Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Michael Hamester

Methodik für die systematische Systemmodellierung als Basis für die Entwicklung und Bewertung sicherheitsrelevanter Systeme

Fakultät Technik und Informatik
Department Fahrzeugtechnik und Flugzeugbau

Faculty of Engineering and Computer Science
Department of Automotive and Aeronautical Engineering

Michael Hamester

Methodik für die systematische Systemmodellierung als Basis für die Entwicklung und Bewertung sicherheitsrelevanter Systeme

Masterarbeit eingereicht im Rahmen der Masterprüfung

im Studiengang Fahrzeugbau
am Department Fahrzeugtechnik und Flugzeugbau
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

in Zusammenarbeit mit:
:em engineering methods AG
Abteilung: MBSE
Rheinstraße 97
64295 Darmstadt

Erstprüfer/in: Prof. Dr. Ing. Jutta Abulawi
Zweitprüfer/in: Dr. Ing. Stephan Husung

Industrieller Betreuer/in: Dr. Ing. Stephan Husung

Abgabedatum: 31.01.2018

Zusammenfassung

Michael Hamester

Thema der Masterthesis

„Methodik für die systematische Systemmodellierung als Basis für die Entwicklung und Bewertung sicherheitsrelevanter Systeme“

Stichworte

Systemmodellierung, Modell-basiertes Systems Engineering, Sicherheitsanalysen

Kurzzusammenfassung

Durch die steigenden Anforderungen und normativen Vorgaben bei der Entwicklung von sicherheitsrelevanten Systemen rücken das methodische Vorgehen und systematische Entwickeln immer mehr in den Vordergrund.

In dieser Arbeit wird eine Methodik aufgezeigt, in der die sicherheitsrelevanten Anforderungen, die sich aus der Gefahren- und Risikoanalyse auf der ITEM Ebene des Entwicklungsprozesses der ISO 26262 ergeben, analysiert und in einem umfassenden Systemmodell repräsentiert werden. Das Systemmodell stellt dabei die zentrale Informationsdatenbasis in der Produktenwicklung sicherheitsrelevanter Systeme dar. Aus dem Systemmodell werden entsprechend der normativen Vorgaben induktive und deduktive Sicherheitsanalysen (FMEA, FTA) abgeleitet. Die gewonnenen Erkenntnisse fließen durch Iterationsschleifen in das Systemmodell mit ein und optimieren bzw. verfeinern dieses stetig. Dabei spielt die kontinuierliche Dokumentation der dynamischen Entwicklung des Systemmodells aus Sicht der Nachverfolgbarkeit eine zentrale Rolle. Anhand eines Systembeispiels wird die Methodik für die modell-basierte Entwicklung und die Ableitung der sicherheitsrelevanten Analysen sowie Maßnahmen dargestellt und validiert.

Abstract

The increasing requirements and normative specifications for the development of safety relevant systems, move the methodology and systematic development increasingly into the focus.

In this work, a method is developed, in which safety relevant requirements are analyzed and represented in a comprehensive system model. These requirements arise from a hazard analysis and risk assessment (HARA) of the ISO 26262 development process on ITEM level. The system model thereby portrays the central information data base in the product development of safety relevant systems. An inductive and deductive safety analysis, respective of normative standards, is derived from the system model. The obtained findings are iteratively entered in the system model, to continually optimize and refine this. The continuous dynamic development documentation of the system model plays a central role in the traceability of the system. The methodology for the model-based development, and the deduction of the safety relevant analyses and measures is outlined and validated by means of a system example.

Vorwort/ Danksagung

Zunächst möchte ich mich bei allen bedanken, die mich bei der Anfertigung dieser Masterarbeit unterstützt haben.

Mein besonderer Dank gilt Frau Prof. Dr.-Ing. Jutta Abulawi für die Betreuung und Unterstützung der Arbeit von Seiten der Hochschule.

Des Weiteren möchte ich mich bei Herrn Dr.-Ing. Stephan Husung bedanken, der die industrielle Betreuung der Arbeit von Seiten des Unternehmens :em engineering methods AG übernommen hat.

Abschließend möchte ich mich bei allen Mitarbeitern der :em engineering methods AG bedanken, die mir während der Anfertigung der Masterarbeit mit voller Unterstützung zur Seite standen.

Inhaltsverzeichnis

Abkürzungsverzeichnis

A	Auftretenswahrscheinlichkeit
act	Activity
ASIL	Automotive Safety Integrity Level
B	Bedeutung
bdd	Block Definitionsdiagramm
C	Controllability, Kontrollierbarkeit
E	Exposure, Entdeckungswahrscheinlichkeit
FMEA	Failure Mode & Effect Analysis
FSC	Functional Safety Concept
FTA	Fault Tree Analysis
FuSi	Funktionale Sicherheit
HARA	Hazard Analysis & Risk Assessment
HSR	Hardware Safety Requirements
ibd	Internes Blockdiagramm
ID	Identification
INCOSE	International Council on Systems Engineering
MBSE	Modell-basiertes Systems Engineering
QM	Qualitätsmanagement
RPZ	Risiko-Prioritätszahl
req	Requirement
S	Severity, Schwere
SE	Systems Engineering
SSR	Software Safety Requirements
SysML	Systems Modeling Language
TSC	Technical Safety Concept
UML	Unified Modeling Language

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Die folgende Masterarbeit ist in Zusammenarbeit mit dem Unternehmen :em engineering methods AG, in Darmstadt entstanden.

1.1 Motivation

Seit 2011 hält die ISO 26262 als normative Vorgabe Einzug in die Automobilindustrie, wenn es um die funktionale Sicherheit und die Entwicklung sicherheitsrelevanter Systeme im Fahrzeug geht. Die Norm stellt die Industrie vor große Herausforderungen, die bis dato in vielen Fällen noch nicht bewältigt sind. Die Grundanforderung der Norm ist, dass die Unternehmen nach dem aktuellen Stand der Technik arbeiten. Dies heißt einerseits, dass die einzelnen Aktivitäten der Norm systematisch durchlaufen und nachvollziehbar dokumentiert werden müssen, um nach dem geforderten Prozess der ISO zu entwickeln. Andererseits müssen dem Stand der Technik entsprechende Entwicklungsmethoden in den Entwicklungsprozess der ISO integriert werden. Derzeit wird eine Vielzahl an Methoden, Maßnahmen und mit vielen unterschiedlichen Tools untersucht, wie die Herausforderungen in der Praxis bewältigt werden können. Ein methodischer Ansatz zur Beherrschung der Komplexität ist das modell-basierte Entwickeln. Die ISO 26262 empfiehlt das modell-basierte Entwickeln, jedoch wird nicht dargestellt, wie dies durchgeführt werden soll.

Auch in anderen Branchen stehen die Entwickler vor ähnlichen Problemen. Schon 2006 beschäftigte sich die NASA in einer Forschungsarbeit [Hei06] mit der modell-basierten Ableitung von Sicherheitsanalysen aus einem übergeordneten Systemmodell. Es wurden vorgefertigte Standardfehler in Bibliotheken zusammengefasst, mit denen die Systemmodelle beaufschlagt wurden. Ein Ziel war, verschiedene definierte Fehler auf die System- bzw. Komponentenstruktur anzuwenden, ohne die Systemstruktur jedes Mal neu erstellen zu müssen. Die Integration dieser Methode in damalig gängige Analyse-Software, wäre möglich, aber mit einem enorm hohen Entwicklungsaufwand verbunden gewesen.

Der logische Zeitpunkt an dem die Sicherheit eines Systems nachgewiesen werden sollte, ist während der Entwicklungsphase bis zur Serienproduktion. Da die einzelnen Entwicklungsschritte aber iterativ durchgeführt werden, müssen die Sicherheitsanalysen und –Maßnahmen ebenfalls iterativ erfolgen. Demnach ist eine modell-basierte Integration der erforderlichen Sicherheitsanalysen und –Maßnahmen in den Entwicklungsprozess unumgänglich. Ein Ansatz ist die Verknüpfung von den Methoden: MBSA (Modell-basierte Sicherheitsanalysen) und MBSE (Modell-basiertes Systems Engineering). In [Zha17] wird vorgeschlagen, beide Methoden im Entwicklungsprozess zu verknüpfen, um die Entwicklung zu optimieren und die Kosten zu senken. Demzufolge wird der Designprozess eines Systems effizienter, wenn die Bewertung des Systemdesigns am Anfang und am Ende der Entwicklungsphase stattfinden und die Analysemethoden modell-basiert durchgeführt werden.

Derzeit werden in der Industrie verschiedene Tools verwendet, die oft als gewachsene Individuallösungen eingesetzt werden, um Sicherheitsanalysen durchzuführen. [Adl15] Der Nachteil ist, dass diese Sicherheitsanalysen rein zur Analyse und Bewertung von internen erstellten Systemstrukturen und Modellen dienen und sich somit nicht ohne Mehraufwände in den Entwicklungsprozess integrieren lassen. Die zu analysierenden

Modelle werden somit separat mit Mehraufwand und mit der Gefahr der Inkonsistenz aufgebaut. [Adl15]

Viele Abteilungen und Zulieferer entwickeln an ein und demselben System. Das Problem ist, dass nicht mit einheitlichen Modellen, Softwarelösungen und Prozessen gearbeitet wird. In [Duo13] wird auf Basis einer Sicherheitsanalysesoftware eine Methodik aufgezeigt, um den Sicherheitsmodellierungsprozess zu vereinheitlichen und den beteiligten Entwicklern ein zentralisiertes, einheitliches, konsistentes und korrektes Datenmodell zur Verfügung zu stellen.

In [Adl12] werden Ansätze aufgezeigt, welche die Integration der Sicherheitsanalysen in den modell-basierten Entwicklungsprozess darstellen. Diese Ansätze wurden im Projekt „*e performance",* einem BMBF-Förderprojekt zur Entwicklung eines rein elektrischen Fahrzeugs von Audi und verschiedenen Industriepartner angewendet. Dabei wurden zwei grundlegende Ansätze aus dem Safety Engineering des Fraunhofer IESE mit in das Arbeitspaket „Funktionale Sicherheit" mit eingebracht. Der „*Failure View"*, der die Ursache-Wirkungszusammenhänge von Fehlern darstellt und somit automatisierte Sicherheitsanalysen ermöglicht und der „*Safety Concept View"*, der die Zusammenhänge von Funktionen mit deren Signalen und Anforderungen aufzeigt. [Adl12]

Nach [Adl15] bietet derzeit kein Werkzeug die vollständige und durchgängige Integration von Sicherheitsanalysen in einem Modellierungstool. Es ist auch nicht ohne weiteres möglich, vorhandene Systemstrukturen aus übergeordneten Systemmodellen in die gängigen Analysesoftwarelösungen zu überführen und zu bewerten. [Adl15]

Die Herausforderung besteht somit in der Integration dieser Safety Engineering Mechanismen in gängige Modellierungswerkzeuge. Ein weiterer Ausblick der von [Adl12] beschrieben wird, ist die modell-basierte Kopplung von Item Definition und datenbankgestützter Gefahren- und Risikoanalyse.

An dieser Stelle wird in dieser Arbeit angesetzt. Ziel ist die Entwicklung einer durchgängigen Methodik, in welcher die normativen Vorgaben des Entwicklungsprozesses der ISO 26262 auf Basis eines einheitlichen Systemmodells, als zentrale Informationsdatenbasis, abgebildet werden.

1.2 Abgrenzung der Arbeit

Bei der vorliegenden Arbeit liegt der Fokus auf der Entwicklung der Methodik mit einem modell-basierten Ansatz und Validierung der dargestellten Methodik an einem konkreten Beispiel. Dabei liegt der Fokus auf der Durchführung und Darstellung der HARA (Hazard Analysis & Risk Assessment) sowie der Sicherheitsanalysen FTA (Fault Tree Analysis) und FMEA (Failure Mode and Effect Analysis) durch das Systemmodell. Die Sicherheitsanalysen sollen möglichst systematisch, ganzheitlich, und durchgängig aus dem Systemmodell aufgebaut und durchgeführt werden. Weitere Aktivitäten, die sich durch modell-basierte Ansätze unterstützen lassen, bilden einen Ausblick für diese Arbeit.

1.3 Aufbau der Arbeit

Die vorliegende Arbeit ist wie folgt aufgebaut.

Zunächst wird ein Überblick über die Grundlagen gegeben, welche benötigt wurden, um die Masterarbeit durchzuführen. Dazu gehört eine Einführung in *Systems Engineering*, welche den Entwicklungsprozess auf Basis des V-Modells, das *Model-based Systems Engineering* und die Sprache *SysML* (Systems Modeling Language) umfasst.

In Kapitel 2.5 wird auf sicherheitsrelevante Normen eingegangen, die für die Masterarbeit von Relevanz sind. In Kapitel 2.6 werden der Aufbau und die Durchführung der Sicherheitsanalysen FMEA und FTA dargestellt.

Der Hauptteil der Arbeit ist in drei Teile untergliedert. Zunächst wird in Kapitel 3 die erarbeitete Methodik ausführlich aufgezeigt. Diese wird in Kapitel 4 anhand eines exemplarischen Beispiels validiert. Anschließend werden in der Ergebnisdiskussion in Kapitel 5, die Ergebnisse der Arbeit, die Vor- und Nachteile der Methode diskutiert.

Abschließend wird in Kapitel 6 das Fazit und der Ausblick diskutiert und aufgezeigt wie weiter auf der Arbeit aufgebaut werden kann.

2 Grundlagen

Im Kapitel 2 sollen die Grundlagen beschrieben werden, die für die Erarbeitung der Masterarbeit von Relevanz sind.

2.1 SE Systems Engineering

In diesem Kapitel soll der Ursprung, die Definition und die Verwendung von Systems Engineering erklärt werden.

Der Ursprung bei der Entwicklung von Systems Engineering als Disziplin geht laut [Wal15] auf die späten 1930er Jahre zurück, als bei komplexen Projekten für neuartige Produkte neue systematische Ansätze zur Umsetzung der Projekte eingesetzt wurden. Diese Ansätze wurden systematisch, bis daraus Standardisierungen hervorgingen, z.B. Norm ISO /IEC 15288. [Wal15]

Nach [Wal15] wurde 2002 die internationale Norm ISO/IEC 15288 eingeführt und Systems Engineering als bevorzugter Standard zwischen Parteien bei der Entwicklung und Erbringung von Dienstleistungen anerkannt. Systems Engineering wird als übergeordnete Disziplin eingesetzt um komplexe Systeme zu beschreiben und stellt die gesamtheitliche Methodik dar, um komplexe Aufgaben vom IST in den SOLL Zustand zu überführen. [Hab12]

In Abbildung 1 ist dies in einer Grafik verdeutlicht.

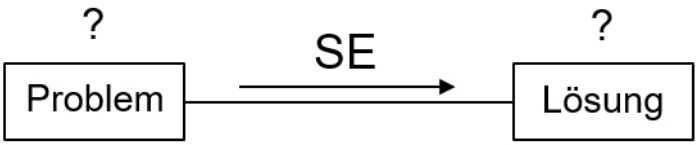


Abbildung 1: SE, Brücke zwischen Problem- & Lösungsraum nach [Hab12]

Nach [Hab12] kann Systems Engineering als Methodik alleine nicht zum Erfolg eines Projektes führen, viel mehr sollte diese Methodik interdisziplinär als übergeordnete Organisationmethodik über alle beteiligten Disziplinen eingesetzt werden.

Laut [Wal15] kann Systems Engineering durch vier Begriffe beschrieben werden:

„*Interdisziplinär, iterativ, soziotechnisch und ganzheitlich*."

Nach [Eig14] sind unabhängig von Art des Systems, drei Aspekte im Systems Engineering bei der Gestaltung des Systems zu berücksichtigen.

Die Anforderungen

Die Anforderungen an das System müssen klar definiert sein und die durchgängige Verifizierung und Validierung muss über den Entwicklungsprozess nachverfolgbar gewährleistet sein. [Eig14]

Der Mensch

Das Systems Engineering wird als interdisziplinärer Entwicklungsprozess von Menschen umgesetzt. Demnach ist eine Vielzahl von unterschiedlichen Menschen an der Entwicklung eines Systems beteiligt. Die Interessen der verschiedenen Stakeholder müssen systematisch erfasst und realisiert werden. Die Kommunikation der Stakeholder spielt dabei eine große Rolle. Der interdisziplinäre Informationsaustausch wird methodisch unterstützt. Es muss ein einheitliches Systemverständnis und ein ganzheitliches Systemdenken geschaffen werden. [Eig14]

Der Systemlebenszyklus

Der Systemlebenszyklus muss bei der Entwicklung des Systems berücksichtigt werden. Demnach werden die Randbedingungen der Systemnutzung in den Anforderungen über die Stakeholder definiert. Jedes System wird für eine bestimmte Nutzungsart und Dauer spezifiziert. [Eig14]

Diese drei Aspekte stellen nach [Eig14] die Grundpfeiler des Systems Engineering dar. Um Systems Engineering als Entwicklungsmethode nachhaltig einzusetzen, muss diese Methodik in den Entwicklungsprozess integriert werden.

2.2 Das V-Modell im Systems Engineering

In diesem Abschnitt werden das V-Model und dessen Verwendung im Systems Engineering dargestellt.

Ursprung

Nach [Jas17] wurde das V-Modell, welches im Systems Engineering Verwendung findet 1991 von Kevin Forsberg und Harold Mooz in der Veröffentlichung *„The Relationship of System Engineering to the Project Cycle“* vorgestellt. Dieses V-Model ist nicht zu verwechseln mit dem V-Model 97 und dem V-Model XT, welche ihren Ursprung in der Softwareentwicklung haben. Das V-Model löst das Wasserfallmodel und das Spiralmodell ab, die in der Fachwelt als zu praxisfremd gehalten wurden. [Jas17]

Aufbau

In Abbildung 2 ist das V-Modell dargestellt.

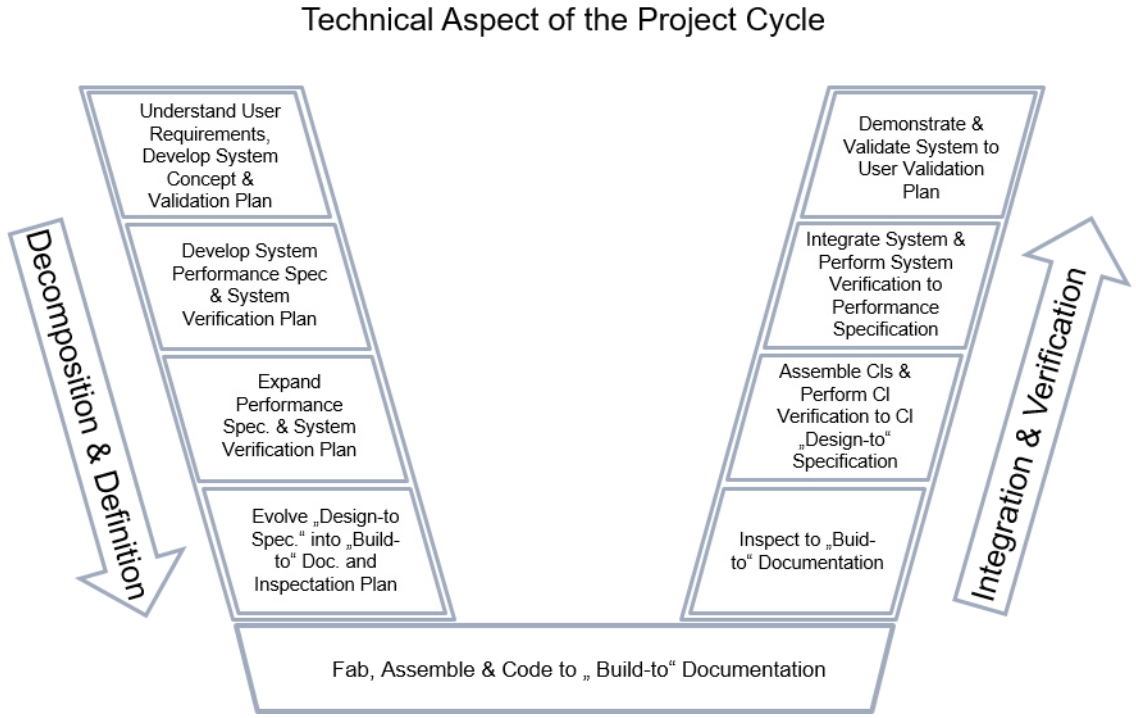


Abbildung 2: V-Modell angelehnt an [For91]

Das V-Modell ist durch zwei Äste, welche die Form des Modells beschreiben, aufgebaut. Der linke absteigende und der rechte aufsteigende Ast. Der Entwicklungsprozess wird über das V-Modell von links oben, durch das V, nach rechts oben beschrieben. [For91]

Links oben startet der Entwicklungsprozess mit der Anforderungsanalyse für das zu entwickelnde System. Der linke Ast wird beim Abstieg mit Informationen und Daten bezüglich Anforderungen, Funktionen und Systemleistung bis zum Scheitelpunkt des V's immer spezifischer angereichert. Der Scheitelpunkt stellt das Tor zum realen Produkt dar. Der aufsteigende rechte Ast stellt die Produktion, Validierung, Verifizierung, und Demonstration des fertigen Produktes dar. Der beschriebene Prozess läuft nicht statisch und einmalig ab. Vielmehr sind in das V-Modell verschiedene Iterationsschleifen eingearbeitet, welche die einzelnen Prozessschritte verifizieren. [For91]

Demnach ist der linke Ast die Dekomposition und Definition des zu spezifizierenden Modells mit ähnlichen Eigenschaften des Wasserfallmodells. Der rechte Ast bildet die Integration und Verifikation des Produktes ab. [For91]
Das V-Modell ist somit ein realitätsnäherer Entwicklungsprozess, der die Prozesse Wasserfallmodell und Spiralmodell ablöst, aber Eigenschaften dieser Prozesse beinhaltet. Nach [For91] ist das V-Modell dreidimensional. Die Dritte Ebene geht in der Darstellung in die Tiefe und stellt die Dekomposition des Systems in Subsysteme dar.

Anwendung mit Systems Engineering

Das V-Modell stellt bei der Entwicklung mechatronische Produkte den übergeordneten Entwicklungsprozess dar. In Abbildung 3 ist das V-Model mit dem RFLP Ansatz abgebildet. Das *Systems Engineering* ist somit die Schlüsseldisziplin zur iterativen Erarbeitung der einzelnen Prozessschritte im V-Model.

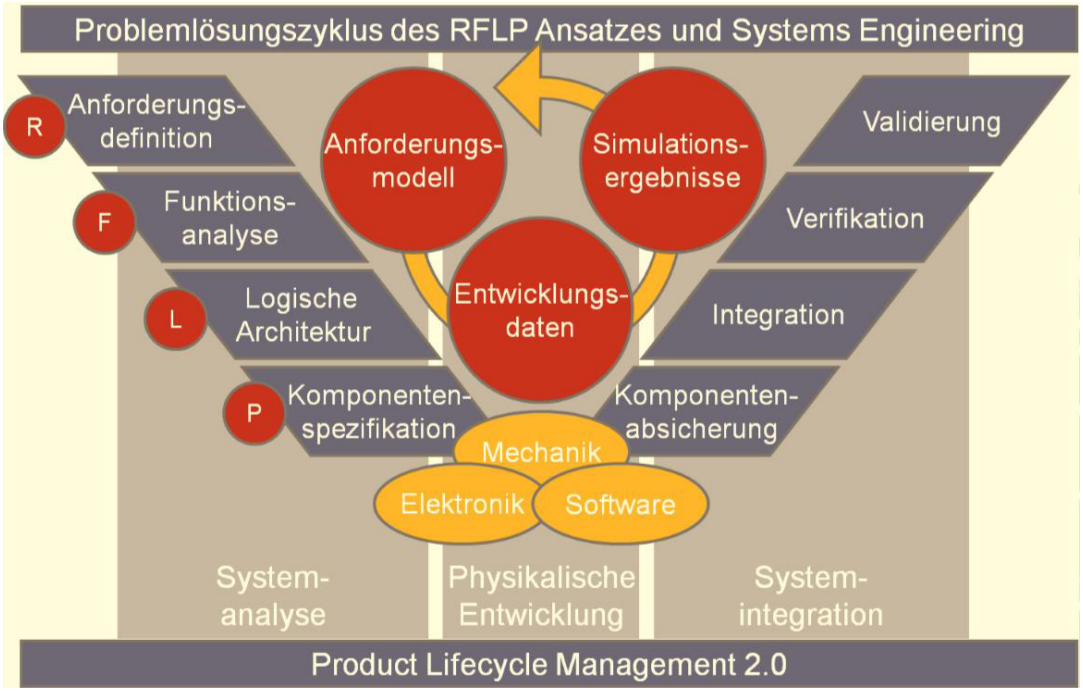


Abbildung 3: V-Modell RFLP Ansatz [Kle13]

Die einzelnen Prozessschritte werden durch das *Systems Engineering* erarbeitet und abgebildet.

2.3 MBSE Modell-basiertes Systems Engineering

Das klassische Systems Engineering stellt einen Text und dokumentenbasierten Entwicklungsansatz dar. Dieser Ansatz wird durch die modell-basierte Entwicklung erweitert. Demnach ist MBSE nach [Wal15] die formalisierte Anwendung von Modellierungsmethoden um die einzelnen Konzeptphasen in der Entwicklung eines Systems zu unterstützen.

Modell-basiertes Systems Engineering stellt demnach eine spezielle Vorgehensweise von Systems Engineering dar, in der alle erfassten Daten und Erkenntnisse eines Projektes/ Systems in einem zentralen Systemmodell erfasst und verwaltet werden sollen. Es soll somit der Schritt von der dokumentenbasierten Entwicklung zur modell-basierten Entwicklung gegangen werden. [Eig14]

Dieser Schritt führt laut [Wal15] zu einer besseren Vernetzung aller Beteiligten eines Projekts und sorgt für Verbesserungen in der Produktqualität, Kommunikation, Zusammenarbeit und Nachvollziehbarkeit. Allerdings stehen und fallen die Vorteile mit der Ausprägung von MBSE. Es sollten möglichst viele Methoden und Prozesse in MBSE überführt werden, da sonst eine Vermischung von Dokumenten zentrierter und modell-basierter Entwicklung stattfindet, welche sich eher negativ auf die Komplexität des Entwicklungsprozesses auswirkt. Das Systemmodell ist somit die zentrale Informationsdatenbasis der modell-basierten Entwicklung. Folgende Aspekte sollte ein Systemmodell nach [Eig14] erfüllen:

Zentrale Verfügbarkeit

- Allen beteiligten Systemingenieuren liegt der aktuelle Stand vor

Eindeutig Interprätierbar

- Die Modellier- bzw. Systemsprache muss einheitlich sein

Ganzheitlich

- System und Modellelemente müssen ganzheitlich und konsistent vorliegen

Demnach muss zur Umsetzung von MBSE als Entwicklungsmethode, eine spezielle Infrastruktur geschaffen werden, welche die drei Aspekte erfüllt. Damit MBSE effektiv umgesetzt werden kann, muss es genau wie SE zur Umsetzung im jeweiligen angewandten Entwicklungsprozess integriert werden.

