sind. Die Sicherheitsanalyse hat jedoch den Nachteil, dass die Verlässlichkeit der gewonnenen Aussagen stark von der Übereinstimmung zwischen dem Systemmodell und der Realität abhängt (vgl. [Li02]). Gründe für eine mangelnde Übereinstimmung zwischen Modell und Realität können aus fehlerhafter Modellierung oder einer zu starke Abstraktion vom System mit entsprechendem Informationsverlust resultieren (vgl. [Tr99]).

Testen ist die vorherrschende Qualitätssicherungsmaßnahme in der Praxis. Tests haben den Vorteil, dass die zu testende Software in ihrer realen Betriebsumgebung ausgeführt werden kann, was die Aufdeckung von Fehlern oder Sicherheitsproblemen gestattet, die durch Sicherheitsanalysen kaum erkennbar sind (vgl. [Li02]). Durch Testen kann man Software jedoch nur stichprobenartig prüfen, da ein erschöpfender Test aufgrund der sehr großen Menge möglicher Eingaben nur in trivialen Fällen möglich ist. Ein systematisches Vorgehen beim Test kann sicherstellen, dass mit den gewählten Stichproben die wesentlichen Aspekte eines Softwaresystems getestet werden. Bei sicherheitsgerichteten, eingebetteten Systemen ist es daher wesentlich, dass diejenigen Teile bzw. Anforderungen getestet werden, deren Versagen bzw. Nichterfüllung zu einem hohen Produktrisiko führen, also z.B. zu sehr hohen Schäden für Mensch und Umwelt. Hier bieten sich risikobasierte Testansätze an, da diese sicherstellen, dass solche „riskanten“ Teile bzw. Anforderungen früher und intensiver getestet werden und damit die Wahrscheinlichkeit erhöht wird, dass kritische Fehler frühzeitig aufgedeckt werden.

1.1 Problemstellung

Der Einsatz eines risikobasierten Testansatzes erfordert eine Produktrisikobewertung. Deren Güte hat maßgeblichen Einfluss auf die Allokation des Testaufwandes und somit auf die Wahrscheinlichkeit des frühzeitigen Aufdeckens kritischer Fehler. Dementsprechend hoch sind die Anforderungen an die Risikobewertung: Sie muss systematisch alle Produktrisiken identifizieren und bewerten. Insbesondere in iterativen Entwicklungsprozessen mit veränderlichen Produktrisiken müssen Produktrisikobewertungen kontinuierlich aktualisiert werden, da veraltete Bewertungen zu einer falschen Allokation des Testaufwandes führen können (vgl. [Bo88], [Ba99], [Pi04]). Dieser Beitrag adressiert das Problem, dass beim risikobasierten Test hoher Zusatzaufwand für kontinuierlich aktualisierte und qualitativ gute Produktrisikobewertungen erforderlich wird.

1.2 Lösungsidee

Sowohl bei Sicherheitsanalysen als auch bei risikobasierten Tests werden Produktrisiken explizit berücksichtigt. In diesem Beitrag wird ein Ansatz vorgestellt, der risikobasierte Testtechniken mit quantitativen Sicherheitsanalysen kombiniert und hierdurch Synergieeffekte realisiert, durch die der in Abschnitt 1.1 dargestellte Zusatzaufwand für den risikobasierten Test vermieden wird. Dies wird erreicht, indem Produktrisikobewertungen

aus quantitativen Sicherheitsanalysen beim anforderungsbasierten Systemtest sicherheitsgerichteter, eingebetteter Systeme zur modell- und risikobasierten Testfallgenerierung und -priorisierung verwendet werden. Gegenüber existierenden risikobasierten Testansätzen bietet der vorgeschlagene Ansatz folgende Vorteile:

1. *Technik für Produktrisikobewertung*: Anstelle einer ansatzspezifischen Technik zur Produktrisikobewertung wird eine ausgereifte, normierte Technik der quantitativen Sicherheitsanalyse verwendet, deren Anwendung von Sicherheitsnormen für die Zertifizierung sicherheitsgerichteter, eingebetteter Systeme empfohlen wird. Dieses Vorgehen verspricht qualitativ gute und normenkonforme Produktrisikobewertungen sowie einen einfachen Transfer des vorgestellten Ansatzes in die Praxis.
2. *Aufwand für Produktrisikobewertung*: Separate Produktrisikobewertungen für den risikobasierten Test werden obsolet, da auf die Produktrisikobewertungen aus der Sicherheitsanalyse zurückgegriffen wird. Dieses Vorgehen verspricht eine Aufwandsvermeidung gegenüber Ansätzen mit separater Produktrisikobewertung.
3. *Aktualisierung von Produktrisikobewertungen:* Typischerweise wird eine prozessbegleitende Durchführung von Sicherheitsanalysen während der Entwicklung gefordert (vgl. [DIN03], [DIN06], [DT08]). Dementsprechend kann im vorgeschlagenen Ansatz auf kontinuierlich aktualisierte Risikobewertungen zurückgegriffen werden.

2 Verwandte Arbeiten

Existierende risikobasierte Testansätze geben entweder keine Anleitung zur Ausgestaltung der Produktrisikobewertung (vgl. [CP03]) oder schlagen jeweils ansatzspezifische Produktrisikobewertungen vor. Hierbei identifizieren und bewerten Experten oder Stakeholder Produktrisiken in Workshops z.B. auf Basis von Checklisten, Qualitätsmerkmalen und Priorisierungsmodellen (vgl. [Pi04]), gewichteter Risikofaktoren (vgl. [SW05], [Sr08]) oder mittels Heuristiken (vgl. [Ba99]). Einige Ansätze schlagen Metriken vor, um den durch Produktrisikobewertungen resultierenden Zusatzsaufwand für den risikobasierten Test zumindest zu reduzieren (vgl. [Ro99], [SM07]).