2.4 SysML – Systems Modeling Language

Um modell-basiert zu entwickeln, ist es notwendig eine einheitliche Modellierungssprache zu haben. Für den MBSE Prozess hat sich die semi-formale Sprache SysML als Quasi-Standard durchgesetzt. Die SysML stammt ursprünglich von der UML (Unified Modeling Language) ab, welche in der Softwareentwicklung zum Einsatz kommt. Da nicht alle Elemente der UML in der SysML Verwendung finden, wurden bestimmte Elemente der UML außen vorgelassen. [Wei14]

In Abbildung 4 ist die Überschneidung von UML und SysML aufgezeigt.

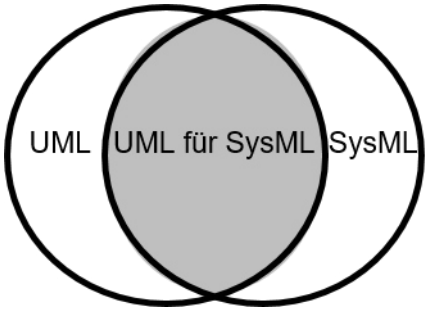


Abbildung 4: Überschneidung UML & SysML angelehnt an [Wei14]

Manche Systemelemente wurden erweitert, andere übernommen oder geändert. Diese Anpassungen sind [Wei14] zu entnehmen. Nach [Wei14] wurde 2006 auf dem OMG-Treffen (Object Management Group) nach einem langjährigen Prozess zweier getrennter Arbeitsgruppen, die SysML als Modellierungsstandard angenommen. 2007 ist OMG SysML als Version 1.0 als offizieller Standard veröffentlicht worden. Die SysML wird stetig in Arbeitsgruppen aus der Industrie weiterentwickelt und es wird nach [Wei14] an einer Version 1.5 gearbeitet.

2.4.1 Aufbau

SysML ist in den Grundzügen der UML sehr ähnlich, wie in Abbildung 4 bereits dargestellt wurde. Bis auf die erweiterten Diagrammtypen kann die SysML über Stereotypen mit jedem UML Werkzeug abgebildet werden. [Wei14] In Abbildung 5 ist der hierarchische Aufbau der SysML-Diagrammstruktur nach [Wei14] aufgezeigt.

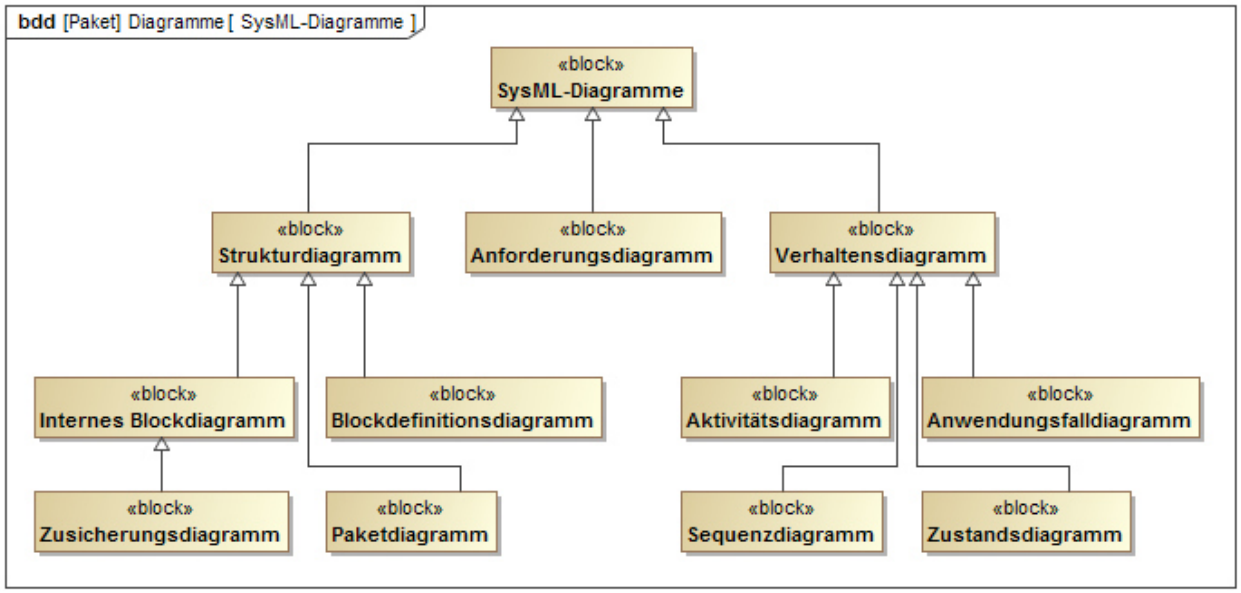


Abbildung 5: SysML Diagrammstruktur angelehnt an [Wei14]

Grundsätzlich kann ein Systemmodell über drei Perspektiven dargestellt werden. Diese sind: die Systemstruktur, die Systemanforderungen und das Systemverhalten. Es ist genau definiert, welche Diagramme benötigt werden um die jeweilige Perspektive abzubilden. Das Interne Blockdiagramm (ibd), das Blockdefinitionsdiagramm (bdd) und das Aktivitätsdiagramm (act) wurden aus der UML umbenannt bzw. erweitert. Neu hinzugekommen sind das Zusicherungsdiagramm und das Anforderungsdiagramm (req). Alle anderen Diagrammarten wurden aus der UML übernommen. [Wei14]

2.4.2 SysML Modellierungstools

Es gibt es sehr viele Modellierungstools, die sich im Aufbau, Preis und Erweiterbarkeit unterscheiden, sie haben ihren Ursprung alle in der UML. Auf der Webseite von [Wei17] ist eine Lister der bekanntesten Modellierungstools aufgezeigt. Einer der bekanntesten und umfangreichsten ist *Cameo Systems Modeler* von der Firma *No Magic*. [Nom17] *Cameo Systems Modeler* stellt den vollen Umfang der SysML Modellierungssprache mit den in Kapitel 2.4.1 aufgezeigten Sichten und Diagrammen zur Modellierung von Systemen bereit. *Cameo Systems Modeler* basiert auf der Magic Draw Modellierungsplattform. Alle in dieser Arbeit modellierten Diagramme sind mit Cameo Systems Modeler 18.4 erzeugt.

2.5 Sicherheitsrelevante Normen

Sicherheitsrelevante Normen gibt es für die unterschiedlichsten Bereiche in der Produktentwicklung. In diesem Kapitel soll der Fokus auf der ISO 26262 liegen, da die Methodik der Masterarbeit welche in Kapitel 3 dargestellt wird, auf sicherheitsrelevante Systeme im Fahrzeugbau abzielt und die ISO 26262 der Standard für die Entwicklung dieser Systeme ist.

2.5.1 IATF 16949

IATF steht für *International Automotive Task Force* und ist der internationale Branchenstandard der Automobilindustrie für Qualitätsmanagementsysteme. Diese Norm steht in enger Verbindung zur Qualitätsmanagement-Norm ISO 9001. In der neuesten Fassung IATF 16949:2016 stellt die Norm keinen eigenständigen QM Standard mehr dar. Sie ist nur in Verbindung mit der Norm ISO 9001:15 anzuwenden. [IAT16]

2.5.2 IEC/ DIN EN 61508

Die IEC 61508 ist eine übergeordnete Norm, für die Entwicklung von elektrischen/ elektronischen und programmierbaren elektronischen sicherheitsrelevanten Systemen, in der Industrie. Die Norm wurde von der *International Electrotechnical Comission* herausgegeben und in die europäische DIN EN 61508 überführt. [IEC615]

Die Norm ist in sieben Teile aufgegliedert:

Teil 1: Allgemeine Anforderungen

Teil 2: Anforderungen für elektrische/ elektronische/ programmierbare elektronische sicherheitsrelevanter Systeme

Teil 3: Software Anforderungen

Teil 4: Definitionen und Abkürzungen

Teil 5: Beispielmethoden zur Bestimmung des Sicherheits-Integritäts-Levels (SIL)

Teil 6: Richtlinien für die Anwendung von Teil 2 und 3

Teil 7: Überblick über Methoden und Maßnahmen

[IEC615]

2.5.3 ISO 26262

Die ISO 26262 ist seit ihrer Veröffentlichung am 15.11.2011 der internationale Standard für die funktionale Sicherheit von Straßenfahrzeugen bis zu einem maximal zulässigen Gesamtgewicht von 3,5t. [ISO11]

Die Norm wurde aus der allgemein gültigen Internationalen Norm IEC 61508 abgeleitet. Die Inhalte wurden an die Bedürfnisse der Entwicklung sicherheitsrelevanter Systeme in Straßenfahrzeugen angepasst.

Die Norm ist in zehn Teile aufgegliedert:

Teil 1: Begriffserklärung

Teil 2: Management der funktionalen Sicherheit

Teil 3: Konzeptphase

Teil 4: Produktentwicklung Systemebene

Teil 5: Produktentwicklung Hardwareebene

Teil 6: Produktentwicklung Softwareebene

Teil 7: Produktion und Betrieb

Teil 8: Unterstützende Prozesse

Teil 9: ASIL (Automotive Safety Integrity Level) und sicherheitsorientierte Analysen

Teil 10: Richtlinien für ISO 26262

Im Folgenden wird nur auf die einzelnen Teile der ISO eingegangen, die später in Kapitel 3, für die Entwicklung der Methodik und in Kapitel 4 zur Validierung der Methodik von Relevanz sind. Diese beschränken sich auf die Teile 3 und 4 bzw. teilweise auf 5, 6 und 9.

Die Erklärung und Übersetzung der normativen Inhalte in diesem Kapitel sind angelehnt an [Hil12].

2.5.3.1 Teil 3: Konzeptphase

In der Konzeptphase gemäß ISO 26262 wird das zu Grunde liegende, zu spezifizierende System (ITEM) definiert. Des Weiteren wird der Sicherheitslebenszyklus initiiert. Das System wird bezüglich der funktionalen Sicherheit, mit Bezug auf Gefährdungen und Risiken, die während des Betriebes auftreten könnten, betrachtet. Diese Gefahren- und Risiken werden nach dem ASIL (Automotive Safety Integrity Level) klassifiziert. Aus dem ASIL werden Sicherheitsziele und Anforderungen definiert, die das System später erfüllen soll.

ITEM Definition

Bei der ITEM Definition wird das System, welches entwickelt werden soll, grundlegend spezifiziert. Das „ITEM“ stellt das zu entwickelnde System dar. Es werden die Komponenten des Systems, die Hauptfunktionen und das Wirkfeld des Systems, bezogen auf Nachbarsysteme und die Umwelt, bestimmt. Alle vorhandenen Informationen über das System sind zu erfassen und in die ITEM Definition miteinzubeziehen. Dieses Vorgehen soll ein grundlegendes Verständnis über das System, welches zu entwickeln ist, schaffen.

Initiation of the Safety Lifecylce

Es wird der Sicherheitslebenszyklus bestimmt. Bei der Produktentstehung wird unterschieden zwischen Systemen die nur modifiziert werden und Systemen die eine komplette Neuentwicklung darstellen. Bei Modifikationen werden einzelne Teile des Systems angepasst und verwendet. In diesem Falle können bestimmte Vorgänge der ISO 26262 entsprechend der Modifikation angepasst werden.

Hazard Analysis & Risk Assessment (HARA)

Basierend auf den Informationen und Erkenntnissen der ITEM Definition, wird die HARA durchgeführt. Die HARA stellt eine Gefahren- und Risikoanalyse dar. Die ISO 26262 stellt ein definiertes Verfahren bereit um die HARA durchführen zu können. Hierbei werden die möglichen Fehlfunktionen des Systems, die zu einer Gefährdung führen könnten, definiert und mit den möglichen Betriebszuständen, in denen das System betrieben werden soll, kombiniert. Das Resultat sind potenzielle Gefahrensituationen die nach dem „Automotive Safety Integrity Level“ (ASIL) qualitativ bewertet werden. Für die Einstufung des Systems in das zugehörige ASIL stellt die ISO 26262 Entscheidungstabellen mit den benötigten Bewertungsparametern zur Verfügung. Jede Gefahrensituation muss bezüglich drei Gesichtspunkten klassifiziert werden, die im nachfolgenden aufgelistet sind.

1. Schwere eines möglichen Schadens mit Bezug auf die Insassen (S=Severity)

S0	keine Verletzungen
S1	leichte bis mittlere Verletzungen
S2	schwere Verletzungen
S3	lebensbedrohliche Verletzungen, überleben unwahrscheinlich

Tabelle 1: Severity angelehnt an [ISO11]

2. Eintrittswahrscheinlichkeit der Situation (E=Exposure)

E0	sehr unwahrscheinlich
E1	sehr niedrig
E2	niedrig
E3	mittel
E4	hoch

Tabelle 2: Exposure angelehnt an [ISO11]

3. Beherrschbarkeit der Situation durch den Fahrer (C=Controllability)

C0	sicher beherrschbar
C1	einfach zu beherrschen
C2	normalerweise beherrschbar
C3	schwer oder nicht zu beherrschen

Tabelle 3: Controllability angelehnt an [ISO11]

Anhand der Klassifizierung der Gefahrensituation wird mit der nachfolgenden Tabelle das jeweilige ASIL bestimmt.

Die Tabelle wird im Folgenden beschrieben.

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

Tabelle 4: Definition ASIL angelehnt an [ISO11]

Es werden für jedes Szenario die jeweiligen Werte für S=Schadensausmaß, E=Eintrittswahrscheinlichkeit und C=Kontrollierbarkeit aus der Tabelle kombiniert und das jeweilige ASIL bzw. des Qualitätsmanagement Standard (QM) abgelesen. Pauschal sind alle Szenarien mit einem Schadensausmaß von S0 nicht sicherheitsrelevant und haben somit kein ASIL, sondern sind nach dem QM-Standard zu behandeln. Aus diesem Grund sind in der Tabelle alle S0, E0 und C0 Kombinationen nicht aufgeführt. Die ISO 26262 stellt an dieser Stelle Tabellen bereit, die beispielhaft aufzeigen, Wie Szenarien für die einzelnen Parameter S, E, und C aussehen könnten. Aus den Bewertungen der Szenarien werden Sicherheitsziele abgeleitet, die als übergeordnete Sicherheitsanforderungen dienen.

In Teil 9 der ISO 26262 wird die Dekomposition des ASIL beschrieben. Demnach kann z.B. ein ASIL D zerlegt und in niedrigere ASIL unterteilt werden. Das Schema der Dekomposition aus Teil 9, Schaubild 2 ist in Abbildung 6 beispielhaft aufgezeigt.

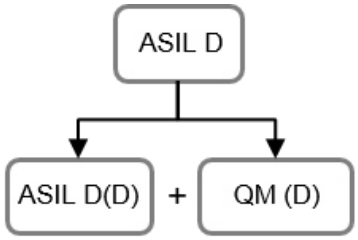


Abbildung 6: Beispiel ASIL Dekomposition angelehnt an [ISO11]

Da nicht alle Systemelemente zwangsläufig das übergeordnete ASIL erfüllen müssen, kann einzelnen Systemelementen ein spezifisches ASIL zugeordnet werden, welches in Abhängigkeit des übergeordneten ASIL steht. Die Bedingungen zur ASIL Dekomposition sind dem Teil 9 der ISO 26262 zu entnehmen.

Functional Safety Concept (FSC)

Im funktionalen Sicherheitskonzept der ISO 26262 werden Anforderungen definiert, die die funktionale Sicherheit des Systems mit Bezug auf die übergeordneten Sicherheitsziele aus der HARA erfüllen sollen. Aus den Sicherheitszielen werden Anforderungen erarbeitet, die das jeweilige ASIL des Sicherheitsziels erben und den jeweiligen Systemelementen zugeteilt werden. Des Weiteren beinhaltet das FSC die Zuteilung von: Fehlererkennungsmaßnahmen, Vermeidungsmaßnahmen, Mechanismen zur Fehlertoleranz, Warnung des Fahrers bei Fehlfunktion, Überführung in sichere Zustände und funktionale Redundanzen zu den einzelnen funktionalen Anforderungen, vorausgesetzt die Sinnhaftigkeit ist gegeben. Das FSC ist am Ende zu verifizieren und zu bewerten.

2.5.3.2 Teil 4: Produktentwicklung auf Systemebene

Im 4. Teil der ISO 26262 wird das ITEM auf der Systemebene weiter spezifiziert. Es werden aus den funktionalen Sicherheitsanforderungen technische Sicherheitsanforderungen abgeleitet, die auf Hard- bzw. Softwareelemente übertragen werden, welche in Teil 5 Produktentwicklung auf Hardwareebene und Teil 6 Produktentwicklung auf Softwareebene detaillierter beschrieben werden. Aus den bestimmten technischen Sicherheitsanforderungen wird die vorläufige Systemarchitektur erstellt, welche das Systemdesign und das technische Sicherheitskonzept (TSC) darstellen. Im TSC wird das System durch Sicherheitsanalysen (FMEA & FTA) auf systematische Fehler untersucht. Erkenntnisse aus den Analysen fließen in die technischen Anforderungen mit ein, die Architektur wird iterativ angepasst. Des Weiteren werden Anforderungen für Produktion, Betrieb, Wartung und Stilllegung des Systems definiert.

Im TSC werden am Ende alle erarbeiteten Erkenntnisse und Maßnahmen bezüglich der funktionalen Sicherheitsziele validiert.

2.5.3.3 Teil 5 & 6: Produktentwicklung auf der Hardware- & Softwareebene

In Teil 5 und 6 werden die Hard- bzw. Softwareebenen spezifiziert. Es werden Hard- bzw. Software spezifische Anforderungen aus dem technischen Sicherheitskonzept abgeleitet. Daraus wird das Design abgeleitet. Die Hard- und Software wird auf verschiedene Weise getestet und validiert.

Auf Teil 5 & 6 wird in dieser Arbeit nicht weiter eingegangen. Weitere Informationen sind in [ISO11] zu finden.

In Abbildung 7 ist das Sicherheitskonzept der ISO 26262 grafisch veranschaulicht.

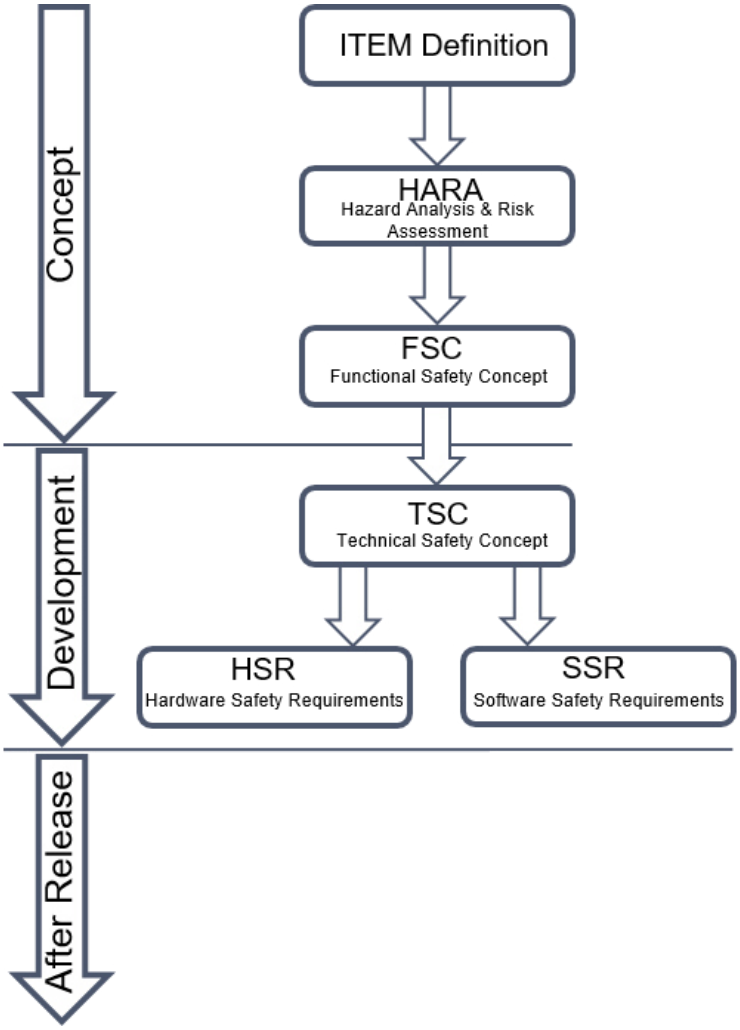


Abbildung 7: Sicherheitskonzept ISO 26262 angelehnt an [ISO11]

Das Sicherheitskonzept der ISO 26262 ist unterteilt in Konzept- und Entwicklungsphase. Die Hauptaktivitäten sind den beiden Phasen zugeteilt.

2.6 Sicherheitsanalysen

Im nachfolgenden Abschnitt wird auf die notwendigen Sicherheitsanalysen eingegangen, die unter anderem für den Entwicklungsprozess für sicherheitsrelevante System im Fahrzeugbau relevant sind. Es gibt Sicherheitsanalysen unterschiedlichster Art, die bekanntesten und am häufigsten eingesetzten sind die FMEA (Failure Mode and Effect Analysis) und die FTA (Fault Tree Analysis), die im Folgenden erklärt werden.

2.6.1 Failure Mode & Effect Analysis (FMEA)

Die Fehler-Möglichkeits- und Einfluss Analyse ist eine induktive Sicherheitsanalyse. Dies bedeutet, dass von einer konkreten Funktion/ Fehler, auf die allgemeinen Ursachen und Wirkungen geschlossen wird. Die FMEA dient der systematischen Identifikation und präventiven Vermeidung von Problemen innerhalb des untersuchten Gegenstandes. Klassisch wird unterschieden zwischen der Produkt- und der Prozess-FMEA. Bei der Produkt-FMEA wird das zu untersuchende Produkt, bzw. das System auf seine Funktionen, deren Zusammenhänge und die Auswirkungen auf die einzelnen Bauteile untersucht. Bei der Prozess FMEA stehen die einzelnen Prozessschritte im Fokus, die nötig sind um das Produkt herzustellen. [Pfe15]

Eine feingranulare Abgrenzung der einzelnen FMEA-Arten bietet [Wer12]. Im Folgenden wird nur die Produkt- bzw. die System-FMEA betrachtet.

Nach [Pfe15] ist die Durchführung der FMEA in fünf Schritte untergliedert.

1. **Strukturanalyse**
2. **Funktionsanalyse**
3. **Fehleranalyse**
4. **Maßnahmenanalyse**
5. **Optimierung**

Die vorgelagerten Schritte aus [Wer12]; “Datensammlung” und “Definition der FMEA Umfänge”, sind hier nicht extra aufgelistet.

Strukturanalyse

Die gesamte Systemstruktur muss bezüglich ihrer Hardware- und Softwarekomponenten und deren Beziehungen zueinander analysiert werden. Das System muss somit runtergebrochen vom Gesamtsystem, über Teilsysteme, bis zu einzelnen Baugruppen und Bauteilen beschrieben werden. Dabei sind der Detaillierungsgrad und die Systemebenen in der Tiefe zu beachten. Des Weiteren müssen die Zusammenhänge der einzelnen Systemelemente bzw. die Verknüpfung abgebildet werden. [Wer12]

Funktionsanalyse

Den einzeln ermittelten Systemelementen werden Funktionen zugeordnet. Das Ziel ist, jedem Bauteil die jeweilige Teilfunktion zuzuordnen. Die Summe der einzelnen Teilfunktionen beschreibt wiederum die Hauptfunktion des Systems und bildet die funktionale Struktur ab. Dabei ist darauf zu achten, dass die Beschreibung der einzelnen Funktionen eindeutig und ausreichend konkret ist. [Wer12]

Fehleranalyse

Die Fehleranalyse ist untergliedert in:

1. Definition Fehlfunktion
2. Definition Fehlerfolge
3. Definition Fehlerursache

Jedem Systemelement ist eine bestimmte Funktion zugeteilt. Demnach resultieren die Fehlfunktionen aus der jeweils identifizierten Funktion und beschreiben z.B. die Nichterfüllung der Funktion. Die Auswirkungen auf nachgelagerte Elemente und Ebenen im System sind durch die Fehlerfolge zu erfassen. Die Ursachen und Gründe der jeweiligen Fehlfunktionen, sind in der Funktion selber, oder in vorgelagerten Systemelementen bzw. Systemebenen begründet. [Wer12]

Maßnahmenanalyse

In der Maßnahmenanalyse werden die Risiken dargestellt, bewertet und dadurch die Entwicklungsschwerpunkte festgelegt, die nötig sind um das System abzusichern. Die Maßnahmen sind unterteilt in Vermeidungsmaßnahmen und Entdeckungsmaßnahmen. Bekannte Maßnahmen werden den jeweiligen Fehlfunktionen zugeordnet und das Risiko abgeschätzt. Des Weitern müssen Maßnahmen erarbeitet werden, welche die fehlerfreie Funktion des Systems ermöglichen. [Wer12]

Vermeidungsmaßnahmen

Vermeidungsmaßnahmen optimieren das Produkt und wirken präventiv. Die Auftretenswahrscheinlichkeit möglicher Fehler wird durch die Vermeidungsmaßnahmen direkt minimiert, somit wirken Vermeidungsmaßnahmen direkt ursachenbezogen. [Wer12]

Entdeckungsmaßnahmen

Durch die Entdeckungsmaßnahmen werden die möglichen Ursachen einer Fehlfunktion bzw. deren mögliche Auswirkung gefunden. Sie wirken ebenfalls präventiv. Die eindeutige, konkrete Beschreibung und Dokumentation ist sehr wichtig. [Wer12]

Verantwortlichkeit schaffen

Des Weiteren wird jede Maßnahme einem Verantwortlichen zugeordnet. Dieser hat sich bis zu einem festen Termin um die Erarbeitung einer Lösung bezüglicher dieser Maßnahme zu kümmern. Alle Bearbeitungsstände und der Maßnahmenstatus müssen dokumentiert werden. [Wer12]

Ursachenbewertung

Anhand von drei Kriterien wird das Risiko der möglichen Fehlfunktionen bewertet

B = Bedeutung

A = Auftretenswahrscheinlichkeit

E = Entdeckungswahrscheinlichkeit

Die Parameter B, A und E können mit Werten von 1-10 bewertet werden. Dabei stellt 1 das geringste und 10 das höchste Risiko dar. [Wer12]

Risikoprioritätszahl RPZ

Das Produkt der Parameter stellt die Risikoprioritätszahl dar, welche sich mit der folgenden Formel berechnen lässt:

Risikoprioritätszahl $RPZ=B*A*E$ (1)

Alternative Erweiterungen der Risikopriorisierung sind in [Wer12] aufgezeigt.

Optimierung

In der Optimierung werden Maßnahmen ermittelt, die zur Verbesserung des Istzustandes des zu untersuchenden Systems beitragen. Außerdem werden diese Maßnahmen auf Wirksamkeit überprüft und dokumentiert. Ist der Maßnahmenstand nicht befriedigend, müssen weitere Maßnahmen erarbeitet werden, die das System verbessern. Danach wird die FMEA wieder aktualisiert. Diese Vorgehensweise wird iterativ durchgeführt, bis das System die benötigte Sicherheit aufweist. [Wer12]

Die fünf Hauptschritte zur Durchführung einer FMEA sind in der nachfolgenden Tabelle noch einmal aufgezeigt. Zusätzlich sind in der Tabelle für jeden Schritt die nötigen Diagramme beschrieben, die benötigt werden, um die FMEA modell-basiert durchzuführen.

Schritt	Vorgehen	Beschreibung	Diagrammtyp	Diagramm SysML
1	Strukturanalyse	Systemanalyse, Dekomposition	Strukturdiagramm	bdd
2	Funktionsanalyse	Zuweisung der Funktionen zu Systemelementen	Verhaltensdiagramm	act
3	Fehleranalyse	Ermittlung von Fehlfunktionen, Ursachen und Auswirkungen	Verhaltensdiagramm	act
4	Maßnahmenanalyse	Ermittlung von Vermeidungs- und Entdeckungs-Maßnahmen	Struktur-/ Verhaltensdiagramme	act, bdd
5	Optimierung	Optimierung der Systemstruktur	Struktur-/ Verhaltensdiagramme	act, bdd

Tabelle 5: FMEA Strukturtabelle

2.6.2 Fault Tree Analysis (FTA)

In diesem Kapitel wird die Fehlerbaumanalyse beschrieben. Die Fehlerbaumanalyse ist ein deduktives Analyseverfahren, dies bedeutet, dass von einem allgemeinen übergeordneten Fehler auf detaillierte Ursachen geschlossen wird.

Grundsätzlich ist das Vorgehen zur Durchführung der FTA ähnlich wie bei der FMEA. Der Ablauf unterteilt sich ebenfalls in fünf Schritte.

1. **Definition des Hauptereignisses**
2. **Definition des zu untersuchenden Systemumfangs**
3. **Erstellung des Fehlerbaums**
4. **Quantitative & qualitative Analyse**
5. **Auswertung und Darstellung des Optimierungsbedarfs**

Definition des Hauptereignisses

Zunächst wird ein Hauptereignis definiert, welches den Einstieg in die Analyse darstellt. Dabei ist das Hauptereignis so zu wählen, dass es einen Ausfall des zu untersuchenden Systems wiederspiegelt. Das Hauptereignis sollte präzise formuliert sein. Bei Industrieanlagen oder im Fahrzeugbau werden diese Hauptereignisse meist aus vorgelagerten Risikoanalysen bestimmt. [Edl15]

Definition des zu untersuchenden Systemumfangs

Der zweite Schritt ist die Abgrenzung des Systems, bzw. die Festlegung des Betrachtungsumfanges für die Analyse. Es sollte das zu betrachtende System und dessen Elemente beschrieben werden. Des Weiteren sind mögliche Fehlerursachen zu bestimmen die im Fehlerbaum von Relevanz sein können.

Weitere Faktoren die nach [Edl15] berücksichtigt werden sollten sind:

- Detaillierungsgrad des Systems
- Systemvarianten
- Einsatzart und Dauer des Systems

Erstellung des Fehlerbaums

Bei der Erstellung des Fehlerbaums wird die “Ursachen- und Wirkungs-Kette” nach dem Top-down Prinzip, von der Wirkung auf die Ursache beschrieben. Angefangen wird mit dem definierten Hauptereignis. Es wird zum Hauptfehler die unmittelbare Ursache im nächst möglichen Element gesucht. In diesem Element kann der beschriebene Fehler mehrere mögliche Ursachen haben, bzw. die Ursachen sind in einem anderen vorgelagerten Element zu finden. Diese mögliche Verkettung von Fehlern bzw. möglichen Fehlerursachen sind durch den Fehlerbaum korrekt abzubilden. Dabei ist immer zu beachten, ob die Ursache des möglichen Fehlers auf derselben Systemebene zu finden ist, oder ob sich dieser evtl. in einer tieferen Ebene befindet. [Edl15]

Dabei sind die möglichen Strukturvarianten nach [Edl15] zu beachten. Der Fehlerbaum kann demnach “Komponentenorientiert”, “Signalpfadorientiert”, oder “Phasen-orientiert” sein. Das bedeutet, dass die Fehlerkette entweder systematisch über die Bauteile, den logischen Signalfluss, oder die Operation und Betriebsarten aufgebaut werden kann.

Anschließend ist der Fehlerbaum mit den logischen Operatoren und Fehlerevents aufzubauen.

In Abbildung 8 sind mögliche Operatoren und Events abgebildet, die in den beiden nachfolgenden Tabellen erklärt werden.

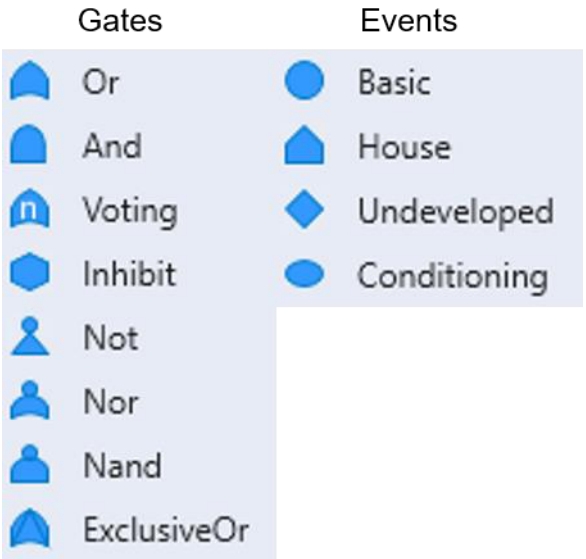


Abbildung 8: Screenshot FTA Software [TOP17], logische Elemente

In Tabelle 6 und Tabelle 7 sind die logischen Elemente zur Erstellung des Fehlerbaums erklärt.

Or	logisches oder
And	logisches und
Voting	bestimmte Anzahl muss eintreten
Inhibit	logisches und, zusätzliche Bedingung
Not	nicht
Nor	nicht-oder
Nand	nicht-und
Exclusive Or	exklusives oder

Tabelle 6: Logische Gatter

Basic	Basis Ereignis des Fehlers
House	„Fehlerhaus“ Ereignis auslösend
Undeveloped	Nicht weiter untersuchtes Ereignis
Conditioning	Ereignis mit bestimmten Bedingungen

Tabelle 7: Logische Ereignisse

Der Fehlerbaum wird mit den dargestellten Operatoren wie beschrieben, vom übergeordneten Ereignis Top-down durch die Systemstruktur aufgebaut. Jede Verknüpfung wird durch das entsprechende logische Gatter beschrieben. Jedes Bauteil, welches durch eine Fehlfunktion das übergeordnete Ereignis beeinflussen kann, wird mit dem entsprechenden Event beschrieben. Dieses Vorgehen wird systematisch durch die komplette, zu untersuchende Systemstruktur durchgeführt.

Quantitative Analyse

Nach der Erstellung des Fehlerbaums ist dieser zu analysieren. Durch die Kennzahlen der einzelnen Elemente (Ausfallart und Ausfallwahrscheinlichkeit), sowie anhand der Struktur des Fehlerbaums, kann die Gesamtausfallwahrscheinlichkeit berechnet werden.

Auswertung über Minimale Schnittmengen (cut-sets)

Das Eintreten der definierten Basisereignisse eines Fehlerbaums führt zum Eintreten des definierten übergeordneten Ereignisses. Die minimale Schnittmenge ist somit die Anzahl der kleinsten Menge an Basisereignissen, die eintreten müssen, damit das übergeordnete Ereignis stattfindet. [Thu04]

Diese und weitere quantitative Parameter werden üblicherweise über die FTA-Software ermittelt, in der auch der Fehlerbaum modelliert wird.

Qualitative Analyse

Bei der qualitativen Fehlerbaumanalyse wird die Struktur des Fehlerbaums untersucht. Dabei wird die Verknüpfungslogik eines Minimalschnittes in der Fehlerbaumstruktur aufgezeigt. Anhand der grafischen Darstellung wird der schwache Fehlerpfad verdeutlicht. Dies dient dem besseren Verständnis und visualisiert an welchen Stellen die Systemstruktur verbessert werden kann. [Edl15]

Auswertung und Darstellung des Optimierungsbedarfs

Die Auswertung des Fehlerbaums und die Optimierung hängen immer vom Einzelfall des Systems und dessen Betrachtung ab. Des Weiteren werden abhängig nach System und Branche, unterschiedliche Anforderungen an die jeweiligen Systemstrukturen und Pfade im Fehlerbaum gestellt. Wichtig in allen Fällen ist die nachvollziehbare Dokumentation der Analysen. Weitere Methoden zum Thema Analysen sind in [Edl15], [Fre15] und [Thu04] zu finden.

In der nachfolgenden Tabelle sind die einzelnen Schritte der FTA dargestellt. Wie bei der FMEA sind in dieser Tabelle ebenfalls die Diagrammtypen und Arten aufgezeigt, die nötig sind um die FTA modell-basiert abzubilden.

Schritt	Vorgehen	Beschreibung	Diagrammtyp	Diagramm SysML
1	Definition Hauptereignis	Systemanalyse	Anforderungsdiagramm	req
2	Definition des zu untersuchenden Systemumfangs	Festlegung des Detaillierungsgrad, Systemtiefe und Systemelemente	Strukturdiagramm	ibd
3	Erstellung des Fehlerbaums	Ermittlung von Fehlfunktionen, Ursachen und Auswirkungen, Darstellung des Fehlerpfades	Strukturdiagramm	ibd
4	Quantitative und Qualitative Analyse	Analyse der Ausfallwahr-scheinlichkeiten und des Fehlerbaums	FTA Fehlerbaum	----------
5	Auswertung & Optimierung	Optimierung der Systemstruktur	Strukturdiagramm	ibd

Tabelle 8: FTA Strukturtabelle

3 Methodik

In diesem Kapitel wird die im Rahmen dieser Masterarbeit entwickelte Methodik aufgezeigt. Die Methodik baut auf den in Kapitel 2 aufgezeigten Grundlagen systematisch auf und verknüpft die einzelnen Bestandteile zueinander. In Abbildung 9 ist der Stand der Technik aus dem Grundlagenkapitel in einem Blockdefinitionsdiagramm hierarchisch dargestellt.

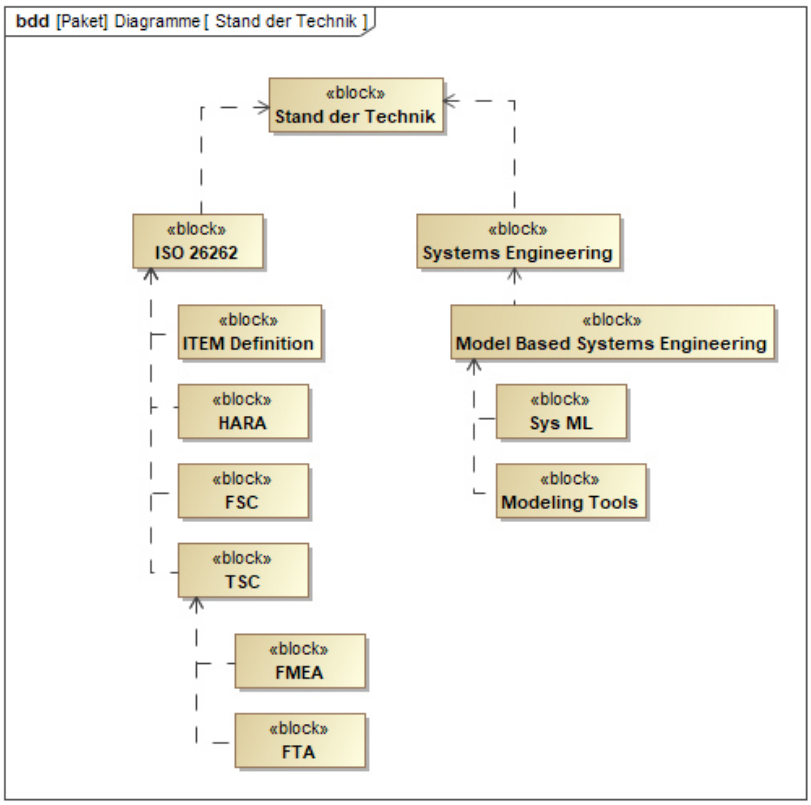


Abbildung 9: Stand der Technik

Die Methodik umfasst, aufbauend auf dem in der ISO 26262 definierten Entwicklungsprozess, die nötigen Aktivitäten für die systematische Entwicklung sicherheitsrelevanter Systeme im Fahrzeugbau auf Basis eines einheitlichen Systemmodells. Das Systemmodell dient in der Methodik als zentrale Informationsdatenbasis, die mit dem Entwicklungsprozess kontinuierlich interagiert.

Auf Basis der Eingangsinformationen des Entwicklungsprozesses wird das Systemmodell an definierten Stellen über Schnittstellen mit Informationen versorgt. Diese Schnittstellen sind an den jeweiligen Stellen in den dementsprechenden Abbildungen durch Icons dargestellt. Die notwendigen Artefakte und Sichten, die für die Abbildung des Entwicklungsprozesses nötig sind, werden mittels SysML Elementen beschrieben. Die systematisch erarbeiteten Erkenntnisse fließen wiederum in den Entwicklungsprozess mit ein und werden stetig erweitert. Durch das Durchlaufen des Entwicklungsprozesses und das parallele, systematische Erstellen von Systemartefakten im Systemmodell findet ein iterativer und bi-direktionaler Informationsaustausch statt, der das Erarbeiten der Prozessziele nachvollziehbar und methodisch abbildet.

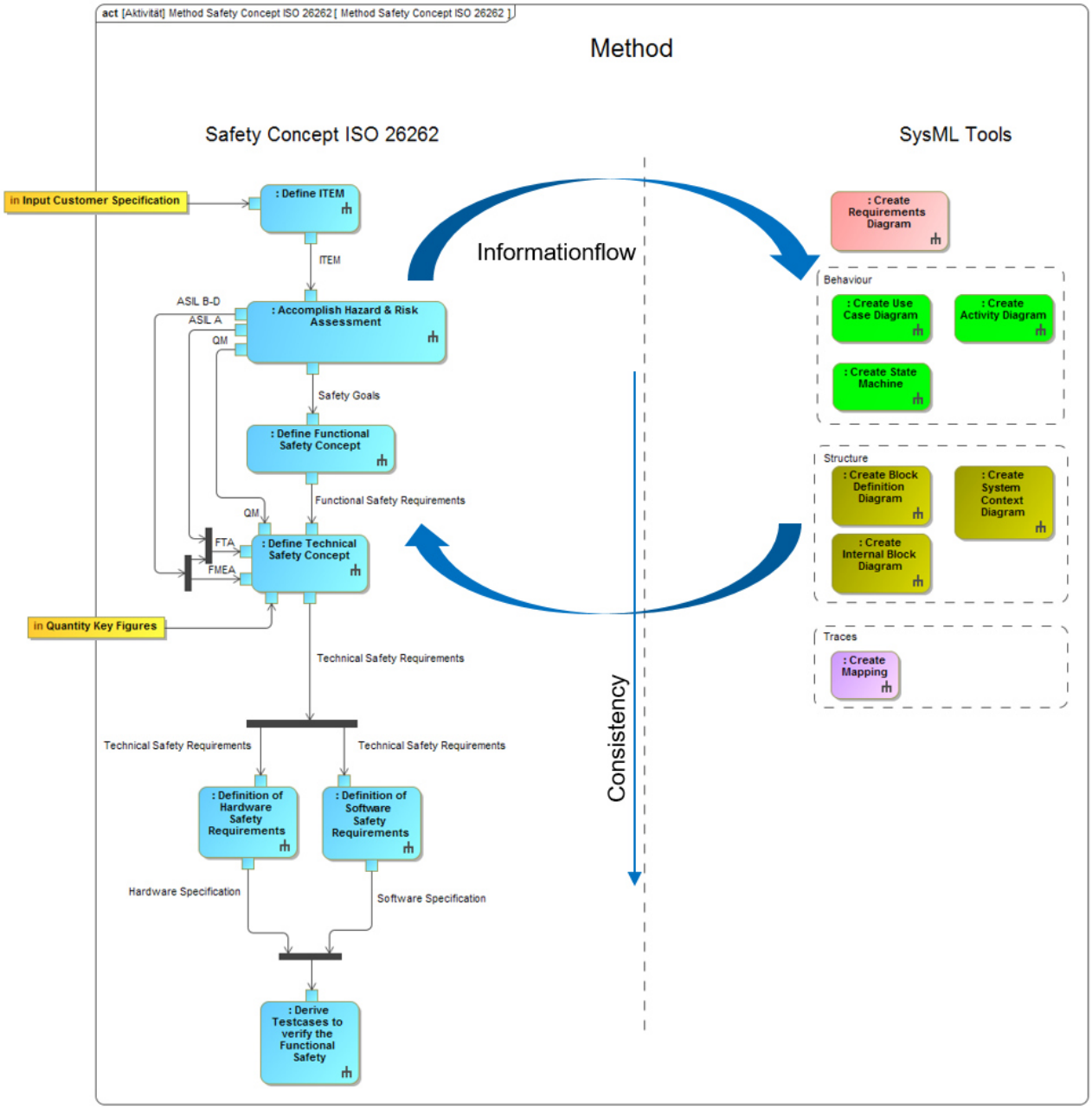


Abbildung 10: Methodik

In Abbildung 10 ist die entwickelte Methodik dargestellt. Auf der linken Seite ist der Entwicklungsprozess nach ISO 26262 durch Aktivitäten abgebildet. Dieser Prozess erhält durch die gelb eingefärbten Ports den nötigen Input. Die blau eingefärbten Aktivitäten stellen die einzelnen detaillierten Prozessschritte dar, die nötig sind, um sicherheitsrelevante Systeme zu entwickeln. Auf der rechten Seite ist das Systemmodell ebenfalls durch Aktivitäten dargestellt und durch die verschiedenen Sichten und Diagrammarten repräsentiert.

Beim Durchlaufen des Prozesses werden innerhalb der einzelnen Aktivitäten an bestimmten Stellen Schnittstellen aufgezeigt, welche die Interaktion mit dem Systemmodell abbilden. Im Folgenden wird die Hauptebene der Methodik in einem kurzen Überblick erklärt. Danach wird die Methodik im Detail beschrieben.

Alle Diagramme sind zur universellen Verwendbarkeit auf Englisch erstellt.

3.1 Überblick

Die Hauptebene des ISO 26262 Prozesses aus Abbildung 10, welcher in Kapitel 2.5.3 beschrieben wurde, ist wie folgt modelliert.

Beginnend mit dem Erhalt der „Customer Spezifications“, in Form eines Lastenhefts, wird die ITEM Definition über die Aktivität „Define ITEM“ eingeleitet. In der ITEM Definition werden das System definiert und der Systemkontext bzw. die Systemgrenzen abgesteckt.

Nachdem das System definiert wurde, wird die Aktivität „Accomplish Hazard & Risk Assessment“ durchgeführt. Die HARA stellt eine Szenarienanalyse dar, aus der die möglichen Risiken abgeleitet und klassifiziert werden.

In der Aktivität „Define Functional Safety Concept“ werden Sicherheitsanforderungen auf funktionaler Ebene in verschiedenen Unteraktivitäten definiert, um die vorher bestimmten Sicherheitsziele zu erfüllen. Des Weiteren wird eine funktionale Architektur aufgebaut und die ASIL Dekomposition durchgeführt.

In der Aktivität „Define Technical Safety Concept“ werden die Fehleranalysen FMEA und FTA durch Aktivitäten die wiederum in Aktionen unterteilt sind detailliert erklärt. Aus diesen Aktionen werden technische Anforderungen und Funktionen definiert, um die Sicherheit des Systems zu spezifizieren. Die Aktivität „Define Technical Safety Concept“ wird in zwei weitere Aktivitäten aufgegliedert.

In den Aktivitäten „Definition of Hardware Safety Requirements“ und „Definition of Software Safety Requirements“ werden Sicherheitsanforderungen und technische Lösungen auf Hardware- bzw. Softwareebene, durch das Durchlaufen von Aktionen erarbeitet. Diese werden in der Aktivität „Derive Testcases to verify the Functional Safety“ verifiziert.

In Abbildung 11 ist eine Matrix abgebildet, welche die einzelnen Rollen innerhalb des Entwicklungsprozesses in Verbindung mit den Hauptprozessschritten darstellt.

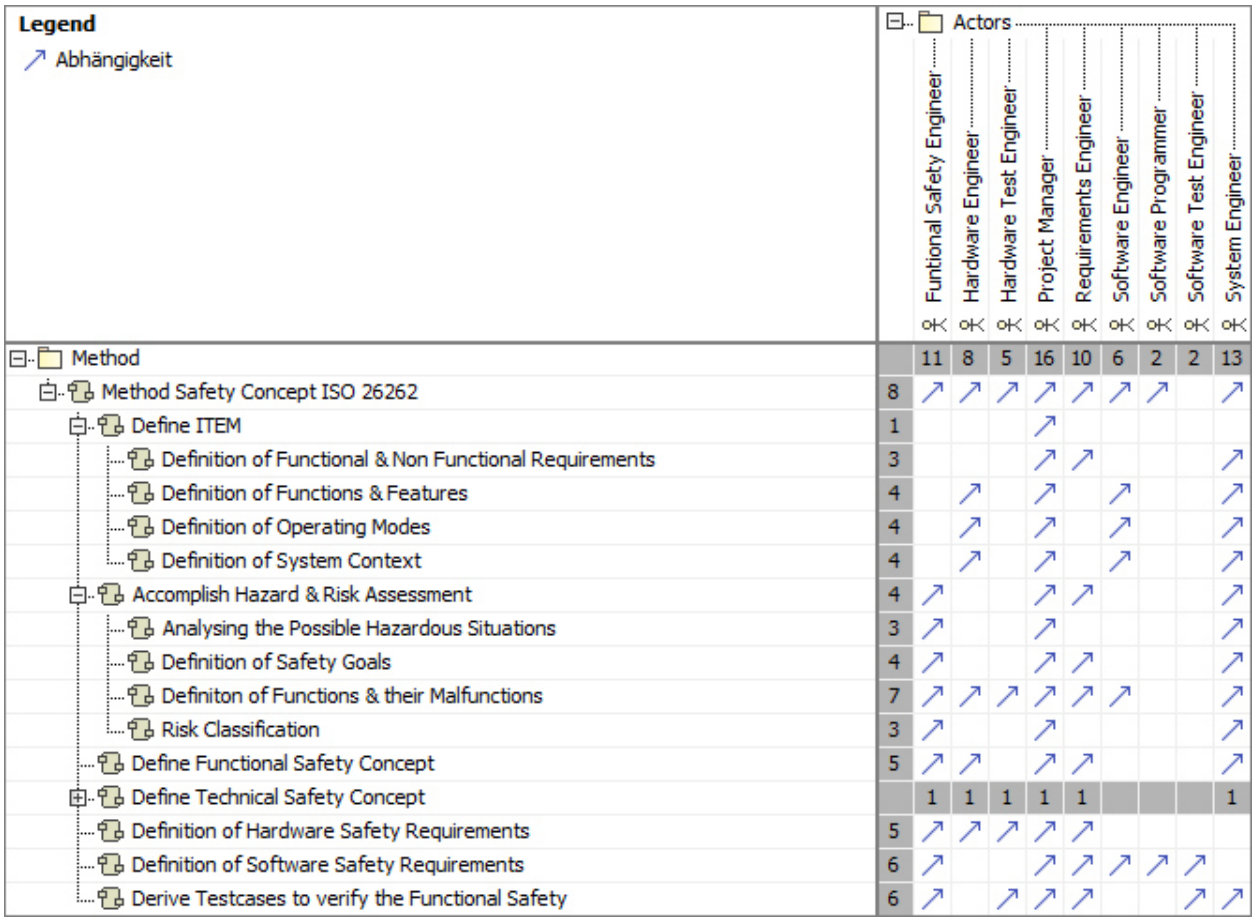


Abbildung 11: Mapping Aktivitäten & Rollen aus dem Entwicklungsprozess

In dieser exemplarischen Darstellung sind der ISO 26262 Entwicklungsprozess und die einzelnen möglichen Rollen der Mitarbeiter aufeinander bezogen. Diese Darstellung veranschaulicht die Verantwortlichkeiten auf einen Blick. In der Tiefe können die einzelnen Prozessaktivitäten beliebig feingranular aufgegliedert werden, womit eine sehr genaue Zuteilung von Arbeitspaketen auf die beteiligten Rollen innerhalb des Systemmodells möglich ist.

3.2 Detaillierte Abfolge der Methodik

Im Folgenden werden die modellierten Ebenen des ISO 26262 Prozesses und deren einzelne Aktivitäten, die wiederum in Unteraktivitäten und Aktionen unterteilt sind, detailliert beschrieben. Dabei wird die Interaktion mit dem Systemmodell mit aufgezeigt. Die dazugehörigen einzelnen Aktivitätsdiagramme der oberen Ebenen sind zum besseren Verständnis im nachfolgenden Kapitelabschnitt teilweise aufgeführt. Für die Darstellungen auf den unteren Detailebenen wird an der entsprechenden Stelle auf den Anhang verwiesen.