Liggesmeyer unterstreicht in [Li02] die bedeutende Rolle des Testens bei der Prüfung sicherheitsgerichteter, eingebetteter Systeme. Sicherheitsanalysen sollten ergänzend hierzu durchgeführt werden. Unter anderem wird vorgeschlagen die FMECA (Failure Mode, Effects and Criticality Analysis, [DIN06]) zumindest für das zu entwickelnde System und die kritischen Komponenten durchzuführen. Dabei erkannte schwerwiegende Risiken sollten mittels FTA (Fault Tree Analysis, [DIN07]) genauer quantitativ bestimmt werden. Die Integration oder Verwendung von (quantitativen) Sicherheitsanalysen in (risikobasierten) Testansätzen wird in [Li02] jedoch nicht adressiert.

Tracey et al. schlagen in [Tr99] die Integration eines Ansatzes zur automatischen Testdatengeneration in die Sicherheitsanalyse vor. Zentraler Bestandteil des Ansatzes ist eine Kostenfunktion, mit der Testdaten gesucht werden, die beim Test mit hoher Wahrscheinlichkeit zu einer Verletzung einer Sicherheitsanforderung führen. Diese Testdaten konzentrieren sich auf Teile des Softwaresystems, deren Sicherheit noch nicht hinreichend analysiert wurde. Bei der Sicherheitsanalyse mittels FTA werden die Testergebnisse

dann verwendet, um ein adressiertes Sicherheitsproblem nachzuweisen oder Vertrauen darin zu schaffen, dass es nicht vorliegt und die entsprechende Sicherheitsanforderung damit erfüllt ist. Dieses Vorgehen soll zu einer Aufwandsreduktion bei der Sicherheitsanalyse führen und das Vertrauen in den Sicherheitsnachweis stärken. Der Ansatz von Tracey et al. nutzt somit Testergebnisse für die Sicherheitsanalyse, beschreibt jedoch nicht, wie in umgekehrter Richtung (quantitative) Sicherheitsanalysen für den (risikobasierten) Test genutzt werden können.

In den Vorarbeiten der Autoren in [Ba08] wurde ein Ansatz zur risikobasierten Ableitung und Priorisierung von Testfällen für den modellbasierten Systemtest vorgeschlagen. Er erlaubt eine weitgehend automatische Generierung und Priorisierung von Testfällen auf Basis eines zustandsbasierten Testmodells ausgehend von risikobewerteten Anwendungsfällen. Zimmermann et al. stellen in [Zi09] aufbauend auf [Ba08] einen Ansatz zur automatisierten, risikobasierten Testfallgenerierung auf Basis statistischer Testmodelle für sicherheitskritische Systeme vor. Zur Risikobewertung sollen Sicherheitsanalysetechniken wie HAZOP (Hazard and Operability Study, [Ea92]), FMEA oder FTA eingesetzt werden. Es werden aber keine Hinweise gegeben, wie die Integration dieser Sicherheitsanalysen in die risikobasierte Testfallgenerierung erfolgen soll.

3 Ansatz für den risikobasierten Test unter Verwendung quantitativer Sicherheitsanalysen

Der in diesem Beitrag vorgeschlagene Ansatz adressiert zwei Phasen des modell- und risikobasierten Tests: (1) Die manuelle Testmodellerstellung sowie (2) die automatische Testfallableitung und -priorisierung (Abbildung 1). Im Gegensatz zu existierenden risikobasierten Testansätzen werden bei der Testmodellerstellung Produktrisikobewertungen aus quantitativen Sicherheitsanalysen verwendet (Abschnitt 3.1). Durch dieses Vorgehen kann der in Abschnitt 1.1 dargestellte Zusatzaufwand für den risikobasierten Test vermieden werden. Die im Rahmen der Sicherheitsanalyse quantifizierten Produktrisiken werden dann zur automatischen Testfallableitung und Priorisierung genutzt (Abschnitt 3.2). Die Aktivitäten der Phasen werden im Folgenden näher beschrieben.

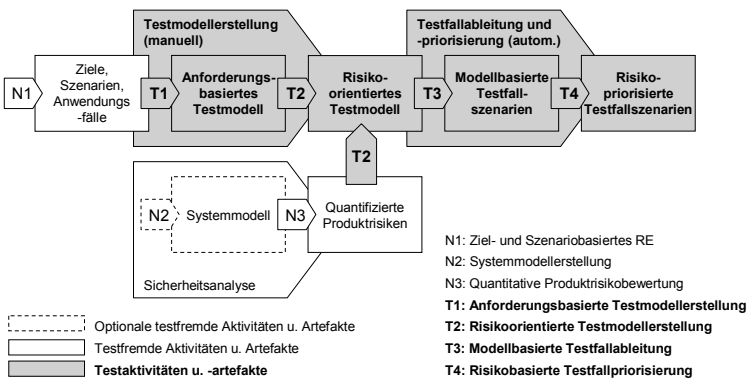

Abbildung 1: Anforderungs- und modellbasierter Ansatz für den risikobasierten Systemtest

3.1 Manuelle Testmodellerstellung unter Verwendung von Sicherheitsanalysen

In der ersten Phase des vorgeschlagenen Ansatzes wird ein sogenanntes risikoorientiertes Testmodell erstellt. Zur Erstellung dieses Modells sind quantifizierte Produktrisiken erforderlich. Bei der Entwicklung eingebetteter Systeme werden solche Risiken durch ausgereifte, normierte Techniken der quantitativen Sicherheitsanalyse ermittelt. Da Sicherheitsanalysen typischerweise prozessbegleitend durchgeführt werden, kann auf kontinuierlich aktualisierte Risikobewertungen zurückgegriffen werden. In diesem Abschnitt wird gezeigt, wie Aktivitäten der Sicherheitsanalyse und der Testmodellerstellung kombiniert werden können, damit separate, ansatzspezifische Produktrisikobewertungen für den risikobasierten Test sowie deren Aktualisierungen überflüssig werden.