3.2.1 ITEM Definition

Die einzelnen Schritte der ITEM Definition sind in Abbildung 12 dargestellt.

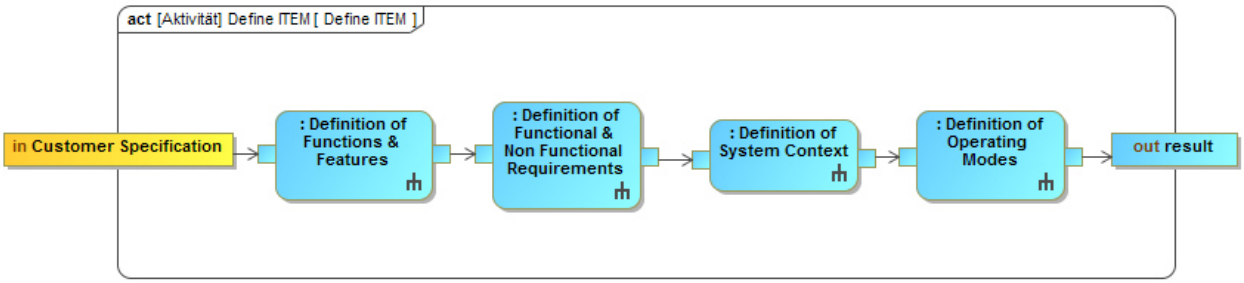


Abbildung 12: Define ITEM

Im ersten Schritt werden die Funktionen und Merkmale des Systems definiert. Die methodische Umsetzung der Aktivität erfolgt durch die in Abbildung 13 dargestellten Aktionen.

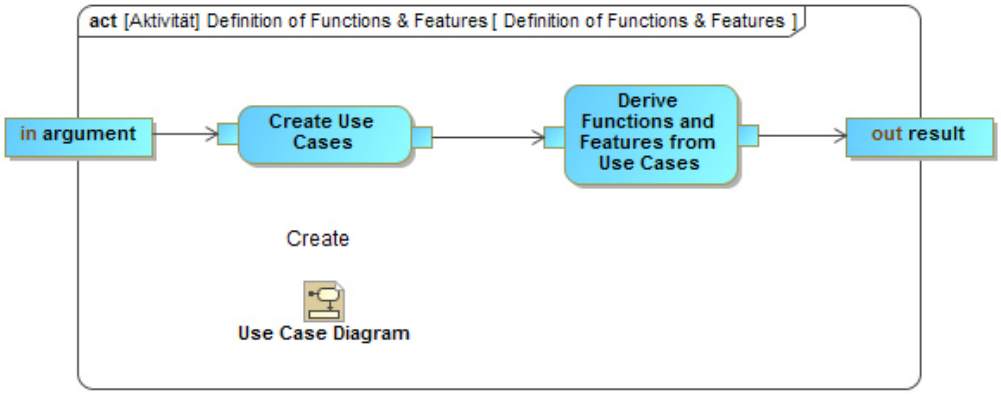


Abbildung 13: Definition of Functions & Features

Es werden Anwendungsfälle erstellt, die im Systemmodell durch ein Anwendungsfalldiagramm repräsentiert werden. Das Diagrammsymbol stellt dabei die Schnittstelle zum Systemmodell dar.

In Abbildung 14 sind die nötigen Schritte zum Aufbau des Anwendungsfalldiagramms aufgezeigt, die durch das Symbol aufgerufen werden. Die einzelnen Schritte sind durch Aktionen in einem Aktivitätsdiagramm dargestellt.

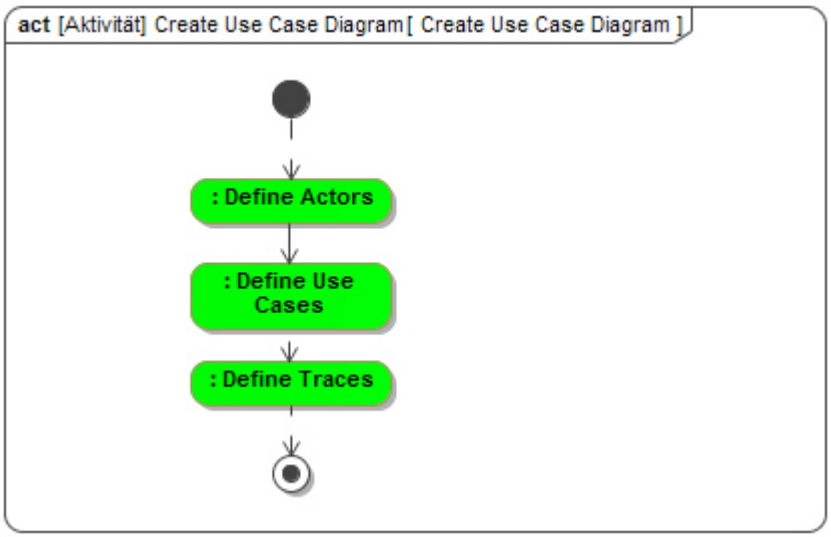


Abbildung 14: Create Use Case

Aus den definierten Anwendungsfällen werden Funktionen und Merkmale abgeleitet, die das System besitzen soll.

Die zweite Aktivität in der Kette aus Abbildung 12 ist in ihrem Aufbau ähnlich, es wird ein Anforderungsdiagramm erstellt, in dem die funktionalen und nicht funktionalen Anforderungen im Systemmodell abgebildet werden.

Des Weiteren wird der Systemkontext in einer separaten Aktivität im Systemmodell erstellt. In der letzten Aktivität der ITEM Definition, werden die Betriebszustände des zu bestimmenden Systems erarbeitet. Diese werden systematisch aus dem zuvor erstellten Anwendungsfalldiagramm abgeleitet. Die Betriebszustände werden durch Zustandsautomaten repräsentiert und im Systemmodell hinterlegt.

Die methodische Erarbeitung der ITEM Definition ist damit abgeschlossen.

Die in der ITEM Definition nicht gezeigten Abbildungen befinden sich im Anhang (Abbildung 64 bis Abbildung 66)

3.2.2 Hazard Analysis & Risk Assessment (HARA)

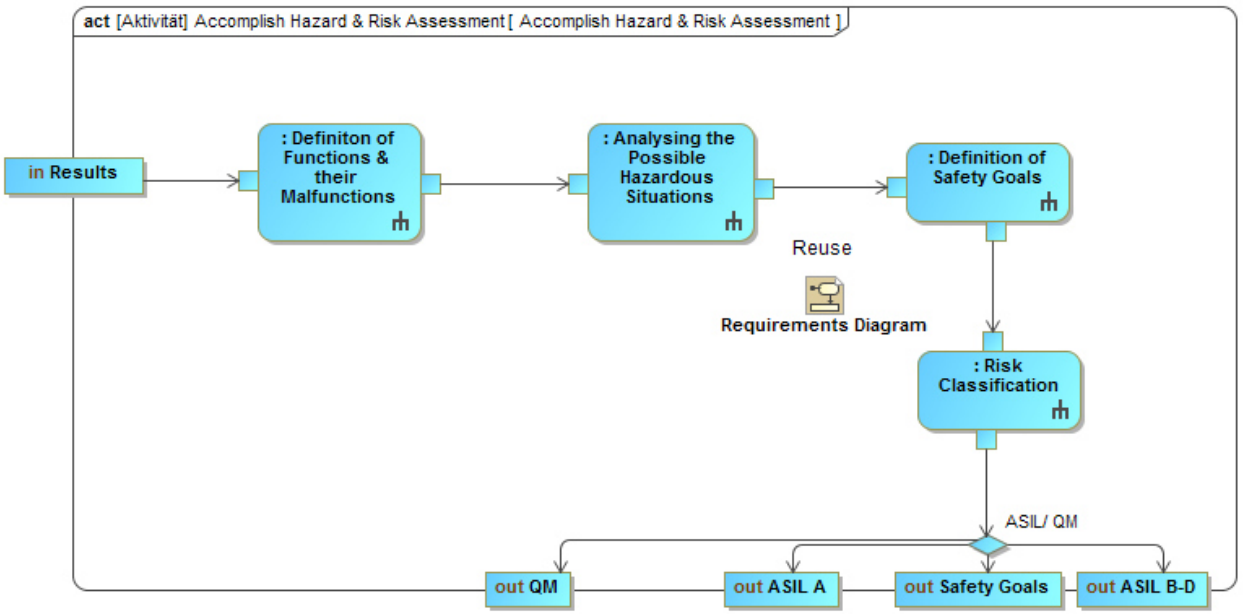


Abbildung 15: Accomplish Hazard & Risk Assessment

In der HARA die in Abbildung 15 dargestellt ist, werden die festgelegten Funktionen aus der ITEM Definition herangezogen, um daraus mögliche Fehlfunktionen abzuleiten, die bei der Anwendung des Systems auftreten könnten.

In der zweiten Aktivität werden aus den Anwendungsfällen der ITEM Definition, aus dem Systemmodell, die möglichen Gefahrensituationen abgeleitet, die bei der Nutzung des Systems entstehen können. Das Anwendungsfalldiagramm wird aus dem Systemmodell wiederverwendet.

Aus den möglichen Gefahrensituationen werden die Sicherheitsziele des Systems abgeleitet, die in jedem Fall erfüllt werden müssen. Dies geschieht durch die Widerverwendung des Anforderungsdiagramms im Systemmodell.

In der nächsten Aktivität, „Risk Classification“, wird das Risiko der möglichen Gefahrensituationen klassifiziert. Das methodische Vorgehen zur Klassifizierung wird im Folgenden durch Abbildung 16 und Abbildung 17 beschrieben.

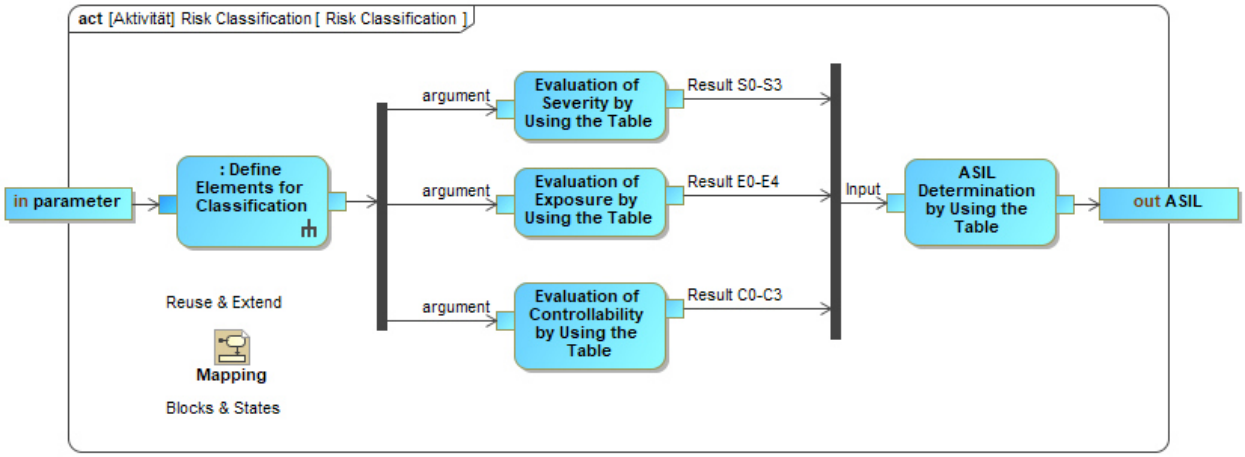


Abbildung 16: Risk Classification

In Abbildung 16 ist der Ablauf der Risikobewertung durch Aktivitäten dargestellt. Zunächst werden Elemente erzeugt, um die Risikobewertung im Systemmodell abbilden zu können. Dieser Vorgang ist durch Abbildung 17 beschrieben.

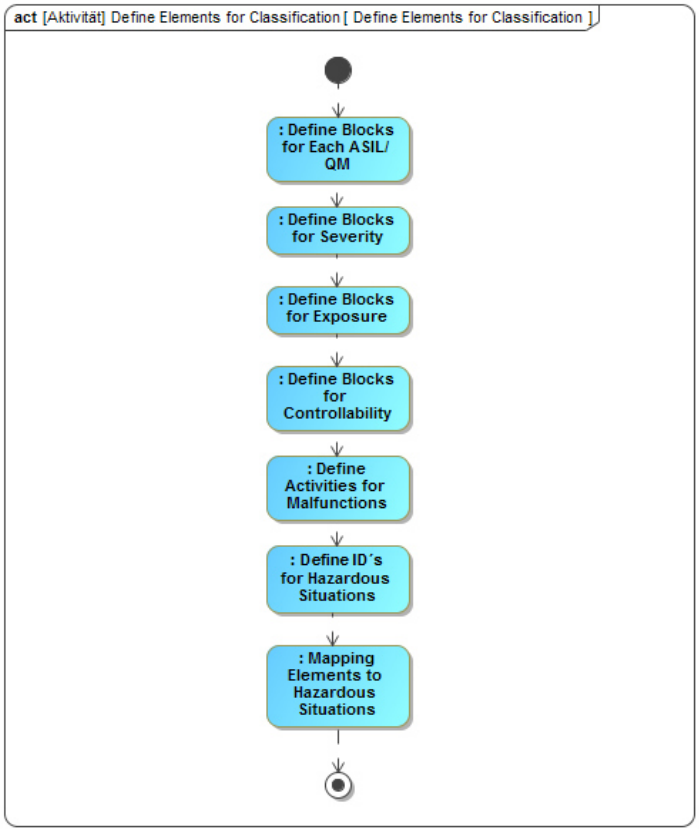


Abbildung 17: Define Elements for Classification

Die definierten Elemente werden in einer Abhängigkeitsmatrix auf einander bezogen, dabei wird die Abhängigkeitsmatrix aus den Betriebszuständen wiederverwendet und erweitert.

Der Ablauf der einzelnen Aktionen, die nötig sind, um das Mapping durchzuführen ist in Abbildung 18 als Aktivität abgebildet. Dieser Ablauf ist für weitere Mappings analog gültig.

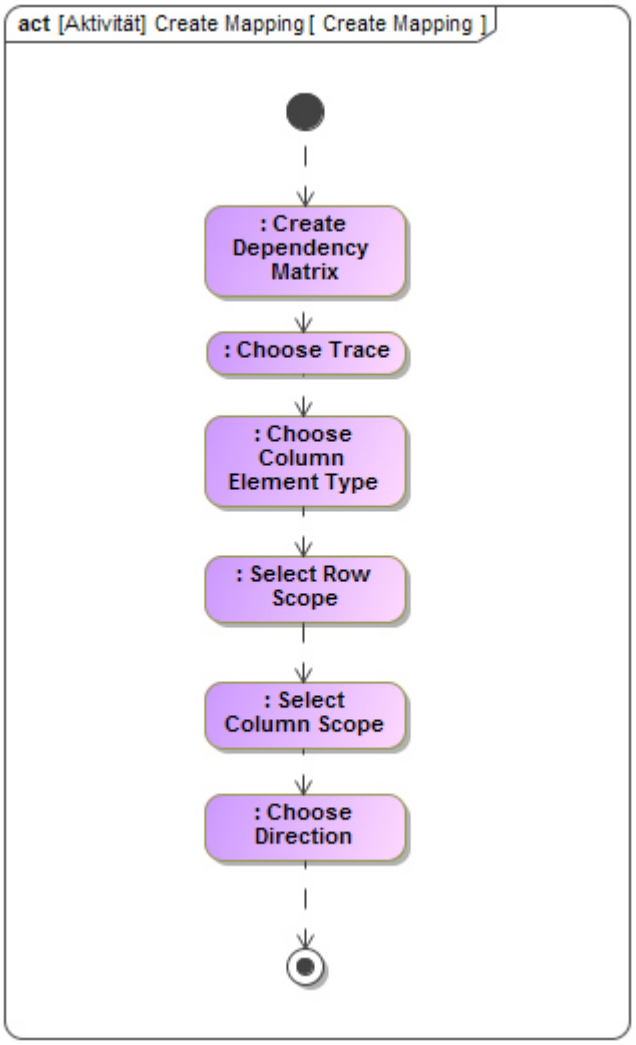


Abbildung 18: Create Mapping

Mit der erstellten Abhängigkeitsmatrix können die einzelnen Szenarien wie in Kapitel 2.5.3 ISO 26262 beschrieben, bewertet werden. Das ASIL ist somit im Systemmodell für jedes Gefahrenszenario systematisch definiert und nachvollziehbar hinterlegt. Die methodische Erarbeitung der HARA ist an dieser Stelle abgeschlossen. Alle nicht weiter aufgeschlüsselten Aktivitäten sind im Anhang (Abbildung 67 bis Abbildung 69) hinterlegt.

3.2.3 Functional Safety Concept

In Abbildung 19 sind die Aktivitäten zur Definition des funktionalen Sicherheitskonzeptes dargestellt. Es werden sicherheitsrelevante Anforderungen definiert und die vorläufige Architektur des Systems auf funktionaler Ebene spezifiziert. Des Weiteren wird die ASIL Dekomposition durchgeführt.

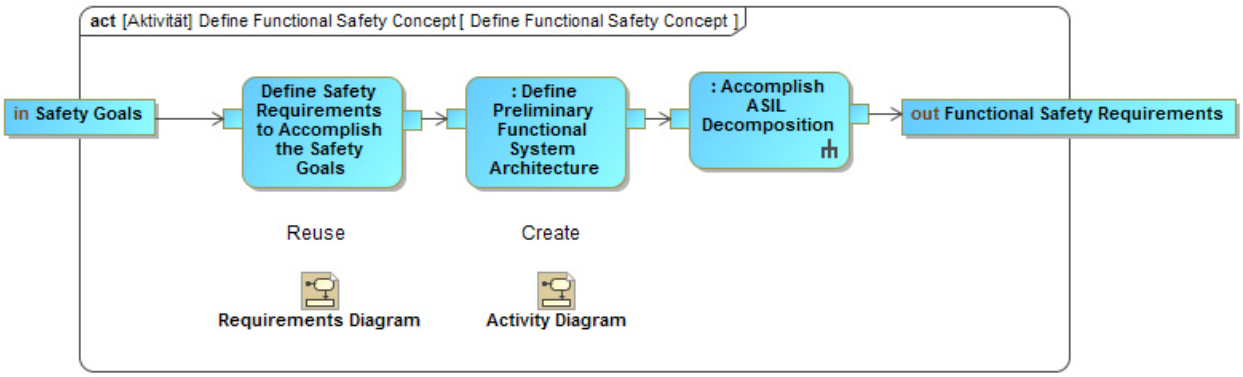


Abbildung 19: Define Functional Safety Concept

Die vorläufige funktionale Architektur wird durch verknüpfte Aktionen in einem Aktivitätsdiagramm beschrieben. In Abbildung 20 ist das Vorgehen zur Erstellung eines Aktivitätsdiagramms exemplarisch dargestellt. Alle weiteren Aktivitäten zur Diagramm Erstellung befinden sich im Anhang (Abbildung 78 und Abbildung 79)

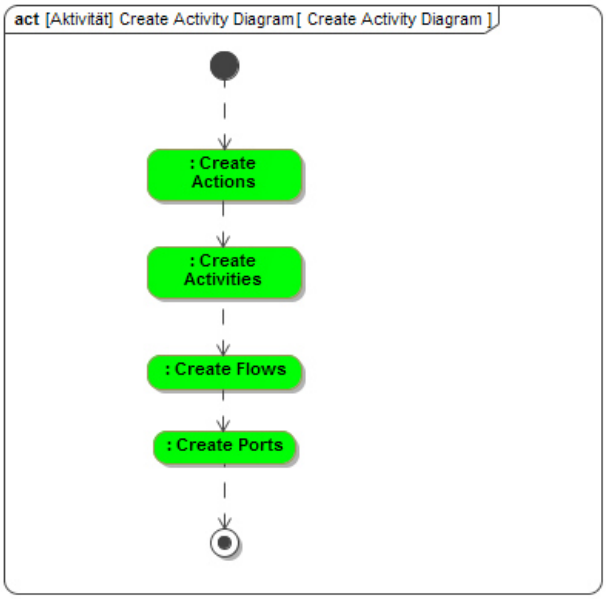


Abbildung 20: Create Activity

Das Diagramm wird als Sicht im Systemmodell erstellt. Das Anforderungsdiagramm wird durch funktionale Sicherheitsanforderungen erweitert. Auf Basis von funktionalen Sicherheitsfunktionen werden funktionale Sicherheitsanforderungen abgeleitet. Diese werden in das Anforderungsdiagramm mit aufgenommen und werden mit den Sicherheitsfunktionen verknüpft.

Die funktionale Architektur wird der ASIL Dekomposition unterzogen. Diese wird in der Methode in einem weiteren Mapping durchgeführt und dargestellt. In Abbildung 21 ist das Vorgehen der ASIL Dekomposition beschrieben.

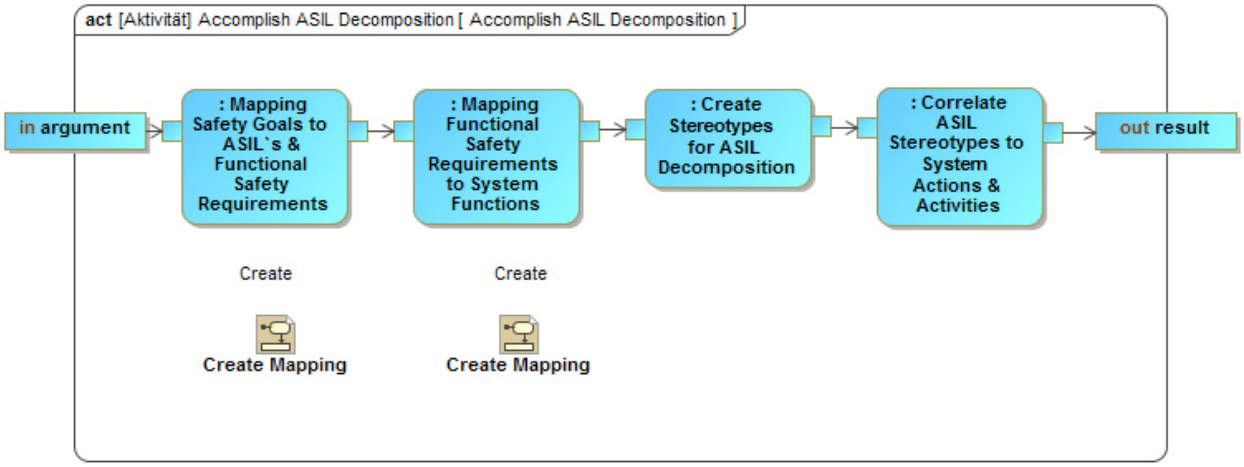


Abbildung 21: Accomplish ASIL Decomposition

Es sind an dieser Stelle zwei Mappings nötig, um die Dekomposition durchzuführen. Zunächst werden die Safety Goals auf die ASIL und funktionalen Anforderungen bezogen. In einem weiteren Mapping werden die funktionalen Anforderungen mit den Systemfunktionen in Bezug gebracht. Dabei wird das ASIL aus dem ersten Mapping, welches auf die jeweilige funktionale Anforderung bezogen wurde, auf die Systemfunktion im zweiten Mapping vererbt.

Nach der ASIL Dekomposition ist das FSC abgeschlossen. Alle Erkenntnisse fließen in das Systemmodell und werden im nächsten Schritt, dem TSC, weiter spezifiziert.

3.2.4 Technical Safety Concept

In Abbildung 22 ist das technische Sicherheitskonzept methodisch aufgezeigt.

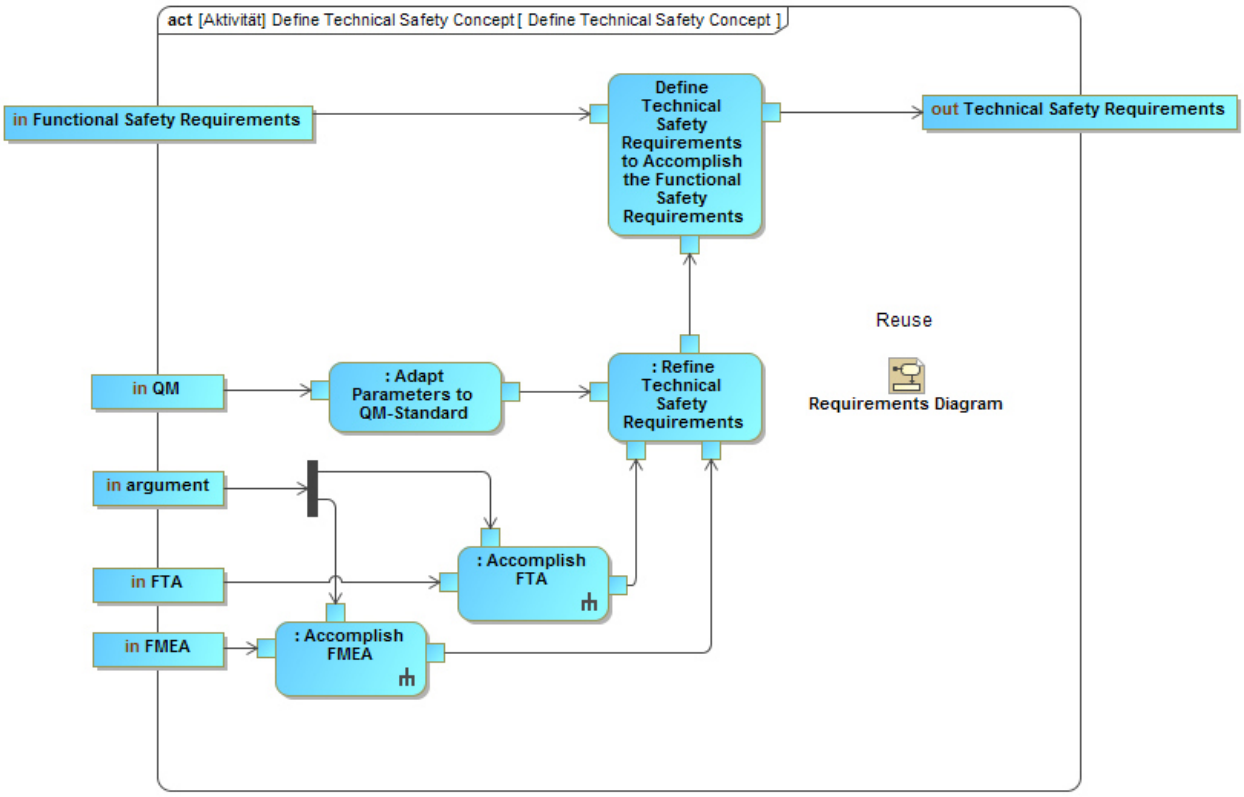


Abbildung 22: Define Technical Safety Concept

Im TSC wird die logische Architektur des Systems, welche die Umsetzung der Systemfunktionen aufzeigt, erstellt. Auf Basis der funktionalen und logischen Architektur können die Sicherheitsanalysen FMEA und FTA durchgeführt werden.