Aktivität N1 (Ziel- und Szenariobasiertes Requirements Engineering): Anforderungen in Form von Anwendungsfällen bilden die Grundlage für den vorgeschlagenen Ansatz. Anwendungsfälle eignen sich zur Dokumentation von Anforderungen und als Grundlage für Sicherheitsanalysen (vgl. [AK01]). Sie aggregieren mehrere zu einem Ziel assoziierte Szenarien (vgl. [Po08]). Der vorgeschlagene Ansatz basiert somit auf Zielen, Szenarien und Anwendungsfällen, welche zentrale Ergebnisse eines ziel- und szenariobasierten Requirements Engineering darstellen.

Aktivität T1 (Anforderungsbasierte Testmodellerstellung): Als Zwischenschritt zum risikoorientierten Testmodell wird aus Anwendungsfällen zunächst ein anforderungsbasiertes Testmodell erstellt. In [Re05] und [Po08] wird mit ScenTED eine Technik dargestellt, mit der u.a. ausgehend von Anwendungsfällen und deren Szenarien das Gesamtverhalten eines Softwaresystems in einem Testmodell beschrieben werden kann. Die ScenTED-Technik verwendet zur Dokumentation des Testmodells UML Aktivitätsdiagramme. Ein wesentlicher Nutzen, der bei ScenTED durch die systematische Testmodellerstellung aus Anwendungsfällen erzielt wird, ist die Nachvollziehbarkeit (engl. Traceability) zwischen den in Anwendungsfällen aggregierten Zielen und Szenarien und Modellelementen des Testmodells (vgl. [Po08]). Im vorgeschlagenen Ansatz erfolgt die anforderungsbasierte Testmodellerstellung entsprechend der ScenTED-Technik.

Aktivität N2 (Systemmodellerstellung): Bei der Sicherheitsanalyse muss das zu analysierende System zunächst modelliert werden. Da das anforderungsbasierte Testmodell das Gesamtverhalten des zu testenden Softwaresystems darstellt, kann es bei der Sicherheitsanalyse als Systemmodell verwendet werden. In diesem Fall ergibt sich neben dem in diesem Beitrag adressierten Synergieeffekt für den risikobasierten Systemtest ein zusätzlicher Vorteil für die Sicherheitsanalyse: Durch Rückgriff auf das anforderungsbasierte Testmodell kann auf eine separate Systemmodellierung für die Sicherheitsanalyse verzichtet werden. Der vorgeschlagene Ansatz erlaubt auch die Verwendung quantitativer Sicherheitsanalysen, die auf anderen Systemmodellen als dem anforderungsbasierten Testmodell beruhen. In diesem Fall ist die Nachvollziehbarkeit zwischen den Modellelementen des verwendeten Systemmodells und den in Anwendungsfällen aggregierten Zielen und Szenarien notwendig, da sich die bei der Sicherheitsanalyse gewonnenen quantifizierten Produktrisiken auf Elemente des Systemmodells beziehen und daher bei der risikoorientierten Testmodellerstellung (Aktivität T2) zu den Elementen des Testmodells assoziiert werden müssen.

Aktivität N3 (Quantitative Produktrisikobewertung): Im Rahmen der quantitativen Sicherheitsanalyse werden Produktrisiken nunmehr auf Grundlage des Systemmodells und in Kenntnis der in Anwendungsfällen aggregierten Ziele und Szenarien systematisch analysiert und quantifiziert. In diesem Beitrag schlagen wir hierfür mit FMECA eine ausgereifte und normierte Technik der quantitativen Sicherheitsanalyse vor. Bei FMECA handelt es sich im Grunde um eine auf Risiken fokussierte Inspektionstechnik (vgl. [Li02]). Während der FMECA werden zunächst mögliche Fehlerarten identifiziert, hierfür dann u.a. Fehlerursachen und -auswirkungen analysiert sowie schließlich Risiken bestimmt, die auf einer Ordinalskala in eine Reihenfolge gebracht werden. Für die Entwicklung sicherheitsgerichteter, eingebetteter Systeme existieren zahlreiche Sicherheitsnormen, welche die Anwendung der FMECA für die Zertifizierung empfehlen (vgl. z.B. [SAE96], [DIN03], [ISO09]). Neben FMECA existieren weitere Techniken der Sicherheitsanalyse. Manche von ihnen sind stärker mathematisch fundiert, jedoch auch komplexer und weniger intuitiv anwendbar (Markov-Ketten, Petrinetze, etc.). Sie sind in der Praxis entsprechend weniger verbreitet und akzeptiert, können intuitiv anwendbare Techniken wie FMECA jedoch ergänzen (vgl. [DT08]). Die Wahl von FMECA verspricht somit qualitativ gute und normenkonforme Produktrisikobewertungen sowie einen einfachen Transfer des vorgestellten Ansatzes in die Praxis.