Je nach Ergebnis aus der HARA werden die einzelnen Sicherheitsanalysen FMEA und FTA durchgeführt. Nach den Sicherheitsanalysen werden die technischen Anforderungen im Anforderungsdiagramm, im Systemmodell nach Bedarf angepasst.

3.2.5 FMEA

In Abbildung 23 sind die fünf Standard-Schritte zur Durchführung der FMEA als Aktivitäten modelliert.

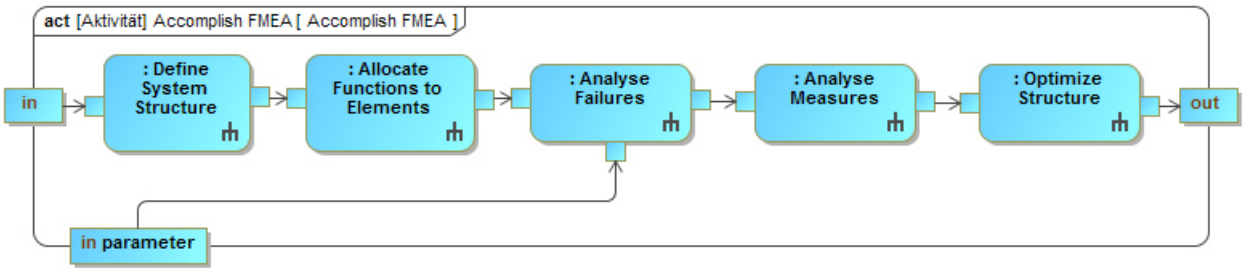


Abbildung 23: FMEA

Der erste Schritt in der Aktivitätskette ist das Beschreiben der Systemstruktur. In Abbildung 24 ist das methodische Vorgehen hierfür dargestellt.

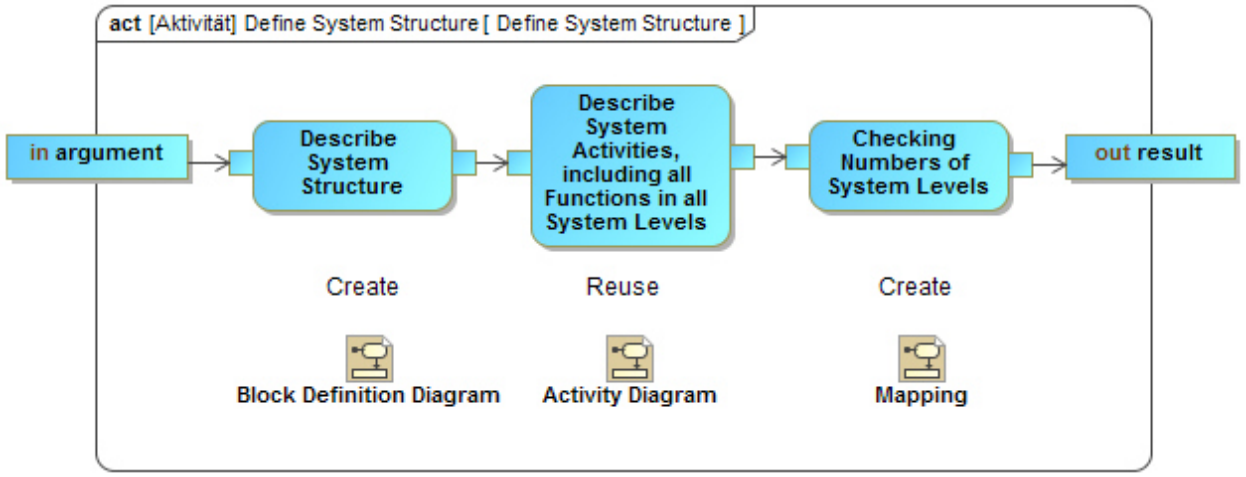


Abbildung 24: Define System Structure

Um die Systemstruktur zu beschreiben, wird ein bdd erstellt. Das bbd wird verwendet, um Systemstrukturen zu beschreiben und Zusammenhänge zwischen den Bauteilen darzustellen. Die einzelnen Elemente und Bauteile sind durch Blöcke definiert, die strukturell hierarchisch angeordnet und über Verbindungen verknüpft werden.

In der zweiten Aktion wird das Verhalten des Systems über ein Aktivitätsdiagramm beschrieben. Dieses Diagramm wird aus dem FSC widerverwendet. Die Möglichkeit der Wiederverwendung und Erweiterung einzelner Diagramme im Systemmodell, stellt einen erheblichen Mehrwert der Methode dar.

Bei der Darstellung des Systems in der Verhaltensperspektive ist darauf zu achten, dass alle Bauteilfunktionen im Aktivitätsdiagramm als Aktivitäten bzw. Aktionen dargestellt sind. Da die FMEA über das Aktivitätsdiagramm abgeleitet wird, werden nur die Funktionen der Bauteile berücksichtigt, die auch explizit modelliert sind.

Es wird eine Abhängigkeitsmatrix erstellt, bei der die Bauteile und die Aktivitäten aufeinander gemappt werden. Dieses Mapping dient zur Kontrolle. Jedem Bauteil werden die Aktivitäten zugewiesen, die die jeweilige Funktion des Bauteils repräsentieren. Somit sind die Bauteile im Systemmodell mit ihren Funktionen verknüpft. Des Weiteren muss die Anzahl der Systemebenen überprüft werden. Es ist zu prüfen ob und wie viele Unterfunktionen eine Funktion besitzt. Diese Analyse ist für die FMEA essenziell, da alle Funktionen für die Analyse bekannt sein und erfasst werden müssen.

In Abbildung 25 sind die Aktionen abgebildet, mit denen das Mapping des Aktivitätsdiagramms und dem FMEA Formblatt durchgeführt wird.

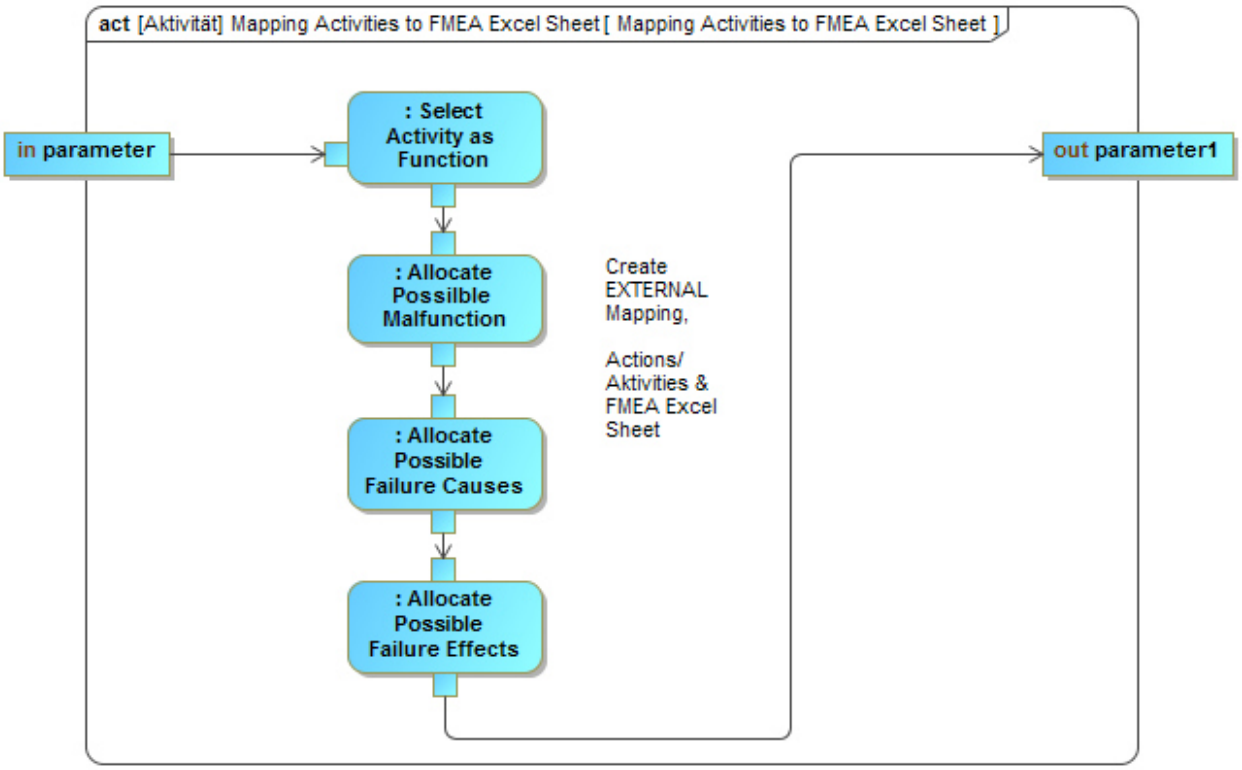


Abbildung 25: Mapping Activities to FMEA Excel Sheet

Es sind alle Aktivitäten als Funktionen in das FMEA Formblatt einzutragen. Für jede Funktion muss die mögliche Fehlfunktion identifiziert werden.

Der nächste Schritt ist das systematische Durchlaufen des Aktivitätsdiagramms aus der funktionalen Architektur des Systems. Jede Fehlfunktion hat eine Auswirkung auf die nachfolgende Funktion, daraus entsteht ein Pfad, der die Fehlerfortpflanzung beschreibt.

Des Weiteren hat jede Fehlfunktion eine Ursache, die über den Ursachenpfad ermittelt wird. Um diese Pfade zu ermitteln, müssen die einzelnen Aktivitäten systematisch analysiert und jede Funktion, Fehlfunktion, Auswirkung und Ursache im FMEA Formblatt festgehalten werden. Die Erkennungs- und Vermeidungsmaßnahmen werden ebenfalls ermittelt und in das FMEA Formblatt eingetragen.

Nachdem die Aktivitätsstruktur systematisch analysiert und alle Parameter ermittelt sind, werden die Fehlfunktionen den jeweiligen Bauteilen zugeordnet. Die Bauteile stellen wiederum die möglichen Fehlerquellen dar. Dieses Vorgehen ist in Abbildung 26 abgebildet. Dies wird mit der Abhängigkeitsmatrix durchgeführt, die am Anfang der FMEA erstellt wurde.

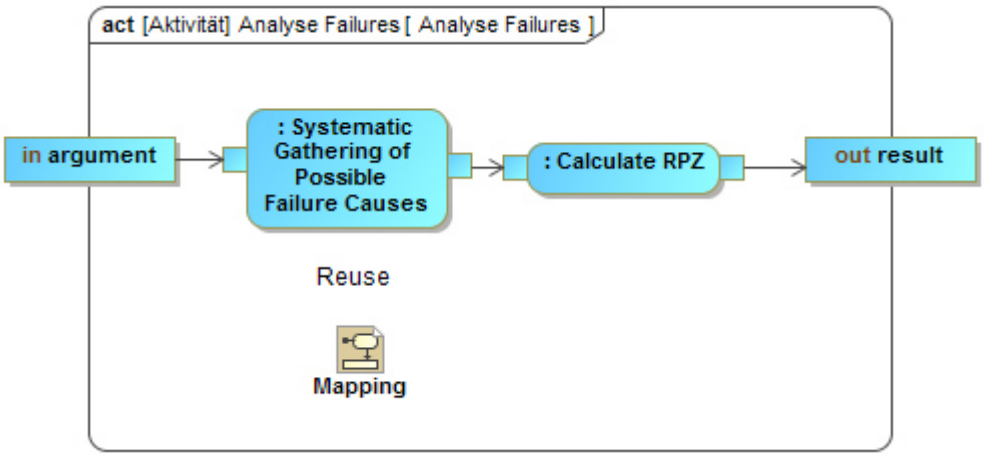


Abbildung 26: Analyse Failures

Die FMEA wird quantitativ bewertet. Dies geschieht über die Risikoprioritätszahl (RPZ) Berechnung. Die notwendigen Input Parameter, um diese Berechnung durchführen zu können, stammen aus Spezifikationen bzw. Bauteilkatalogen.

Der RPZ-Wert wird ausgewertet. Ist er oberhalb einer vordefinierten Schwelle, müssen Verbesserungen an Bauteilen bzw. Systemstruktur vorgenommen werden. Es findet eine Anpassungsschleife statt, die Einfluss auf die Systemstruktur, Anforderungen und die Bauteilauswahl haben kann. Nach der Optimierungsschleife wird die FMEA erneut durchgeführt und ausgewertet.

Liegt die RPZ nach der ersten FMEA im definierten Bereich, wird keine Optimierung durchgeführt. Der Status des Systems wird auf „Monitoring“ gesetzt und erst wieder bewertet, wenn Änderungen am System stattgefunden haben.

Zuletzt müssen Erkennungs- und Vermeidungsmaßnahmen erarbeitet werden, die als Fehlerprävention dienen. Alle notwendigen erarbeiteten Maßnahmen fließen in die jeweiligen Systemebenen mit ein. Das System und die Anforderungen werden an die gewonnenen Erkenntnisse angepasst. Die übrigen FMEA Darstellungen befinden sich im Anhang (Abbildung 73 und Abbildung 74).

3.2.6 FTA

Die fünf Schritte zur Durchführung der FTA sind in Abbildung 27 durch Aktivitäten dargestellt.

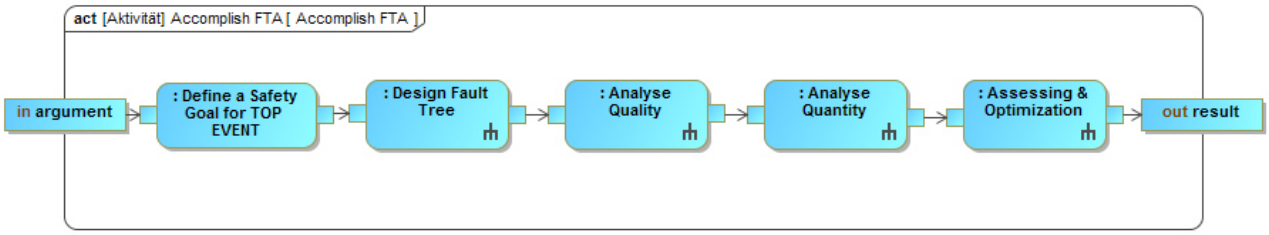


Abbildung 27: FTA

In den Aktivitäten der FTA wird zunächst ein Sicherheitsziel aus der HARA als Top Event festgelegt, das als Hauptfehler definiert wird und dazu dient, das System zu untersuchen.

In Abbildung 28 sind die methodischen Schritte zur Erstellung des Fehlerbaums beschrieben.

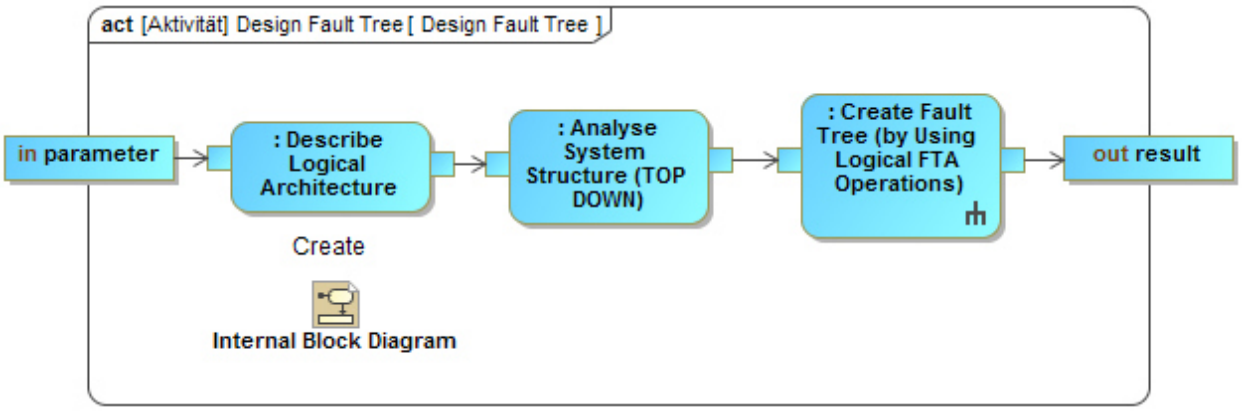


Abbildung 28: Design Fault Tree

Das interne Blockdiagramm wird erstellt, um die logische Architektur des Systems darzustellen. Anhand des ibd's wird das System auf der logischen Ebene systematisch analysiert. Dabei stellen die logischen Blöcke die Bauteile dar, die über verschiedene Flüsse Informationen austauschen. Das Top Event ist der Fehler, der am Ende des Systems auftritt und das Eintreten eines Gefährdungszustandes beschreibt.

In Abbildung 29 sind die methodischen Schritte zur Erstellung des Fehlerbaums beschrieben.

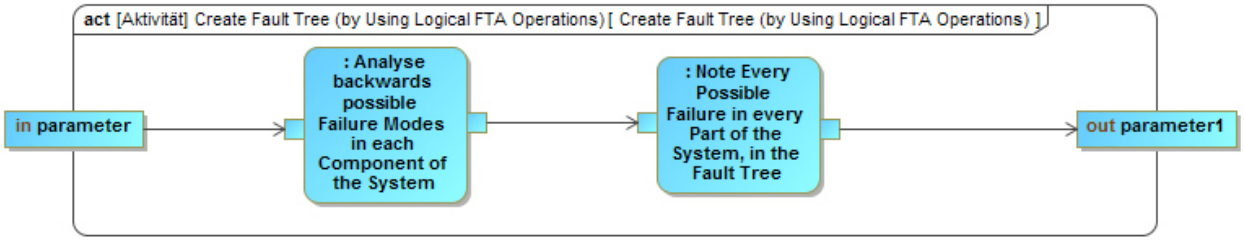


Abbildung 29: Create Fault Tree

Das System wird im Top-down Prinzip analysiert. Beginnend am Ausgang wird jedes Element und dessen Ein- und Ausgänge der logischen Architektur systematisch betrachtet. Jede mögliche Stelle im ibd, an der ein Fehler auftreten könnte, wird mit einem „Event“ gekennzeichnet. Dabei wird der Fehlerpfad systematisch analysiert und aufgezeigt.

Aus dem ibd mit dem beschriebenen Fehlerpfad und den möglichen Fehlerevents wird der Fehlerbaum erstellt. Der Fehlerbaum wird in einer separaten FTA-Software [Top17] beschrieben. Beginnend mit dem Top-Event wird der Fehlerbaum von oben nach unten systematisch entsprechend dem methodisch beschriebenen Fehlerpfad aufgebaut. Dabei werden die normativ definierten FTA-Verknüpfungen verwendet.

Nach Fertigstellung des Fehlerbaums ist dieser qualitativ zu bewerten. Die Auswirkungen möglicher eintretender Fehler auf die Hauptfunktion sind zu analysieren.

Nach der qualitativen Bewertung wird der Fehlerbaum quantitativ bewertet. Die hierzu nötigen Input Parameter für die Ausfallwahrscheinlichkeiten der Bauteile sind den jeweiligen Spezifikationen bzw. Bauteilkatalogen zu entnehmen.

Es folgt die Bewertung des Fehlerbaumes. Sind die Eintrittswahrscheinlichkeiten über halb eines gesetzten Schwellwertes, ist es vorerst nicht notwendig die Architektur, bzw. einzelne Bauteile des Systems zu ändern. Die Architektur wird auf den Status „Monitoring“ gesetzt und beobachtet.

Falls das Resultat der quantitativen Bewertung höher ausfällt als der gesetzte Schwellwert, wird eine Optimierungsschleife eingeleitet. Es sind Optimierungen der Architektur, bzw. der Bauteile möglich, um das Einhalten der Sicherheitsziele zu gewährleisten. Die Änderungen werden in die FTA übertragen, und dann wird die quantitative Auswertung der FTA wiederholt.

Alle weiteren Diagramme, welche die FTA beschreiben, befinden sich im Anhang (Abbildung 70 bis Abbildung 72).

3.2.7 Hardware & Software Safety Requirements

Alle erarbeiteten Erkenntnisse des technischen Sicherheitskonzepts fließen als technische Sicherheitsanforderungen in die Hardware & Software Ebene mit ein. Die Sicherheitsziele werden mit Hardware spezifischen Anforderungen auf Bauteilebene und mit Software spezifischen Anforderungen auf Software Ebene verfeinert um die übergeordneten funktionale Sicherheit zu gewährleisten.

Aus den erarbeiteten Erkenntnissen werden Testfälle für Software, Hardware und Anforderungen abgeleitet.

Die Durchführung der Hardware und Software Spezifizierung ist im Anhang (Abbildung 75 bis Abbildung 77) exemplarisch dargestellt.

Die Methodik wird an dieser Stelle nicht weiter ausgeführt.

4 Validierung der Methodik

In diesem Kapitel ist die Validierung der Methodik dargestellt. Die einzelnen Schritte, welche in Kapitel 3 erläutert wurden um die Methodik zu beschreiben, werden anhand eines Beispielsystems aus der Fahrzeugtechnik durchlaufen und validiert.

Die Darstellung und Spezifizierung des Systems ist exemplarisch und somit nicht als vollständige Entwicklung eines Systems zu verstehen. Es steht immer die systematische Anwendung der Methodik im Vordergrund.

4.1 Beispiel System Elektromechanische Überlagerungslenkung

Als Beispielsystem wird eine Überlagerungslenkung herangezogen. Die Überlagerungslenkung ist teilweise an den in [Wal06] beschriebenen Aufbau angelehnt. In diesem System soll die Lenkbewegung des Fahrers erfasst und geschwindigkeitsabhängig überlagert werden. Das in dem Beispiel aufgezeigte System hat keinen Anspruch auf Vollständigkeit oder Korrektheit der einzelnen Funktionen, vielmehr soll es die Anwendung der in Kapitel 3 aufgezeigten Methodik an einem einfachen Beispiel veranschaulichen.

4.2 ITEM Definition

Entsprechend dem ISO 26262 Prozess (siehe Abschnitt 3.2.1) wird im ersten Schritt das „ITEM“ über die Funktionen beschrieben. Die entsprechenden Funktionen werden aus den Anwendungsfällen abgeleitet. In Abbildung 30 ist das Anwendungsfalldiagramm dargestellt.

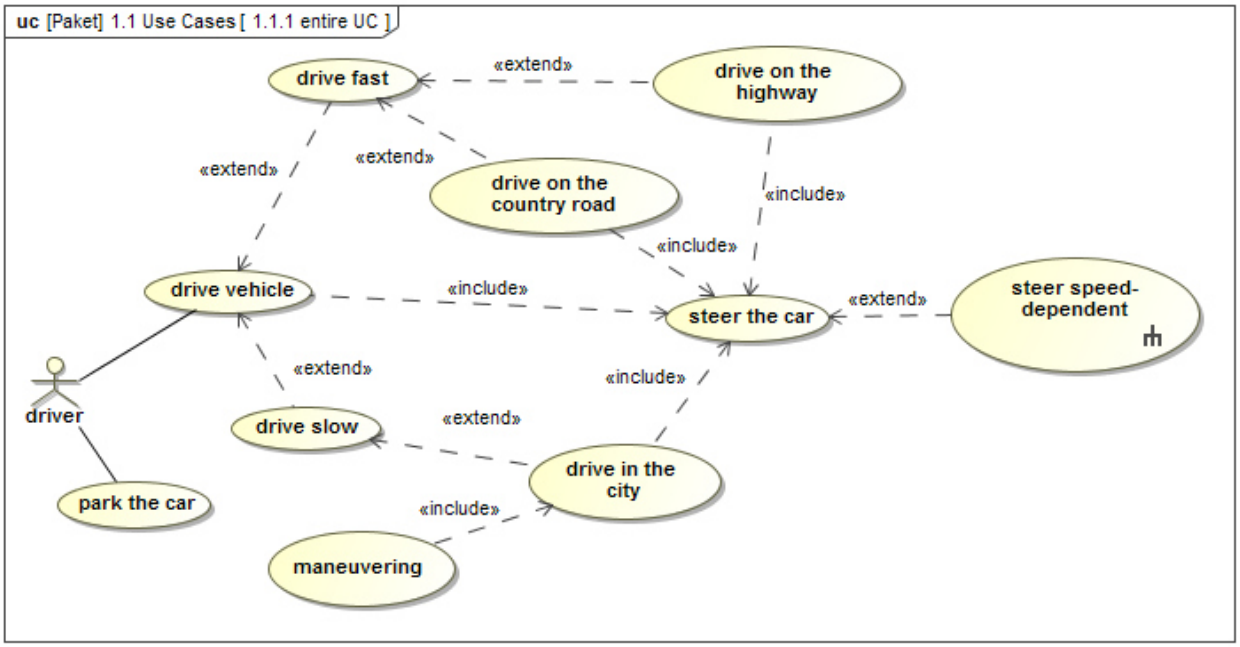


Abbildung 30: Create Use Cases, ITEM Definition

Im Anwendungsfalldiagramm sind die Anwendungsfälle dargestellt, in denen der Akteur, in diesem Fall der Fahrer, mit dem zu definierenden System agiert. Die übergeordneten Anwendungsfälle sind „Fahrzeug fahren“ (engl. drive vehicle) und „Fahrzeug lenken“ (engl. steer the car). Ausgehend vom Anwendungsfall „drive vehicle“ wird im Anwendungsfalldiagramm das Fahrverhalten des Fahrers über weitere konkrete Anwendungsfälle beschrieben. Es wird beschrieben, wie der Anwender fahren möchte. Der resultierende Hauptanwendungsfall ist das „geschwindigkeitsabhängige Lenken“ (engl. steer speed dependent) des Fahrzeuges.

In Abbildung 31 sind die Anwendungsfälle der Fahrzeuglenkung hervorgehoben. Aus den Anwendungsfällen wird exemplarisch die Funktion „geschwindigkeitsabhängiges Lenken“ abgeleitet und mit dem Stereotyp „General Requirement“ als übergeordnete funktionale Anforderung beschrieben.