Aktivität T2 (Risikoorientierte Testmodellerstellung): Die im Rahmen der Sicherheitsanalyse gewonnenen quantifizierten Produktrisiken müssen explizit dokumentiert und in einem ausreichenden Umfang formalisiert werden. Im vorgeschlagenen Ansatz erfolgt die Dokumentation in einem zentralen intermediären Artefakt, dem risikoorientierten Testmodell. Es bildet die Grundlage für die weiteren Testaktivitäten wie etwa der modellbasierten Testfallableitung (Aktivität T3) oder der risikobasierten Testfallpriorisierung (Aktivität T4). Beide Aktivitäten können auf dieser Grundlage vollständig automatisiert durchgeführt werden. Das risikoorientierte Testmodell erweitert das anforderungsbasierte Testmodell um eine explizite Dokumentation der quantifizierten Produktrisiken. Als formale Notation wird aufbauend auf vorherigen Arbeiten das UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms (UML QoS-Profil, [OMG08]) genutzt und eine QoS-Characteristic Risiko definiert (vgl. [Ba08]). Jede Aktualisierung einer Produktrisikobewertung aufgrund der prozessbegleitenden Durchführung von Sicherheitsanalysen erfordert lediglich die Aktualisierung des entsprechenden Produktrisikos im Testmodell.

3.2 Automatische Testfallableitung und -priorisierung

Das risikoorientierte Testmodell bildet die Grundlage für die zweite Phase des modell- und risikobasierten Tests: Die automatische Testfallableitung und -priorisierung. In [St08] haben wir mit RiteDAP eine Technik zur modellbasierten Testfallableitung und risikobasierten Testfallpriorisierung vorgestellt, die in Zusammenarbeit mit Industriepartnern im Rahmen des Forschungsprojektes ranTEST entwickelt wurde. Die RiteDAP-Technik verwendet zur Ableitung der Testfälle ein Testmodell, welches dem in Abschnitt 3.1 beschriebenen risikoorientierten Testmodell des vorgeschlagenen Ansatzes entspricht. Ein wesentlicher Nutzen, der bei RiteDAP durch die modellbasierte Testfallableitung und die explizite Dokumentation quantifizierter Produktrisiken im Testmodell

erzielt wird, ist der hohe Automatisierungsgrad (vgl. [St08]). Die umfassende Automatisierung der RiteDAP-Technik ermöglicht eine kosteneffiziente und zuverlässige Testfallableitung und -priorisierung beim risikobasierten Test, selbst wenn kontinuierliche Produktrisikobewertungen aufgrund veränderlicher Produktrisiken erforderlich sind (vgl. Abschnitt 1.1). Im vorgeschlagenen Ansatz erfolgt die Testfallableitung und -priorisierung entsprechend der RiteDAP-Technik.

Aktivität T3 (Modellbasierte Testfallableitung): Zur modellbasierten Ableitung von Testfällen werden bei RiteDAP automatisch Pfade durch das Testmodell bestimmt. Diese Pfade stellen Testfallszenarien dar, die von Testdaten abstrahieren. Die Auswahl der Testfallszenarien aus der Menge aller möglichen Testfallszenarien wird bei RiteDAP mit Hilfe des Boundary-Interior-Pfadüberdeckungskriteriums gesteuert. Testfallszenarien werden so abgeleitet, dass eine 100%-ige Abdeckung aller Pfade unter Berücksichtigung des Boundary-Interior-Kriteriums im risikoorientierten Testmodell erreicht wird.

Aktivität T4 (Risikobasierte Testfallpriorisierung): Die im risikoorientierten Testmodell annotierten quantifizierten Produktrisiken werden bei RiteDAP zur Priorisierung der abgeleiteten Testfallszenariomenge herangezogen. Hierzu wird für jedes Testfallszenario ein Risikowert errechnet. Der Risikowert ergibt sich aus der Summe der Risikowerte aller Aktivitäten, die das Testfallszenario im risikoorientierten Testmodell abdeckt (vgl. [CP03]). Je nach gewählter Priorisierungsstrategie werden hierbei schon abgedeckte Aktionen in die Summierung einbezogen (Total Risk Score Prioritization, TRSP) oder nicht (Additional Risk Score Prioritization, ARSP). Beide Priorisierungsstrategien sind Adaptionen grundlegender Priorisierungsstrategien aus [RE03] für den und modell- und risikobasierten Test auf Basis von Aktivitätsdiagrammen und werden in [St08] detailliert beschrieben.

4. Anwendung des Ansatzes

Zur Demonstration der Anwendbarkeit des Ansatzes wurde mit der adaptiven Fahrgeschwindigkeitsregelung bei Automobilen (engl. Adaptive Cruise Control, ACC) ein bekanntes sicherheitsgerichtetes, eingebettetes System aus dem industriellen Umfeld als Anwendungsbeispiel gewählt. Das ACC-System ist eine Weiterentwicklung der konventionellen Fahrgeschwindigkeitsregelung, die bei der Regelung den Abstand zu einem vorausfahrenden Fahrzeug als zusätzliche Stellgröße einbezieht. Das ACC-System reagiert auf langsamer vorausfahrende oder einscherende Fahrzeuge mit einer Reduzierung der eigenen Fahrgeschwindigkeit, so dass der vorgeschriebene Sicherheitsabstand zum vorausfahrenden Fahrzeug nicht unterschritten wird. Das im Folgenden beschriebene System basiert auf der Spezifikation des ACC-Systems nach [Wi03]. Es enthält auf Gesamtsystemebene insgesamt 16 Anwendungsfälle. Mit dem Ziel, die Anschaulichkeit und Verständlichkeit der Anwendbarkeitsdemonstration zu erhöhen, wurden technische Details des ACC-Systems zum Teil stark vereinfacht und die Betrachtung des ACC-Systems wurde auf einen einzelnen beispielhaften Anwendungsfall begrenzt. Im Folgenden werden die Ergebnisse der Aktivitäten des Ansatzes (s. Abschnitt 3) für diesen beispielhaften Anwendungsfall dargestellt. Eine weitergehende Validierung ist Gegenstand zukünftiger Arbeiten im Rahmen der nationalen Innovationsallianz SPES 2020.

Aktivität N1 (Ziel- und Szenariobasiertes Requirements Engineering): Als Ergebnis des ziel- und szenariobasierten Requirements Engineering dokumentiert Tabelle 1 den betrachteten Anwendungsfall mit Hilfe einer Schablone (vgl. [Po08]). Der Anwendungsfall gruppiert jene Szenarien, die das ACC-System ausführt, um das Ziel „Abstand zum vorausfahrenden Fahrzeug vergrößern um Sicherheitsabstand einzuhalten" zu erreichen.

Abschnitt	Inhalt
Name	Geschwindigkeitsverringerung bei vorausfahrendem Fahrzeug innerhalb des Sicherheitsabstandes
Ein- und Ausgabevariablen	▪ Input: Abstand zum vorausfahrenden Fahrzeug von Abstandssensoren des ACC-Systems, Ist-Geschwindigkeit von Tachometer ▪ Output: Signal zur Geschwindigkeitsverringerung an Motorsteuerung, ggf. Bremssignal an Bremssystem
Ziel(e)	Abstand zum vorausfahrenden Fahrzeug vergrößern um Sicherheitsabstand einzuhalten
Primärer Akteur	Vorausfahrendes Fahrzeug
Andere Akteure	Tachometer, Motorsteuerung, Bremssystem
Vorbedingungen	ACC-System wurde mit Soll-Geschwindigkeit aktiviert
Nachbedingungen	Geschwindigkeit wurde auf Folgegeschwindigkeit verringert
Auslöser	Abstand zum vorausfahrenden Fahrzeug ist zu gering
Hauptszenario	1. ACC-System errechnet Folgegeschwindigkeit auf Grundlage von Ist-Geschwindigkeit und Abstand zum vorausfahrenden Fahrzeug 2. ACC-System sendet Signal zur Geschwindigkeitsverringerung auf Folgegeschwindigkeit an Motorsteuerung
Alternativszenarien	2a1. ACC-System stellt fest, dass Geschwindigkeitsverringerung durch Motorsteuerung nicht ausreichend ist 2a2. ACC-System sendet Bremssignal auf Folgegeschwindigkeit an Bremssystem

Tabelle 1: Betrachteter Anwendungsfall des ACC-Systems

Aktivität T1 (Anforderungsbasierte Testmodellerstellung): Auf Basis der Anwendungsfälle eines Systems und insbesondere auf Basis der in ihnen enthaltenen Szenarien wird mittels der ScenTED-Technik ein anforderungsbasiertes Testmodell erstellt, welches als Aktivitätsdiagramm dokumentiert wird. Mit ScenTED wird die Nachvollziehbarkeit zwischen den in Anwendungsfällen aggregierten Zielen und Szenarien und Modellelementen des Testmodells gewährleistet. Abbildung 2 stellt den Ausschnitt des anforderungsbasierten Testmodells für das ACC-System dar, welcher das Verhalten des betrachteten Anwendungsfalls modelliert.

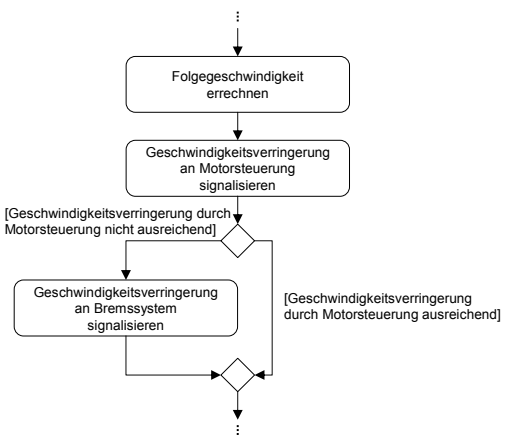

Abbildung 2: Ausschnitt aus anforderungsbasiertem Testmodell zum betrachteten Anwendungsfall

Aktivität N2 (Systemmodellerstellung): Im Rahmen der quantitativen Sicherheitsanalyse kann das anforderungsbasierte Testmodell als Systemmodell verwendet werden. Wir

demonstrieren im Folgenden die Verwendung dieses Modells bei der quantitativen Sicherheitsanalyse, da hierdurch eine separate Systemmodellierung obsolet wird.