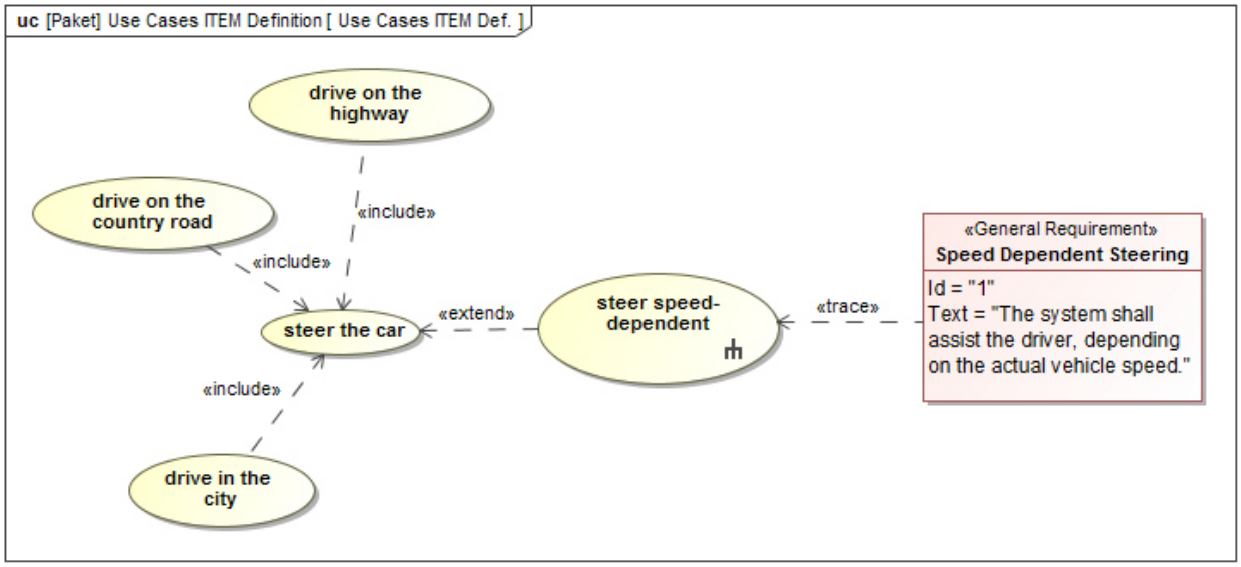


Abbildung 31: Use Cases for Steering

In Abbildung 32 und Abbildung 33 werden aus den Anwendungsfällen funktionale Anforderungen abgeleitet, die die Funktion aus Abbildung 31 detaillierter beschreiben und sicherstellen sollen.

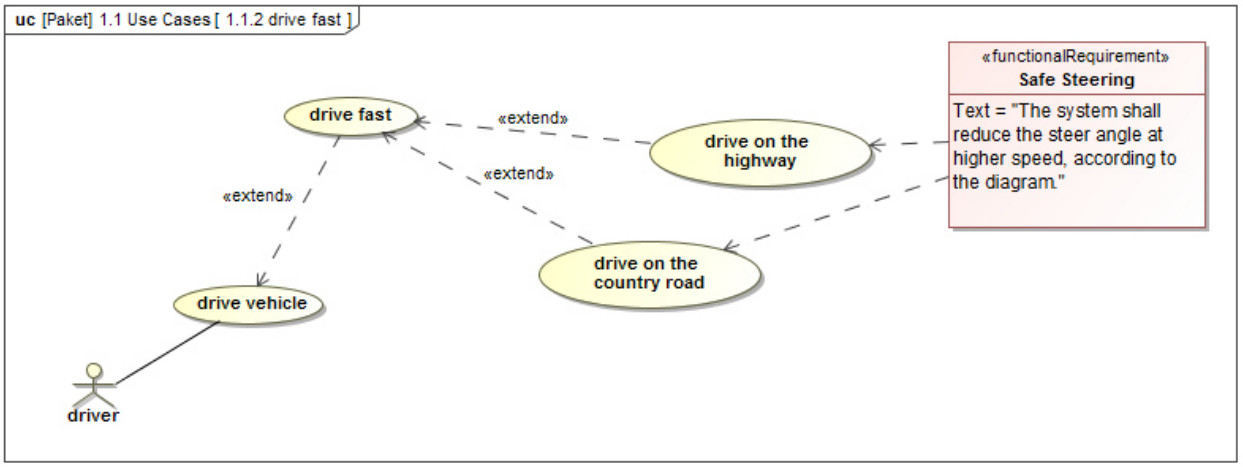


Abbildung 32: Derive Requirement from Use Case „drive fast“

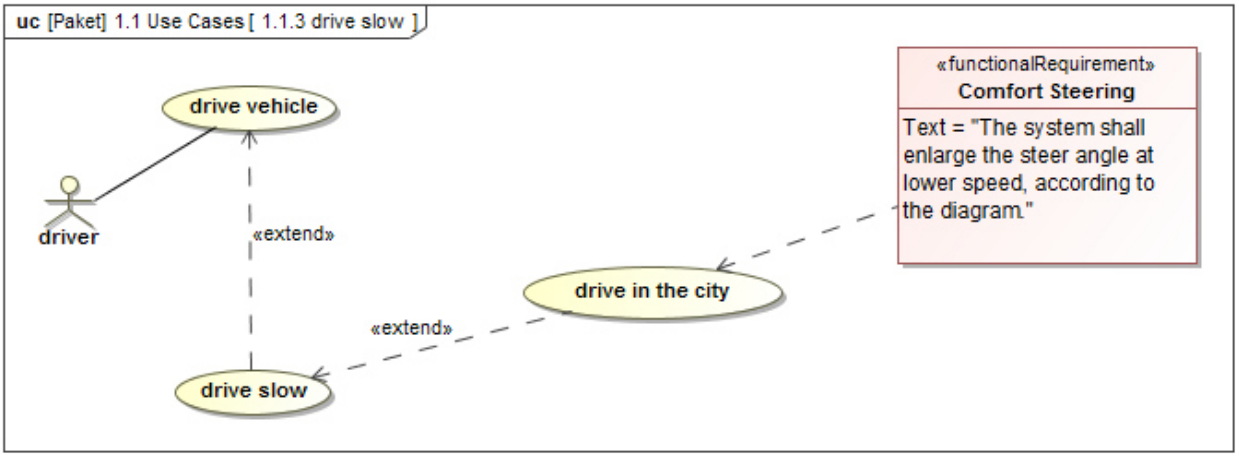


Abbildung 33: Derive Requirement from Use Case „drive slow“

In Abbildung 34 sind die funktionalen sowie nicht-funktionalen Anforderungen, welche die Hauptfunktion spezifizieren, in einem Anforderungsdiagramm dargestellt.

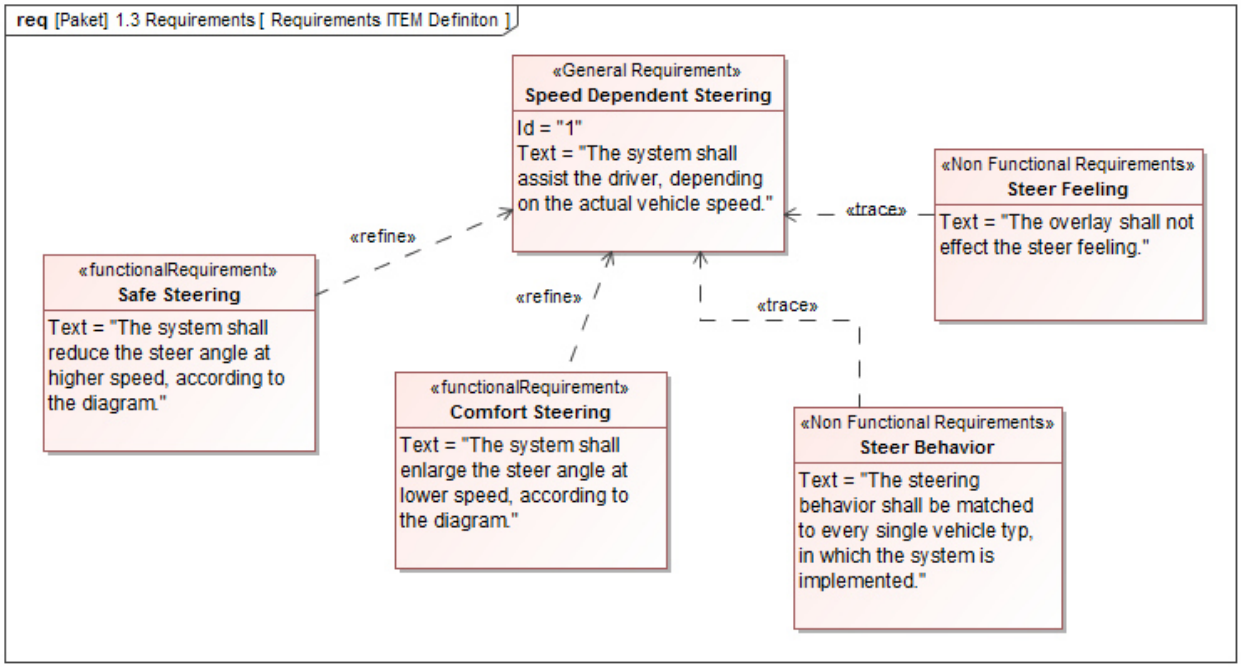


Abbildung 34: Main functional Requirements based on ITEM Definition

Alle Funktionen und Anforderungen, die das System beschreiben, werden im Systemmodell systematisch erfasst und dargestellt. Durch das Zuweisen von bestimmten Verbindungen, ist sofort ersichtlich, in welchem Zusammenhang die Anforderungen und Funktionen zu-einander stehen. Mit der Möglichkeit der Erzeugung von Stereotypen, können beliebig viele Anforderungsarten erstellt werden.

Die grundlegenden Funktionen des Systems, die sich aus der Hauptanforderung „geschwindigkeitsabhängiges Lenken“ ergeben, sind in einem Aktivitätsdiagramm, das in Abbildung 35 zu sehen ist als Aktivitäten dargestellt.

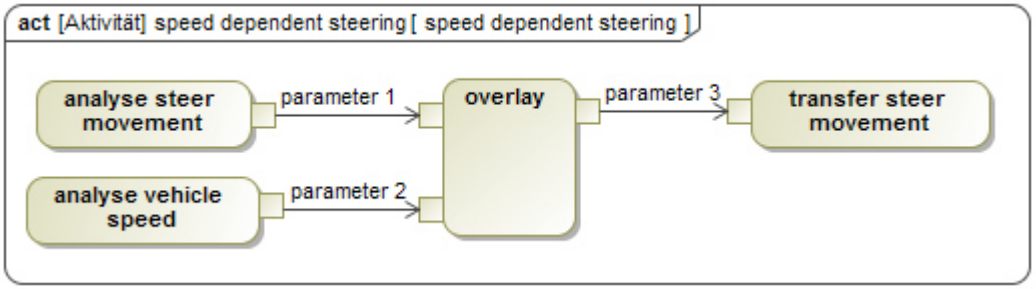


Abbildung 35: Simple System Functions

Um das System weiter zu definieren, ist eine Abgrenzung des Systems und des Systemkontextes notwendig. In Abbildung 36 ist das entsprechende Systemkontextdiagramm dargestellt.

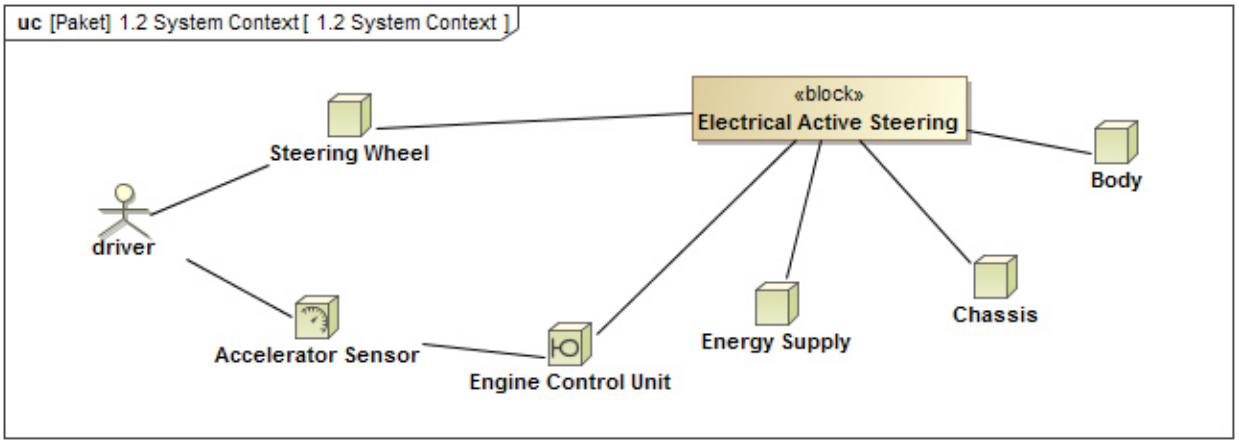


Abbildung 36: System Context

Es sind exemplarisch Nachbarsysteme, und die Umfeldelemente, mit deren Hilfe der Fahrer mit dem Lenkungssystem interagiert definiert.

Um die ITEM Definition abzuschließen müssen gemäß der ISO 26262 die Betriebsarten festgelegt werden, in denen das System betrieben wird. Diese Betriebsarten sind für die Durchführung der HARA essenziell und werden aus dem Anwendungsfalldiagramm aus Abbildung 37 abgeleitet. Demnach wird der Fahrer das System in den Betriebsarten betreiben, die sich aus den Geschwindigkeitsbereichen des Straßenverkehrs ergeben.

In Abbildung 37 sind die Anwendungsfälle, aus denen die Betriebsarten abgeleitet werden, separat dargestellt und violett eingefärbt.

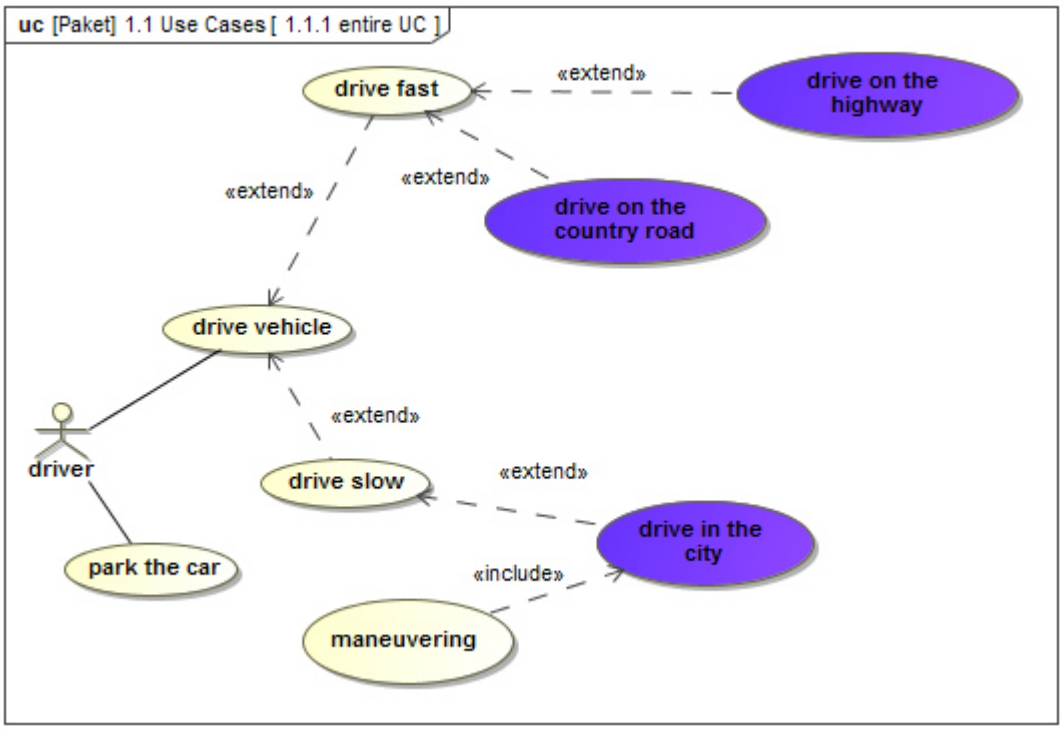


Abbildung 37: Derive Operating States from Use Cases

Aus den Anwendungsfällen werden Zustandsautomaten erstellt, welche die Arten der Fahrzeugnutzung im Betrieb detaillierter beschreiben.

Der systemspezifische Zustand ist die jeweilige Geschwindigkeit, in der das System betrieben wird. Wie in Abbildung 37 dargestellt ist, wird die Geschwindigkeit in drei Bereiche unterteilt. Jeder Bereich stellt im Zustandsautomaten einen Zustand dar.

Des Weiteren muss die Fahrzeugumgebung herangezogen werden, in der das Fahrzeug betrieben wird. Hierfür stellt die ISO 26262 Tabellen bereit, aus denen die jeweiligen Umgebungszustände entnommen werden können.

Abbildung 38 zeigt den Zustandsautomaten, der die geschwindigkeitsabhängigen Fahrzustände des Fahrzeuges beschreibt.

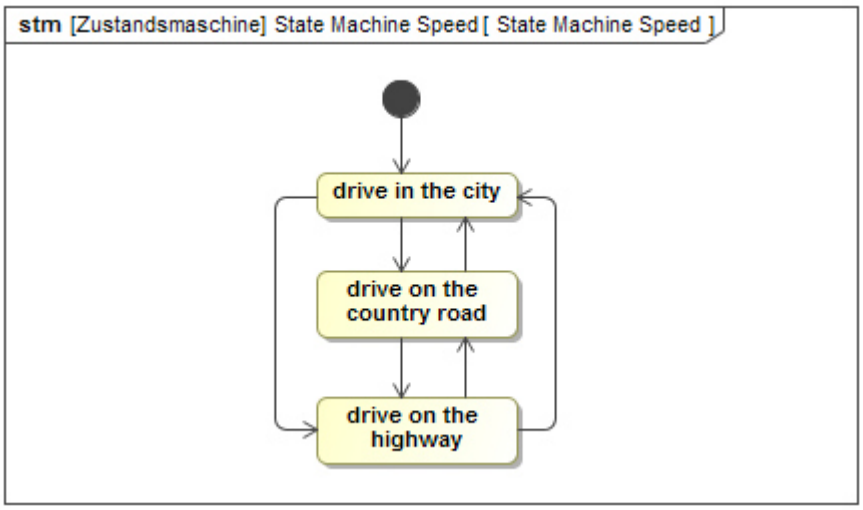


Abbildung 38: State Machine Speed Range

Abbildung 39 stellt die jeweiligen Umgebungszustände dar, in denen das Fahrzeug betrieben wird.

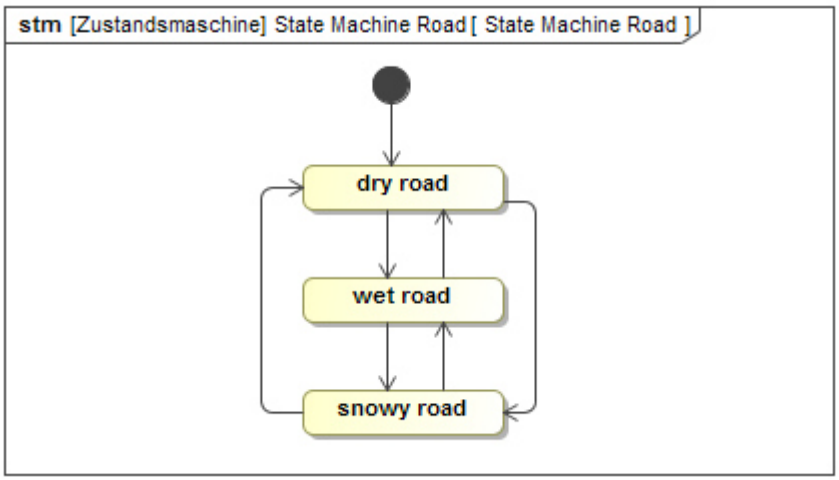


Abbildung 39: State Machine Road Condition

Die einzelnen Zustände der Zustandsautomaten werden im Folgenden kombiniert. Aus der Kombination der Zustände in den Zustandsautomaten, werden Betriebszustände abgeleitet.

In Abbildung 40 sind diese Kombinationen der Zustände in einer Matrix dargestellt.

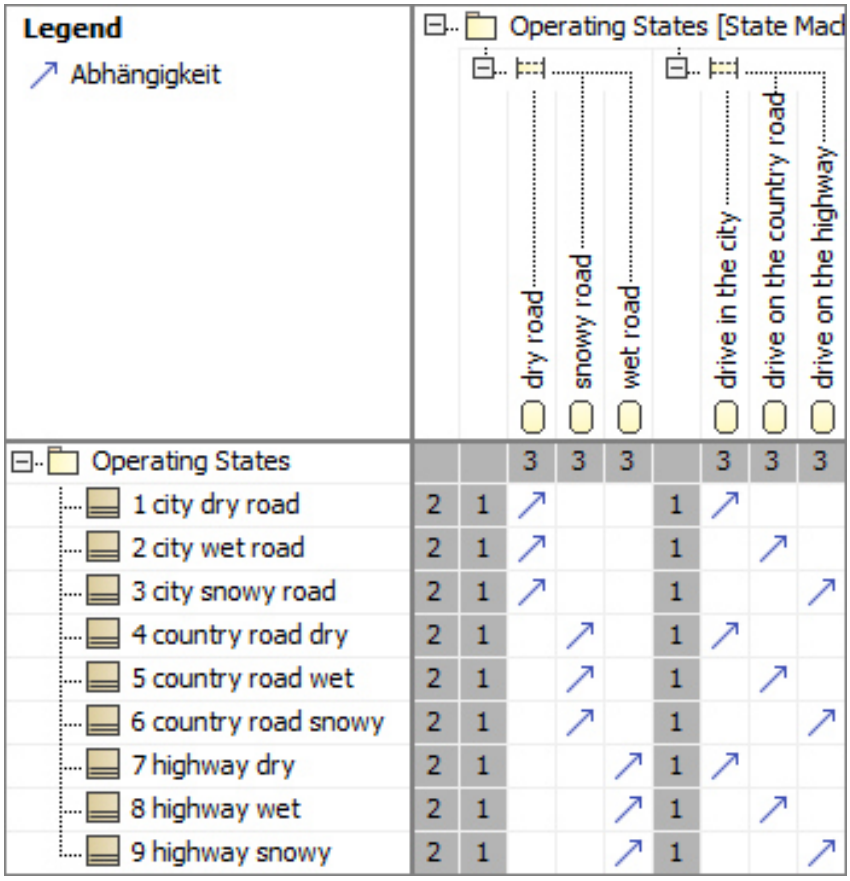


Abbildung 40: Operating States

Es ist zu prüfen, ob die Kombination aller Zustände sinnvoll ist. Kombinationen, die bezogen auf die praktische Nutzung des Fahrzeugs keinen Sinn ergeben bzw. in der Realität nicht vorkommen, sind auszuschließen.

Die Zustände werden miteinander kombiniert, daraus entstehen neun mögliche Betriebszustände.

Die ITEM Definition ist an diesem Punkt abgeschlossen. Alle systematisch erarbeiteten Erkenntnisse fließen in das Systemmodell ein und dienen als Grundlage für die HARA.

4.3 Hazard Analysis & Risk Assessment (HARA)

In der HARA werden die Funktionen, die in der ITEM Definition bestimmt wurden, auf mögliche Gefahren und Risiken untersucht, die während des Betriebes auftreten könnten. Es werden die Fehlfunktionen bestimmt und die Situationen, in denen diese auftreten könnten. Diese Szenarien werden anhand von definierten Bewertungskriterien bewertet und klassifiziert.

Im ersten Schritt werden aus den Hauptfunktionen des Systems, die in der ITEM Definition bestimmt wurden, die Fehlfunktionen abgeleitet.

Die Hauptfunktion des Systems ist, wie in Abbildung 31 aufgezeigt, das „geschwindigkeitsabhängige Lenken“. Aus dieser Hauptfunktion werden folgende Fehlfunktionen definiert:

1. **Steer Movement Inverting**
2. **Blocking**
3. **Oversteering**

Im zweiten Schritt werden die Gefahrensituationen ermittelt. Diese sind die möglichen Kombinationen aus den Betriebszuständen und den Fehlfunktionen, die während des Betriebes auftreten könnten. In Abbildung 41 auf der folgenden Seite, sind die Zusammenhänge in einer Abhängigkeitsmatrix dargestellt.