Aktivität N3 (Quantitative Produktrisikobewertung): Sicherheitsanalysen (z.B. durch FMECA) werden von Sicherheitsstandards für eingebettete Systemen im Automobil gefordert (vgl. [ISO09]). Im Rahmen der quantitativen Sicherheitsanalyse mit FMECA werden mögliche Fehlerarten zum ACC-System identifiziert, zugehörige Fehlerursachen und -auswirkungen analysiert sowie Produktrisiken quantifiziert. Die Ergebnisse der FMECA werden in FMECA-Arbeitsblättern dokumentiert. In Tabelle 2 ist ein Teil des Analyseergebnisses der FMECA für den betrachteten Anwendungsfall in einem solchen FMECA-Arbeitsblatt dargestellt. Die Risikoprioritätszahl RPN dient zur Quantifizierung der identifizierten Produktrisiken und wurde im Beispiel als Produkt der Schwere der Fehlerauswirkung und des Auftretens bzw. der Auftretenswahrscheinlichkeit der Fehlerursachen berechnet.[2] Hierbei wurden die in der Automobilindustrie weit verbreiteten Skalen für Schwere und Auftreten von 1 bis 10 (sehr gering = „1“ bis sehr hoch = „10“) verwendet (vgl. [DIN06]).

Tabelle 2: Ausschnitt aus FMECA-Arbeitsblatt zum betrachteten Anwendungsfall

Funktion / Anforder.	Fehlerart(en)	Fehlerauswrikung(en)	Schwere	Fehlerursache(n)	Auftreten	RPN
Folgegeschwindigkeit errechnen	Folgegeschwindigkeit kann nicht errechnet werden	Fahrzeug wird nicht verlangsamt und kollidiert mit vorausfahrendem Fahrzeug	10	Ist-Geschwindigkeit und / oder Abstand zum vorausfahrenden Fahrzeug liegen nicht vor	1	10
			10	Softwarefehler bei der Berechnung	2	20
	Folgegeschwindigkeit wird zu langsam errechnet	Fahrzeug wird verspätet verlangsamt und kollidiert mit vorausfahrendem Fahrzeug	10	Ist-Geschwindigkeit und / oder Abstand zum vorausfahrenden Fahrzeug liegen verspätet vor	1	10
			10	Softwarefehler bei der Berechnung	2	20
		Fahrzeug wird verspätet verlangsamt und kann Sicherheitsabstand nicht einhalten	4	Ist-Geschwindigkeit und / oder Abstand zum vorausfahrenden Fahrzeug liegen verspätet vor	1	4
			4	Softwarefehler bei der Berechnung	2	8
	Folgegeschwindigkeit wird zu hoch errechnet	Fahrzeug wird zu wenig verlangsamt und kollidiert mit vorausfahrendem Fahrzeug	10	Ist-Geschwindigkeit liegt falsch (zu niedrig) und / oder Abstand zum vorausfahrenden Fahrzeug falsch (zu groß) vor	3	30
			10	Softwarefehler bei der Berechnung	2	20
	Folgegeschwindigkeit wird zu niedrig errechnet	Fahrzeug wird stärker verlangsamt als zur Einhaltung des Sicherheitsabstandes notwendig	1	Ist-Geschwindigkeit liegt falsch (zu hoch) und / oder Abstand zum vorausfahrenden Fahrzeug falsch (zu klein) vor	3	3
			1	Softwarefehler bei der Berechnung	2	2

Aktivität T2 (Risikoorientierte Testmodellerstellung): Aus dem anforderungsbasierten Testmodell zum ACC-System, welches im Beispiel auch das Systemmodell bei der FMECA ist, sowie aus den quantifizierten Produktrisiken der FMECA wird das risikoorientierte Testmodell gebildet. Im risikoorientierten Testmodell werden die analysierten Risiken explizit dokumentiert und formalisiert. Abbildung 3 stellt den Ausschnitt des risikoorientierten Testmodells für das ACC-System dar, welcher das Verhalten des betrachteten Anwendungsfalls modelliert. Den drei Aktivitäten des Anwendungsfalls wird als Risikowert jeweils die Summe aller Risikoprioritätszahlen aus dem FMECA-Arbeitsblatt zugewiesen, die sich auf Fehlerauswirkungen bzw. Fehlerursachen zur Aktivität beziehen (analog zur Summierung von Risiken verschiedener Fehlerereignisse in [Ro99]). Die Annotation der Risikowerte im Testmodell erfolgt auf Basis des UML

[2] Einige Anwendungen der FMECA unterscheiden zusätzlich den Grad der Fehlererkennung auf Systemebene. In diesen Anwendungen wird eine zusätzliche Kategorie für Fehlererkennung benutzt um die RPN als Produkt dreier Faktoren zu bilden.

QoS-Profils in Kommentaren zu den entsprechenden Testmodellelementen (Aktionen). Eine erneute Durchführung der FMECA im Rahmen der prozessbegleitenden Sicherheitsanalyse macht lediglich die Aktualisierung dieser Kommentare notwendig.

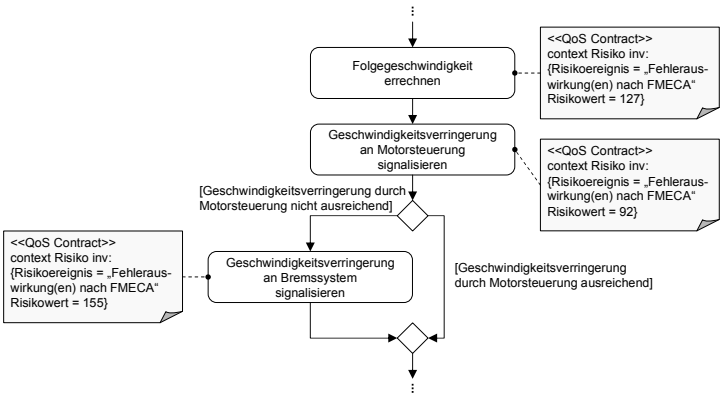

Abbildung 3: Ausschnitt aus risikoorientiertem Testmodell zum betrachteten Anwendungsfall

Aktivität T3 (Modellbasierte Testfallableitung): Das risikoorientierte Testmodell kann direkt zur automatischen Testfallableitung mittels RiteDAP genutzt werden. Tabelle 3 listet die beiden möglichen Pfade durch den Testmodellausschnitt aus Abbildung 3. Die Pfade stellen Teilpfade durch das risikoorientierte Testmodell des Gesamtsystems dar und somit Teilszenarien der für das Gesamtsystem ableitbaren Testfallszenarien für den Systemtest.

Aktivität T4 (Risikobasierte Testfallpriorisierung): In RiteDAP werden die abgeleiteten Testfälle automatisch priorisiert. Dies erfolgt auf Basis der im risikoorientierten Testmodell explizit dokumentierten quantifizierten Produktrisiken. In Tabelle 3 wird das Ergebnis dieser risikobasierten Testfallpriorisierung mit RiteDAP unter Verwendung der Priorisierungsstrategie TRSP für beide Testfallteilszenarien dargestellt.

Nummer	Testfallteilszenario	Risikowert
1	Folgegeschwindigkeit errechnen, Geschwindigkeitsverringerung an Motorsteuerung signalisieren, Geschwindigkeitsverringerung an Bremssystem signalisieren	374 (127+92+155)
2	Folgegeschwindigkeit errechnen, Geschwindigkeitsverringerung an Motorsteuerung signalisieren	219 (127+92)

Tabelle 3: Mögliche Testfallteilszenarien zum betrachteten Anwendungsfall und ihre Risikowerte

Das risikoorientierte Testmodell für das gesamte ACC-System enthält auf Gesamtsystemebene (16 Anwendungsfälle) ca. 110 Aktionen und ca. 30 Fallunterscheidungen. Die Vielzahl der hieraus ableitbaren Testfallszenarien macht die Notwendigkeit der automatischen Testfallpriorisierung deutlich. Ohne Testfallpriorisierung könnte eine Ordnung oder Auswahl der Testfallszenarien für die Testdurchführung nur zufällig getroffen werden. Und ohne Automatisierung der Testfallpriorisierung wäre mit jeder Aktualisierung der im risikoorientierten Testmodell dokumentierten Produktrisiken eine aufwändige und fehleranfällige manuelle Neupriorisierung aller Testfallszenarien notwendig.

5. Zusammenfassung und Ausblick

Dieser Beitrag hat gezeigt, wie sich Synergieeffekte durch die Nutzung quantitativer Sicherheitsanalysen für den risikobasierten Test realisieren lassen. Im vorgeschlagenen Ansatz werden quantifizierte Produktrisiken aus Sicherheitsanalysen beim risikobasierten Test sicherheitsgerichteter, eingebetteter Systeme zur automatischen Testfallableitung und -priorisierung genutzt. Hiermit wird der beim risikobasierten Test typischerweise anfallende Zusatzaufwand vermindert, da keine separaten Produktrisikobewertungen durchgeführt werden müssen. Insbesondere in iterativen Entwicklungsprozessen mit veränderlichen Produktrisiken ist eine fortlaufende Aktualisierung der Produktrisikobewertung für risikobasierte Tests notwendig, da veraltete Bewertungen zu einer falschen Allokation des Testaufwandes führen können. Da Sicherheitsanalysen während der Entwicklung prozessbegleitend durchgeführt werden, kann der vorgeschlagene Ansatz kontinuierlich aktualisierte Produktrisiken zur Testfallableitung und -priorisierung verwenden, was zu einer kosteneffizienten und zuverlässigen Allokation des Testaufwandes entsprechend der laufend aktualisierten Produktrisiken führt.

Mit der FMECA wurde in diesem Beitrag ein ausgereiftes und normiertes Verfahren der quantitativen Sicherheitsanalyse für die Produktrisikobewertung vorgeschlagen. Während die Anwendung der FMECA auf System- und Hardwareebene in der Praxis akzeptiert und weit verbreitet ist, wird Software bei der Sicherheitsanalyse eingebetteter Systeme vorwiegend als Blackbox betrachtet (vgl. [DT08]). Mit dem stetig wachsenden Anteil der durch Software realisierten Funktionalität eines eingebetteten Systems gewinnt die Sicherheitsanalyse der Software jedoch an Bedeutung. Oftmals wird der FMECA nur eine eingeschränkte Anwendbarkeit auf Software zugesprochen (vgl. [Li00]). Eine genaue Analyse der FMECA sowie weiterer Techniken der quantitativen Sicherheitsanalyse (z.B. FTA) hinsichtlich ihrer Eignung für den vorgeschlagenen Ansatz ist deshalb Gegenstand unserer zukünftigen Arbeiten.

Die FMECA wird erfolgreich vorzugsweise im frühen Entwicklungszyklus durchgeführt, damit die Behebung identifizierter Sicherheitsprobleme möglichst kosteneffizient ist. Sie ist ein iterativer Prozess, der den Entwicklungsprozess begleitet (vgl. [DIN06]). Üblicherweise wird die FMECA auf Wirkstrukturen durchgeführt, die sich aus Architekturen ergeben. Im vorgeschlagenen Ansatz haben wir gezeigt, wie in Anwendungsfällen aggregierte Ziele und Szenarien die Grundlage für die quantitative Sicherheitsanalyse bilden können und somit eine frühe, anforderungsbasierte FMECA ermöglicht wird (vgl. [AK01]). Es ist zu erwarten, dass zusätzliches Architekturwissen eine genauere Analyse ermöglicht und die Analyse normativen Anforderungen zudem besser gerecht wird. In SPES 2020 wird daher untersucht, wie sich Entwicklungsansätze, die den Architekturentwurf mit dem ziel- und szenariobasierten Requirements Engineering verzahnen (z.B. COSMOD-RE, [Po08]), positiv auf frühe Sicherheitsanalysen auswirken können.