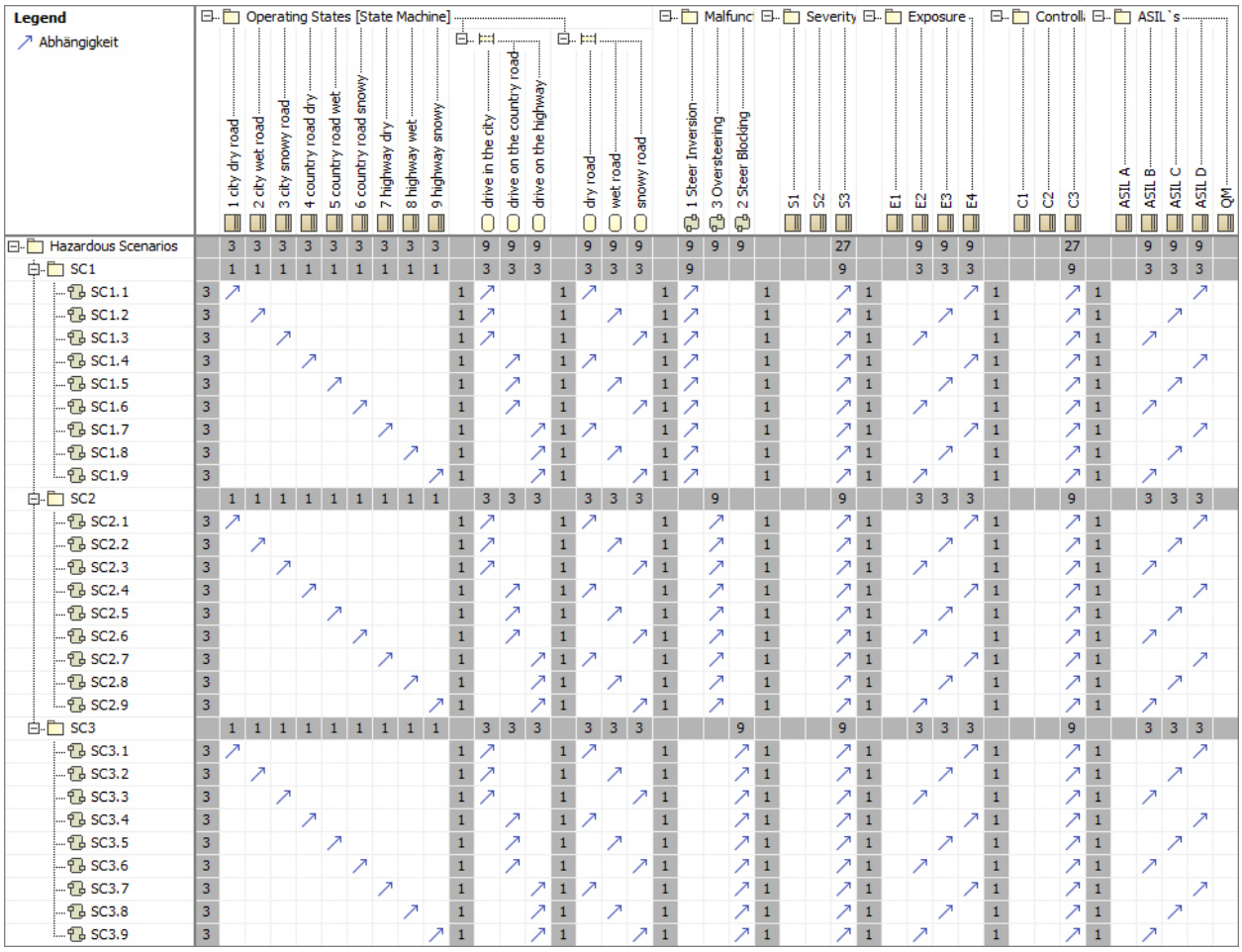


Abbildung 41: Hazard & Risk Matrix

Jede Zeile in der Matrix stellt eine Gefahrensituation dar. Diese sind das Resultat aus der Kombination der Betriebszustände aus Abbildung 40 und den definierten möglichen Fehlfunktionen. Die Anzahl der Szenarien ergibt sich aus der Kombination der neun Betriebszustände mit den drei Fehlfunktionen. Jedes Szenario ist systematisch erfasst und im Systemmodell mit einer ID hinterlegt.

Des Weiteren ist in der Matrix die Risikobewertung nach ISO 26262 integriert. Die Bewertungsparamter „Severity“, „Exposure“ und „Controllability“ sind nach den normativen Vorgaben bewertet und das jeweilige ASIL kann direkt nachvollzogen werden. Die Bewertung der einzelnen Szenarien ist an die in der ISO 26262 Teil 3 vordefinierten Tabellen angelehnt. Durch die jeweilige Summe der einzelnen Spalten, kann direkt abgelesen werden, wie oft jedes ASIL in der Analyse vorkommt und eine erste Aussage über die Sicherheitsrelevanz des Systems ist qualitativ und quantitativ möglich. Szenario SC1.1 ist z.B. eine Fahrt in der Stadt auf trockener Straße, bei der eine Umkehrung der Lenkbewegung auftritt. Die Schwere ist mit S3 bewertet, da das Fahrzeug unkontrolliert in den Gegenverkehr geraten kann und ein Frontallaufprall mit einem anderen Fahrzeug möglich ist. Die Auftretenswahrscheinlichkeit des Betriebszustandes „Fahrt in der Stadt auf trockener Straße“ ist sehr hoch und mit E4 bewertetet. Die Kontrollierbarkeit des Fahrzeuges während des Szenarios ist mit C3 bewertet, da davon ausgegangen wird, dass mehr als 90% der Autofahrer die Kontrolle verlieren würden.

Aus den möglichen Gefahrensituationen sind die Sicherheitsziele so zu formulieren, dass die Gefahrensituationen nicht eintreten dürfen und das oberste Ziel immer die Verhinderung der Gefahr ist.

In Abbildung 42 sind die Sicherheitsziele als übergeordnete Anforderungen in einem Anforderungsdiagramm dargestellt.

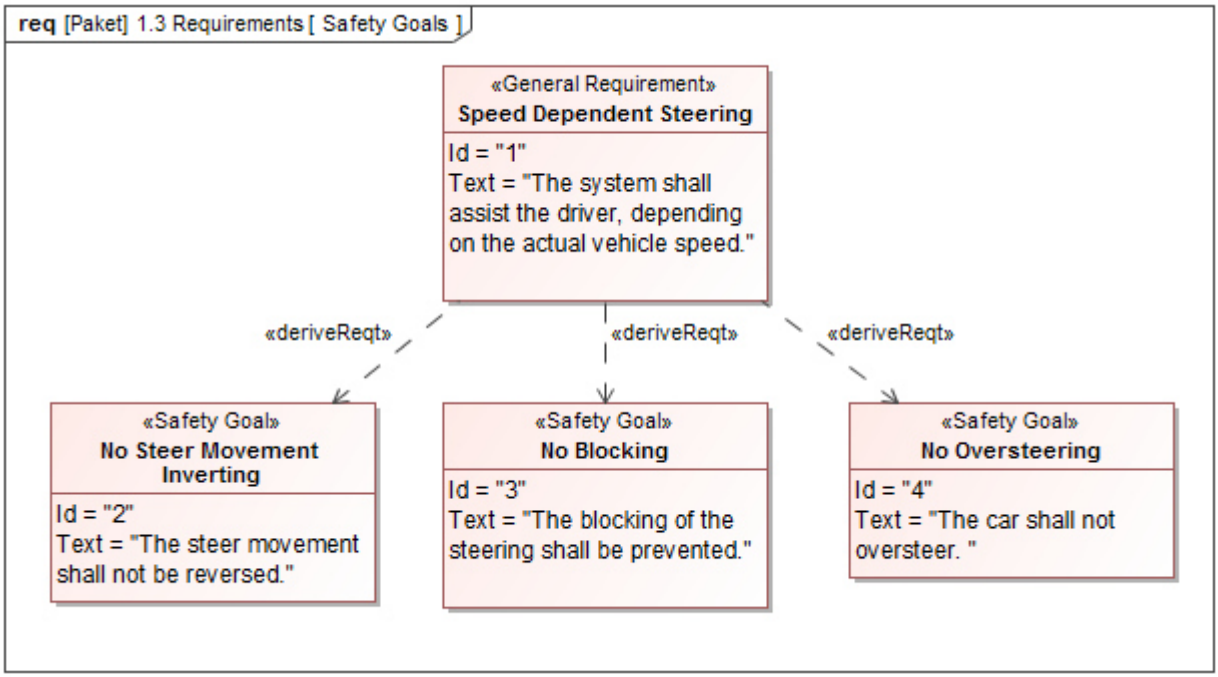


Abbildung 42: Safety Goals

Die Sicherheitsziele beschreiben somit die übergeordneten Fehlfunktionen, die nicht eintreten dürfen, damit das System als sicher gilt.

An diesem Punkt ist die Gefahren- und Risikoanalyse abgeschlossen. Die Erkenntnisse der ASIL Bewertung fließen in das funktionale Sicherheitskonzept ein und spezifizieren, wie die Sicherheitsziele erreicht werden sollen. Entsprechend der normativen Vorgaben folgt im nächsten Schritt das funktionale Sicherheitskonzept. Im nächsten Abschnitt wird aufgezeigt, wie dieses durch den modell-basierten Ansatz unterstützt werden kann.

4.4 Functional Safety Concept (FSC)

Im funktionalen Sicherheitskonzept (engl. functional safety concept) werden Anforderungen definiert, die die Sicherheitsziele aus der HARA auf funktionaler Ebene weiter spezifizieren sollen. Entsprechend der Methodik kann hierfür auch das Systemmodell mit der funktionalen Architektur und der entsprechenden Sicht verwendet werden. An dem Beispiel soll im Folgenden exemplarisch die Ermittlung des FSC aufgezeigt werden.

In Abbildung 43 ist das Vorgehen exemplarisch in einem Anforderungsdiagramm dargestellt. Für die Umsetzung wurden SysML Stereotypen für die Sicherheitsziele und die Sicherheitsanforderungen erstellt. Dadurch werden die spezifischen FuSi-Anforderungen entsprechend der normativen Vorgaben ersichtlich.

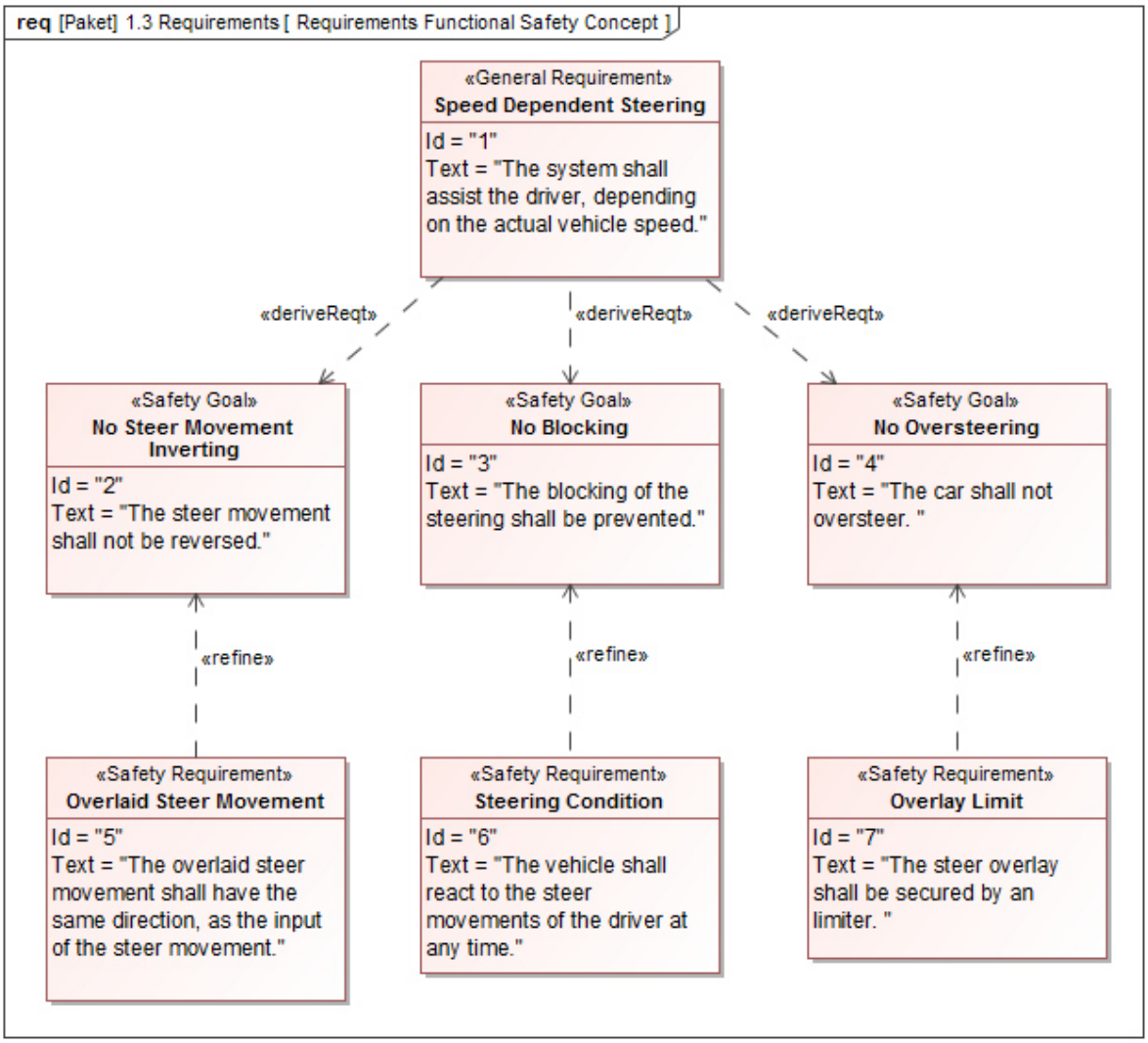


Abbildung 43: Requirements Functional Safety Concept

Die funktionalen Sicherheitsanforderungen sind so zu formulieren, dass die Sicherheitsziele (Safety Goals) auf funktionaler Ebene abgesichert werden.

Im funktionalen Sicherheitskonzept wird die vorläufige funktionale Architektur des Systems beschrieben und in Abbildung 44 in einem Aktivitätsdiagramm dargestellt.

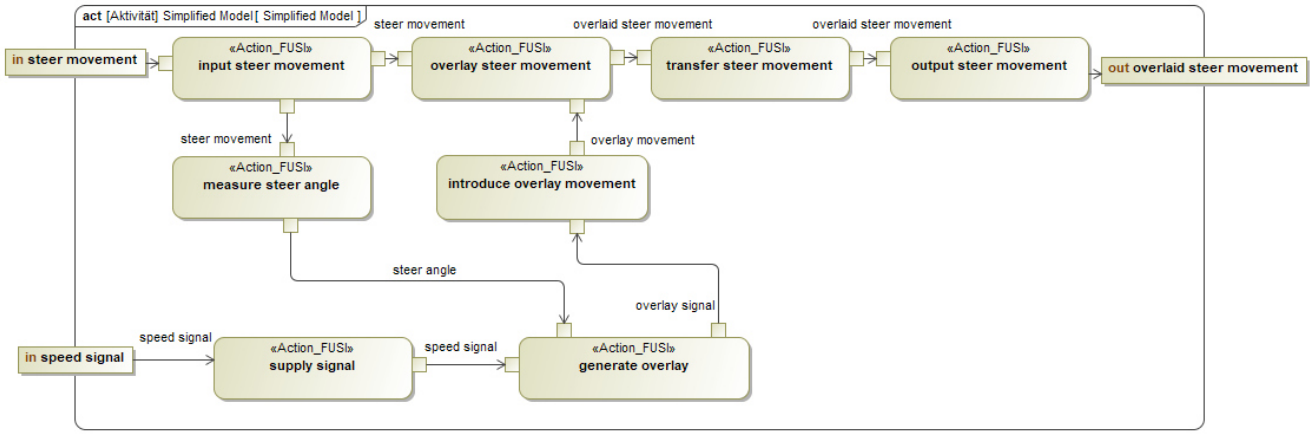


Abbildung 44: Functional Architecture

Die Funktionalität wird über die einzelnen Aktionen dargestellt. Die Lenkbewegung wird über die Aktivität „input steer movement“ in das System eingebracht. Das Geschwindigkeitssignal wird über die Aktion „supply signal“ zur Verfügung gestellt. Der Lenkwinkel sowie die Richtung des Lenkwinkels wird über die Aktion „measure steer angle“ ermittelt. Beide Signale fließen in die Aktion „generate overlay“, in der sie verarbeitet werden und die Überlagerung definiert wird. Das daraus resultierende Überlagerungssignal wird über die Aktion „introduce overlay movement“ in eine Bewegung umgewandelt. Die Aktion „overlay steer movement“ überlagert schließlich die Lenkbewegung, welche dann über die Aktionen „transfer steer movement“ und „output steer movement“ übertragen und ausgeben werden.

Des Weiteren findet im funktionalen Sicherheitskonzept die ASIL Dekomposition statt. Aus der HARA werden die jeweiligen ASIL der Safety Goals den jeweiligen funktionalen Sicherheitsanforderungen zugeordnet. Dies wird systematisch über das Systemmodell in einem Mapping erarbeitet und dargestellt.

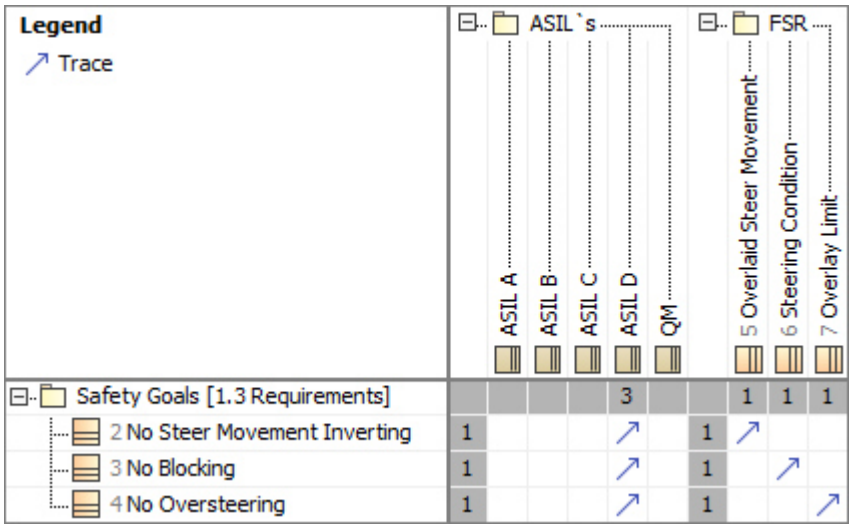


Abbildung 45: Mapping ASIL Decomposition 1

In Abbildung 45 ist der erste Schritt der Dekomposition dargestellt, in dem eine systematische Verbindung zwischen den Safety Goals, dem ASIL und den funktionalen Sicherheitsanforderungen im Systemmodell beschrieben wird. Da für jede Fehlfunktion in der HARA mindestens eine ASIL D Einstufung vorliegt, werden die Safety Goals alle mit ASIL D bewertet. Somit wird das „Worst Case Scenario“ in jedem Falle berücksichtigt. In einer weiteren Abhängigkeitsmatrix in Abbildung 46 werden die funktionalen Anforderungen den jeweiligen Aktionen, der vorläufigen funktionalen Architektur zugeordnet. Durch diese Zuweisung wird den Aktionen das jeweilige ASIL aus dem ersten Mapping über die funktionalen Anforderungen vererbt.

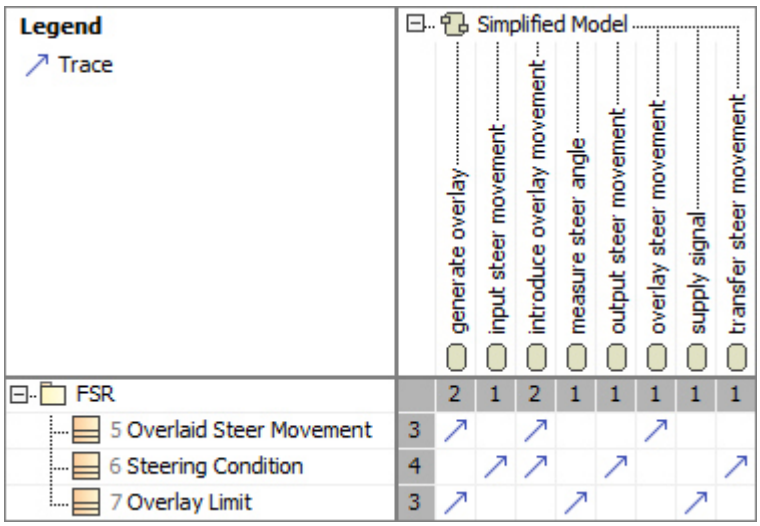


Abbildung 46: Mapping ASIL Decomposition 2

In Abbildung 47 ist die funktionale Struktur des Systems in einem Aktivitätsdiagramm dargestellt. Durch die Ergebnisse der ASIL Dekomposition und den SysML Stereotypen können nun die einzelnen ASIL an die Funktionen gekoppelt werden.

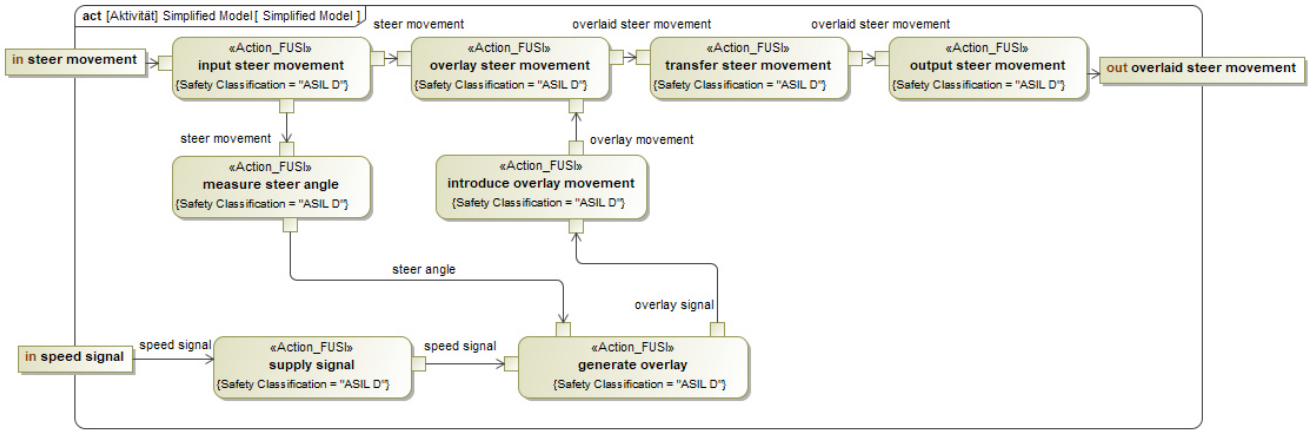


Abbildung 47: Functional Architecture, ASIL

Das FSC ist somit abgeschlossen. Die Erkenntnisse aus der ASIL Dekomposition, sowie die vorläufige funktionale Architektur fließen in das technische Sicherheitskonzept (TSC) mit ein und werden dort weiter spezifiziert.

4.5 Technical Safety Concept (TSC)

Im technischen Sicherheitskonzept wird die vorläufige funktionale Architektur des Systems aus dem funktionalen Sicherheitskonzept weiter spezifiziert und es werden induktive und deduktive Sicherheitsanalysen durchgeführt, um das System abzusichern. Die Erkenntnisse aus den Sicherheitsanalysen fließen in die Architektur des Systemmodells mit ein. Des Weiteren werden aus den Erkenntnissen technische Sicherheitsanforderungen definiert.

Für das Beispiel sollen die systematischen modell-basierten Sicherheitsanalysen anhand der FMEA und FTA aufgezeigt werden.

Zunächst wird die Systemstruktur aufgebaut. In Abbildung 48 ist die Systemstruktur in einem Block Definition Diagramm (bdd) dargestellt.

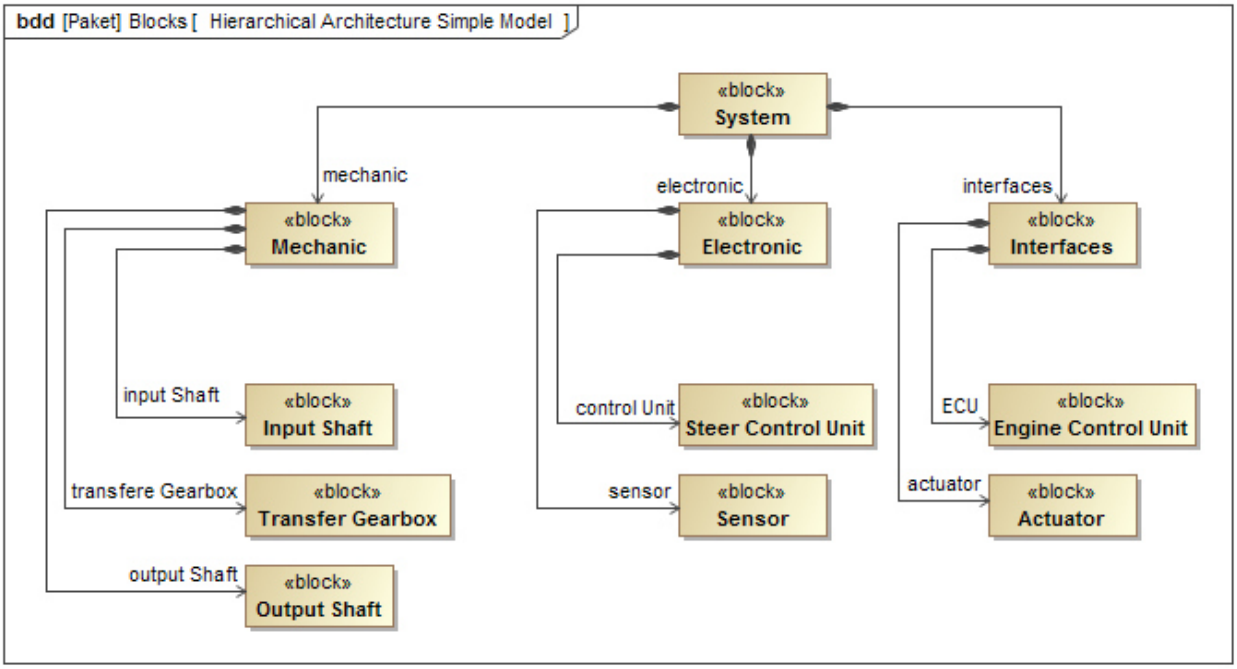


Abbildung 48: bdd, Hierarchical Architecture

Die Systemstruktur ist hierarchisch beschrieben. Der übergeordnete Block „System“ repräsentiert das Gesamtsystem. Dieses wird unterteilt in mechanische und elektronische Komponenten sowie Schnittstellen. Diese Darstellung dient dem besseren Verständnis des Systems und stellt den grundlegenden Aufbau dar. Die Strukturierung und Einteilung kann dem jeweiligen Fokus beliebig angepasst werden.

Die Aktionen, welche die Bauteilfunktionen repräsentieren, werden in einer Matrix in Abbildung 49 mit den jeweiligen Bauteilen gemappet.

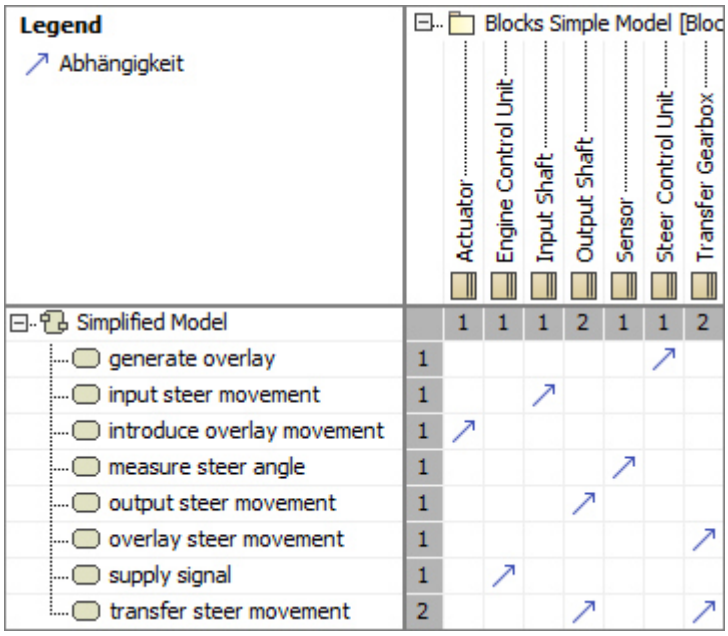


Abbildung 49: Mapping Actions & Blocks

Diese Gegenüberstellung dient der Zuordnung und der Kontrolle der Systemstruktur. Es zeigt, welche Bauteile welche Funktionen erfüllen. Dabei tritt eine n zu m Zuordnung auf. Die Darstellung der Beziehungen ist neben dem Ziel der systematischen Entwicklung später bei der Durchführung der FMEA zur Ursachenidentifikation relevant.