Literaturverzeichnis

[AK01] K. Allenby, T. Kelly: Deriving Safety Requirements Using Scenarios. In: Proc. of 5th RE, Toronto, Ontario, Canada, 2001, pp. 228-235.

[Ba99] J. Bach: Risk-Based Testing. How to conduct heuristic risk analysis. In: Software Testing & Quality Engineering Magazine, vol. 1, iss. 6, 1999, pp. 23-28.
[Ba08] T. Bauer, H. Stallbaum, A. Metzger, R. Eschbach: Risikobasierte Ableitung und Priorisierung von Testfällen für den modellbasierten Systemtest. In: Proc. of SE, München, Germany, ser. LNI, vol. 121. GI, 2008, pp. 99-111.
[Bo88] B. W. Boehm: A spiral model of software development and enhancement. In: IEEE Computer, vol. 21, no. 5, pp. 61-72, 1988.
[CP03] Y. Chen, R.L. Probert: A Risk-based Regression Test Selection Strategy. In: Proc. of 14th ISSRE, Denver, CO, USA, 2003, pp. 305-306.
[DIN03] DIN EN 61508: Funktionale Sicherheit sicherheitsbezogener elektrischer / elektronischer / programmierbarer elektronischer Systeme. DIN, Germany, 2003.
[DIN06] DIN EN 60812: Analysetechniken für die Funktionsfähigkeit von Systemen – Verfahren für die Fehlzustandsart- und -auswirkungsanalyse (FMEA). DIN, Germany, 2006.
[DIN07] DIN EN 61025: Fehlerzustandsbaumanalyse. DIN, Germany, 2007.
[DT08] D. Domis, M. Trapp: Integrating Safety Analyses and Component-based Design. In: Proc. of 27th SAFECOMP, Newcastle upon Tyne, UK, 2008, pp. 58-71.
[Ea92] J. V. Earthy: Hazard and Operability Study as an Approach to Software Safety Assessment. In: Colloqium on Hazard Analysis, London, UK, 1992, pp. 5/1-5/3.
[ISO09] ISO/DIS 26262: Road vehicles – Functional safety. ISO, Switzerland, 2009.
[Li00] P. Liggesmeyer: Formale und stochastische Methoden zur Qualitätssicherung technischer Software (eingeladener Vortrag). In: Softwaretechnik-Trends, vol. 20, no. 3, 2000.
[Li02] P. Liggesmeyer: Software-Qualität. Testen, Analysieren und Verifizieren von Software. Spektrum Akademischer Verlag, 2002.
[OMG08] UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms. Version 1.1. OMG, 2008.
[Pi04] I. Pinkster, B. van de Burgt, D. Janssen, and E. van Veenendaal: Successful Test Management: An Integral Approach, 2nd ed., Springer, Berlin, 2004.
[Po08] K. Pohl: Requirements Engineering: Grundlagen, Prinzipien, Techniken. 2nd ed., dpunkt Verlag, 2008.
[Re05] A. Reuys, E. Kamsties, K. Pohl, S. Reis: Model-based System Testing of Software Product Families. In: Proc. of 17th CAiSE, Porto, Portugal, 2005, pp. 519-534.
[RE03] G. Rothermel, S. Elbaum: Putting your best Tests forward. In: IEEE Software, vol. 20, no. 5, 2003, pp. 74-77.
[Ro99] L.H. Rosenberg, R. Stapko, A. Gallo: Risk-based Object Oriented Testing. In: Proc. of 24th annual Software Engineering Workshop, NASA, SEL, Greenbelt, MD, USA, 1999.
[SAE96] ARP 4761: Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment. SAE, 1996.
[SM07] H. Stallbaum, A. Metzger: Employing Requirements Metrics for Automating Early Risk Assessment. In: Proc. of MeReP07, Palma de Mallorca, Spain, 2007, pp. 1-12.
[Sr08] P. R. Srivastva, K. Kumar, and G. Raghurama: Test case prioritization based on requirements and risk factors. In: Softw. Eng. Notes, vol. 33, no. 4, 2008, pp. 1-5.
[St08] H. Stallbaum, A. Metzger, K. Pohl: An automated Technique for risk-based Test Case Generation and Prioritization. In: Proc. of 3rd AST, Leipzig, Germany, 2008, pp. 67-70.
[SW05] H. Srikanth, L. Williams: On the economics of requirements-based test case prioritization. In: Proc. of 7th EDSER, New York, NY, USA, 2005, pp. 1-3.
[Tr99] N. Tracey, J. Clark, J. Mcdermid, K. Mander: Integrating Safety Analysis with Automatic Test-data Generation for Software Safety Verification. In: Proc. of 17th ISSC, Orlando, FL, USA, 1999, pp. 128-137.
[Wi03] H. Winner: ACC Adaptive Cruise Control. Robert Bosch GmbH, Stuttgart, 2003.
[Zi09] F. Zimmermann, R. Eschbach, J. Kloos, T. Bauer: Risk-based statistical Testing: A refinement-based Approach to the Reliability Analysis of Safety-critical Systems. In: Proc of 12th EWDC, Toulouse, France, 2009.