Da nur mechatronische Komponenten eine ASIL Einstufung bekommen können, kann die ASIL Dekomposition aus dem FSC durch das Mapping aus Abbildung 49 geschärft werden. Alle Funktionen die einer rein mechanischen Komponente zugeordnet werden, bekommen ein QM in Abhängigkeit des vererbten ASIL. In Abbildung 50 sind die Stereotypen der rein mechanischen Aktivitäten angepasst.

4.5.1 FMEA

Wie in Kapitel 3 methodisch aufgezeigt, wird die FMEA nun exemplarisch durchgeführt.

Der erste Schritt ist die strukturelle Systembeschreibung des Systems. Hierzu ist das bdd aus Abbildung 48 heranzuziehen.

Als nächstes sind die Systemfunktionen der einzelnen Komponenten zu identifizieren. Die Funktionen der einzelnen Komponenten sind dem Aktivitätsdiagramm aus Abbildung 47 dem funktionalen Sicherheitskonzept zu entnehmen.

Die Anzahl der Systemlevel ist in diesem Fall auf ein Level begrenzt. Das System wird somit durch eine Aktivität, welche in mehrere Aktionen unterteilt ist, dargestellt. Bei dieser Granularität der Systembeschreibung sind somit alle Funktionen sichtbar und es gibt keine Unterfunktionen.

Im nächsten Schritt werden die Funktionen auf mögliche Fehlfunktionen und deren Auswirkungen untersucht.

In Abbildung 50 ist die funktionale Architektur des Systems als Aktivitätsdiagramm dargestellt. Auf Basis des Mappings der Funktionen zu den Bauteilen (siehe Abbildung 49) werden die rein mechanischen Funktionen hier auch nur mittels QM (D) klassifiziert. Der Fokus liegt hierbei auf der Überlagerungsfunktion des Systems, die grün markiert ist.

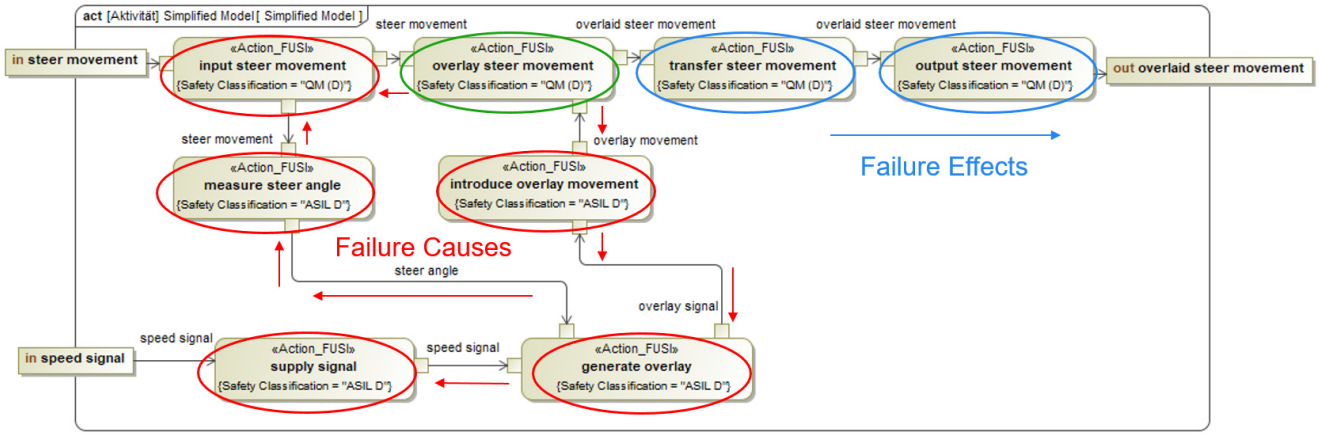


Abbildung 50: Reuse Functional Architecture for FMEA

Die Aktivitäten, auf die sich die Fehler auswirken können, sind blau markiert. Die Aktivitäten, die neben den Eingangssignalen als Fehlerursache in Frage kommen, sind rot gekennzeichnet. Es sind somit zwei Pfade systematisch zu analysieren, der Ursachenpfad und der Auswirkungspfad. Die Funktionen (Ursachen und Auswirkungen) können anschließend in das FMEA Formblatt übertragen werden.

In Abbildung 51 ist ein Ausschnitt des FMEA Formblatts dargestellt.

FMEA			FAILURE-MODE- AND EFFECT-A	
				Design-FMEA:
				Prozess-FMEA:
				System-FMEA:
Function	**Possible Failure Effects**	**Relevanz**	**Possible Failure / Failure Mode**	**Possible Failure Cause**
1				
1.1				
Function				
overlay steer movement	1. wrong transfered steer movement 2. wrong output steer movement **Hazardous Situations**		wrong overlaid steer movement	1. wrong input steer movement 2. wrong introduced overlay movement 3. wrong generated overlay 4. wrong supplied speed signal 5. wrong measured steer angle

Abbildung 51: Extract of FMEA Form

Die farbig markierten Aktionen aus dem Aktivitätsdiagramm aus Abbildung 50 sind in die jeweils farbig markierte Spalte der Tabelle einzutragen. Die mögliche Fehlfunktion der Überlagerungsfunktion ist in der weißen Spalte aufgeführt. Mit dieser Vorgehensweise ist eine systematische Untersuchung der Systemstruktur auf die mögliche Fehlfunktion eines Bauteils und deren Ursachen und Auswirkungen aufgezeigt.

Über die Abhängigkeitsmatrix aus Abbildung 49 können nun die Fehlfunktionen einzelnen Bauteilen zugeordnet werden. Somit ist sofort ersichtlich, welches Bauteil als mögliche Fehlerquelle weiter zu betrachten ist.

In Abbildung 52 ist dies visuell hervorgehoben.

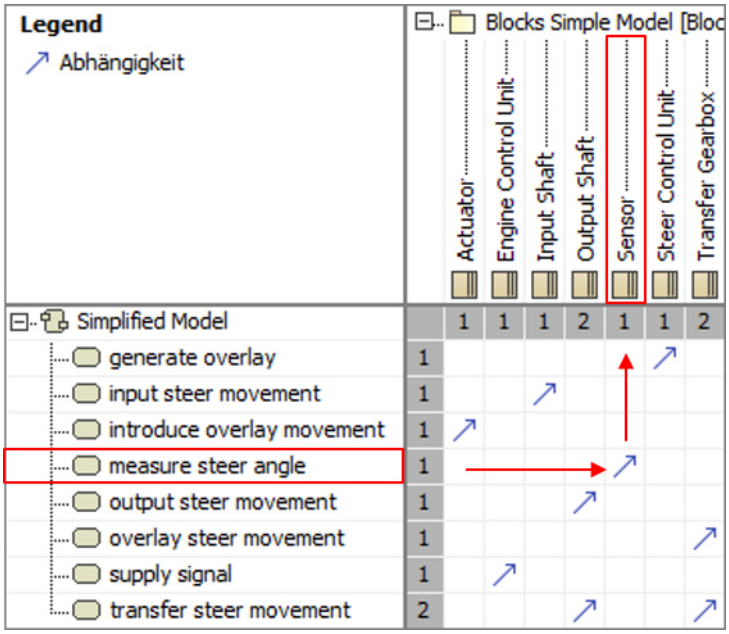


Abbildung 52: Reuse Mapping Actions & Blocks for Failure Analysis

Am Beispiel der möglichen Fehlerursache in der Aktion „measure steer angle“ ist das Bauteil „Sensor“ als mögliche Fehlerquelle identifiziert. Durch diese Art der Gegenüberstellung ist eine systematische Zuweisung auf Basis des Systemmodels sichergestellt.

Im nächsten Schritt sind Erkennungs- und Präventionsmaßnahmen zu ermitteln, um das Eintreten der beschriebenen Fehler zu verhindern, bzw. diese im Vorfeld zu erkennen.

Es ist zu erkennen, dass es keine Entdeckungsmaßnahmen oder Präventionsmaßnahmen gibt, die einen fehlerhaft überlagerten Lenkwinkel erkennen und eingreifen könnten.

Der Aufbau der Funktionen wird exemplarisch angepasst und die Steuerung wird zu einer Regelung ausgebaut.

In Abbildung 53 ist der modifizierte Aufbau der Funktionskette dargestellt. Die hinzugefügten Funktionen sind farbig markiert. Dabei sind die violett eingerahmten Aktionen Präventionsmaßnahmen. Die orange eingefärbte Aktion stellt eine Erkennungsmaßnahme dar.

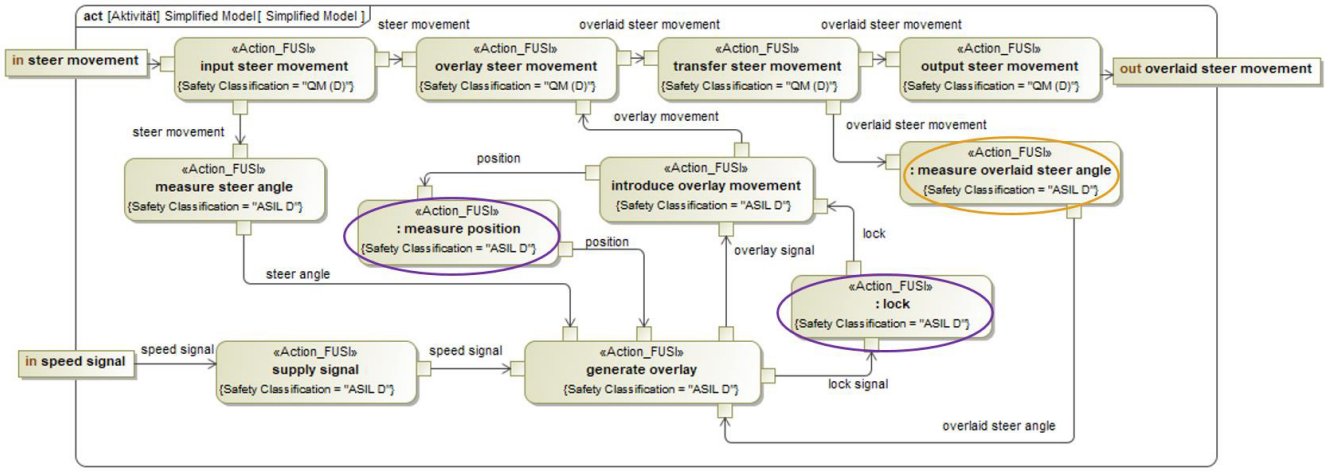


Abbildung 53: Extended Functional Architecture

Die Steuerung wird durch die Aktionen „measure overlaid steer angle“, „lock“ und „measure position“ erweitert. Somit wird eine fehlerhafte Überlagerung der Lenkbewegung erkannt und es gibt die Möglichkeit der Regelung. Des Weiteren wird die Aktion „introduce overlay movement“ überwacht. Sollten bei der Einbringung der Überlagerungsbewegung Fehler auftreten, kann die Funktion durch die Aktion „lock“ stillgelegt werden, und das Lenksystem arbeitet rein über die mechanischen Komponenten linear weiter. Die Aktion „lock“ stellt somit eine Integration eines „Safe States“ im System dar, welcher als „Fallback“ Maßnahme das System weiter absichert.

In Abbildung 54 werden die Entdeckungs- und Präventionsmaßnahmen in das FMEA Formblatt eingetragen.

FMEA

FAILURE-MODE- AND EFFECT-ANALYSES

Design-FMEA: ☐
Prozess-FMEA: ☐
System-FMEA: ☑

Processname
model/system/assembly
created by (name/dept.)

Function	Possible Failure Effects	Relevanz	Possible Failure / Failure Mode	Possible Failure Cause	Actual Status: Prevention-Method	Actual Status: Appearance	Actual Status: Detection-Method	Actual Status: Detection
1								
1.1								
Function								
overlay steer movement	1. wrong transfered steer movement		wrong overlaid steer movement	1. wrong input steer movement				
	2. wrong output steer movement **Hazardous Situations**			2. wrong introduced overlay movement	measure position , lock		measure overlaid steer angle	
				3. wrong generated overlay	measure position , lock			
				4. wrong supplied speed signal				
				5. wrong measured steer angle				

Abbildung 54: Extract of FMEA Form 2

Durch die Erweiterung der Funktionen in einen geschlossenen Regelkreis mit zusätzlichen Überwachungsmaßnahmen zur Absicherung der Hauptfunktion werden die Safety Goals auf der Systemebene erfüllt.

4.5.2 FTA

In diesem Abschnitt wird die FTA, wie sie in Kapitel 3 methodisch beschrieben ist, exemplarisch durchgeführt.

Der erste Schritt ist die Definition eines Top Events, welches durch ein Sicherheitsziel beschrieben wird. Als Top Event wird die Fehlfunktion „Oversteering“ ausgewählt.

Um den Fehlerbaum zu beschreiben, muss die logische Architektur des Systems herangezogen werden. Diese wird in Abbildung 55 durch ein ibd beschrieben. Die logischen Blöcke repräsentieren die Bauteile, welche über Flüsse miteinander verbunden sind.

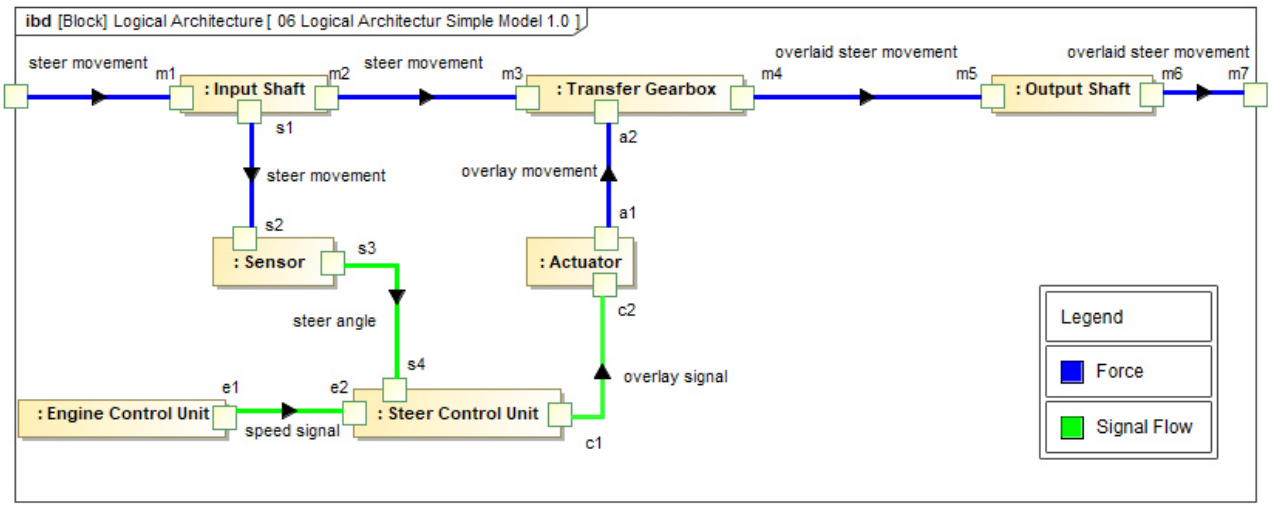


Abbildung 55: Create Logical Architecture

Im nächsten Schritt wird das System nach dem Top-down Prinzip untersucht. Jedes Bauteil stellt eine mögliche Fehlerquelle dar, somit wird an jedem Bauteil ein möglicher Fehler im Bauteil definiert. In Abbildung 56 ist die Vorgehensweise visuell hervorgehoben.

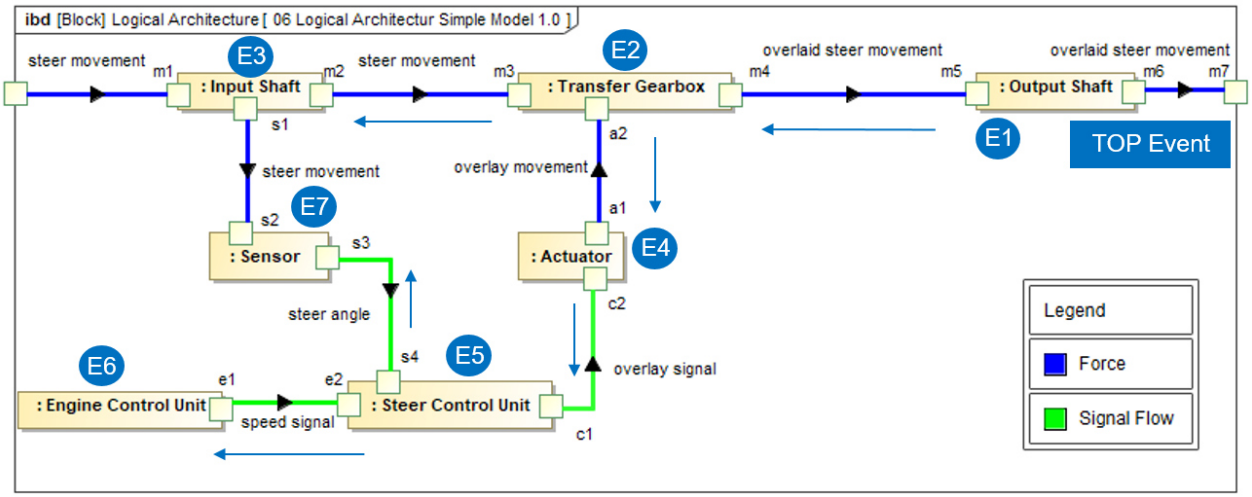


Abbildung 56: Create Logical Architecture Failure Path

Die möglichen auftretenden Fehler in den Bauteilen sind mit einem „E“ für Event bezeichnet und der Reihenfolge nach Top-down durchnummeriert. Die blauen Pfeile verdeutlichen den Fehlerpfad, nach dem das System systematisch analysiert wird. Das Top-Event steht am Systemausgang. Je nach Systemstruktur und Komplexität gibt es nicht nur einen Fehlerpfad, sondern es besteht auch immer die Möglichkeit der Teilung des Pfades oder der Zusammenführung.

Aus dem in Abbildung 56 systematisch erstellten Fehlerpfad wird im nächsten Schritt der Fehlerbaum erstellt. Dieser ist in Abbildung 57 abgebildet.

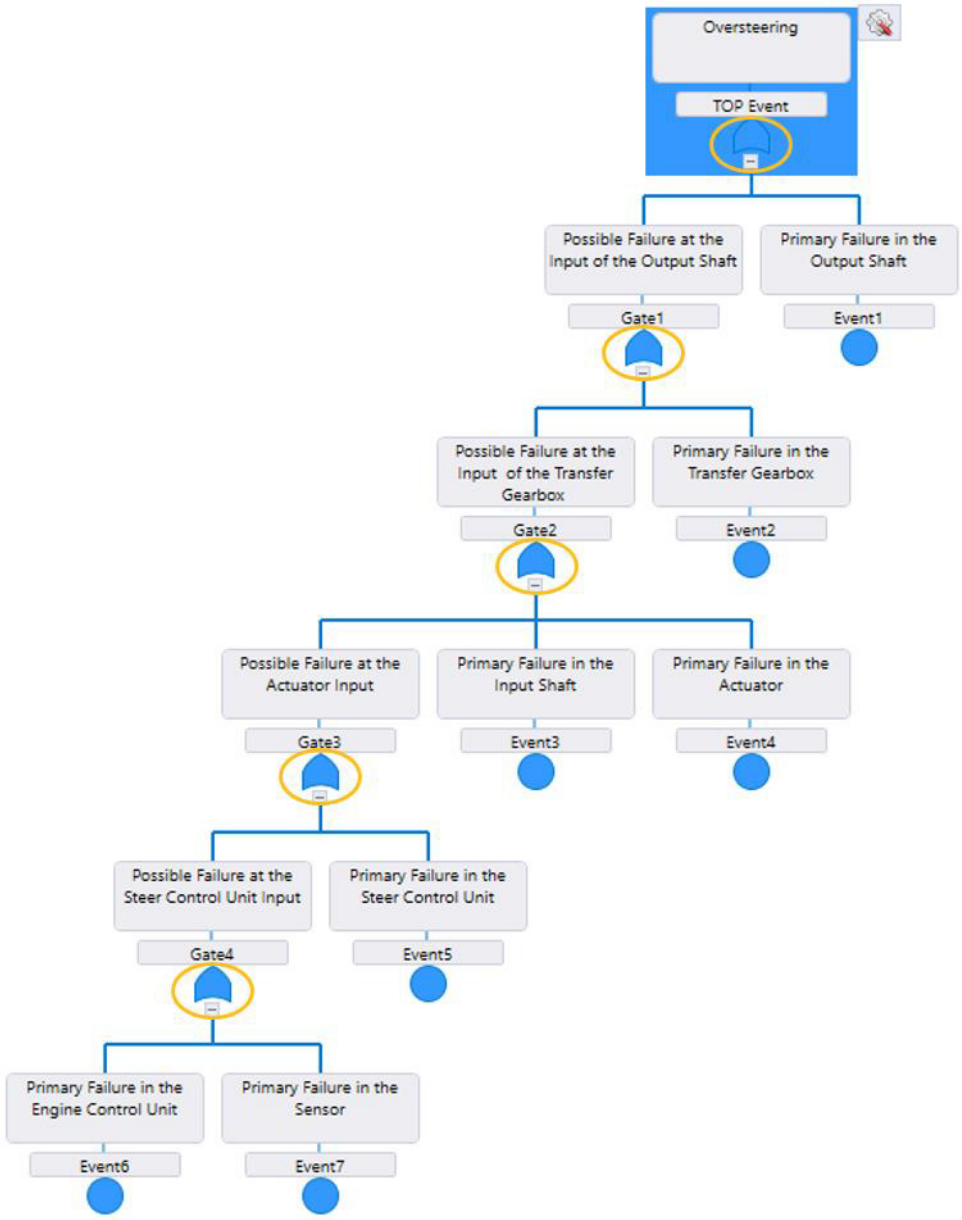


Abbildung 57: Create Fault Tree 1

Die mögliche Fehlerkette wird systematisch beschrieben. Das Eintreten eines möglichen Fehlers wird als Event beschrieben. Ist der Fehler in einem vorgelagerten Bauteil, wird der Fehlerpfad über ein Gate bis zum nächsten Bauteil weitergeleitet, in dem ein möglicher Fehler eintreten könnte. Dieser mögliche Fehler wird ebenfalls als Event gekennzeichnet. Dieses Vorgehen wird so lange systematisch weitergeführt, bis die komplette Bauteilstruktur im Fehlerbaum abgebildet ist.

Als nächstes ist das System anhand der Architektur und des Fehlerbaumes qualitativ zu bewerten. In dem exemplarischen Beispiel der logischen Architektur sind keine Absicherungsmaßnahmen durch Redundanzen integriert. Dies spiegelt sich im Fehlerbaum wider. Damit das Top Event eintritt, ist nur ein einziger Fehler in einem Bauteil der Kette notwendig. Dies ist an den orange markierten ODER-Verknüpfungen im Fehlerbaum zu erkennen. Das System ist somit nur durch die jeweilige Ausfallwahrscheinlichkeit der einzelnen Bauteile abgesichert.

Der Fokus sollte auf der Sensorik, der Regelung und der Aktorik liegen. Das Übersteuern des Fahrzeuges kann nur durch eine fehlerhafte Überlagerung eintreten. Hierzu sollten die Sensorik, die Verarbeitung der Signale und die Aktorik, welche für die Überlagerung der Lenkbewegung zuständig ist, abgesichert werden.

Die quantitative Bewertung der FTA ist in die jeweilige Software integriert. Bei jedem Event, welches zu einem Bauteil gehört, wird die Ausfallwahrscheinlichkeit des Bauteils angegeben.

In Abbildung 58 ist ein Screenshot der FTA Software [Top17] aufgeführt.

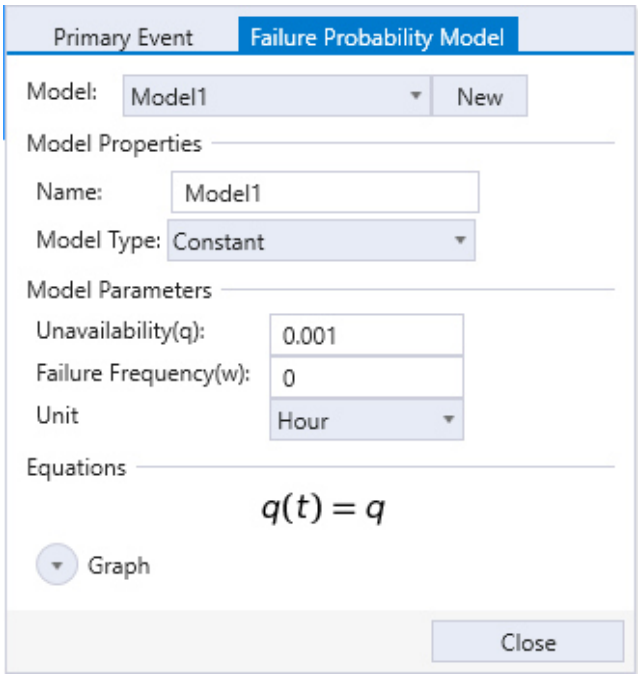


Abbildung 58: Screenshot Fehlerdeklaration

Für jeden Fehler werden der Fehlertyp und die Ausfallwahrscheinlichkeit angegeben. In Abhängigkeit der jeweiligen Daten der Bauteile und der Verknüpfung dieser untereinander, wird eine Gesamtausfallwahrscheinlichkeit des Systems bestimmt. Die Informationen zu Ausfallwahrscheinlichkeiten sind den jeweiligen Bauteilspezifikationen zu entnehmen.

Im nächsten Schritt wird der modifizierte Aufbau der Regelung untersucht. In Abbildung 59 ist der Aufbau als logisches Modell dargestellt.

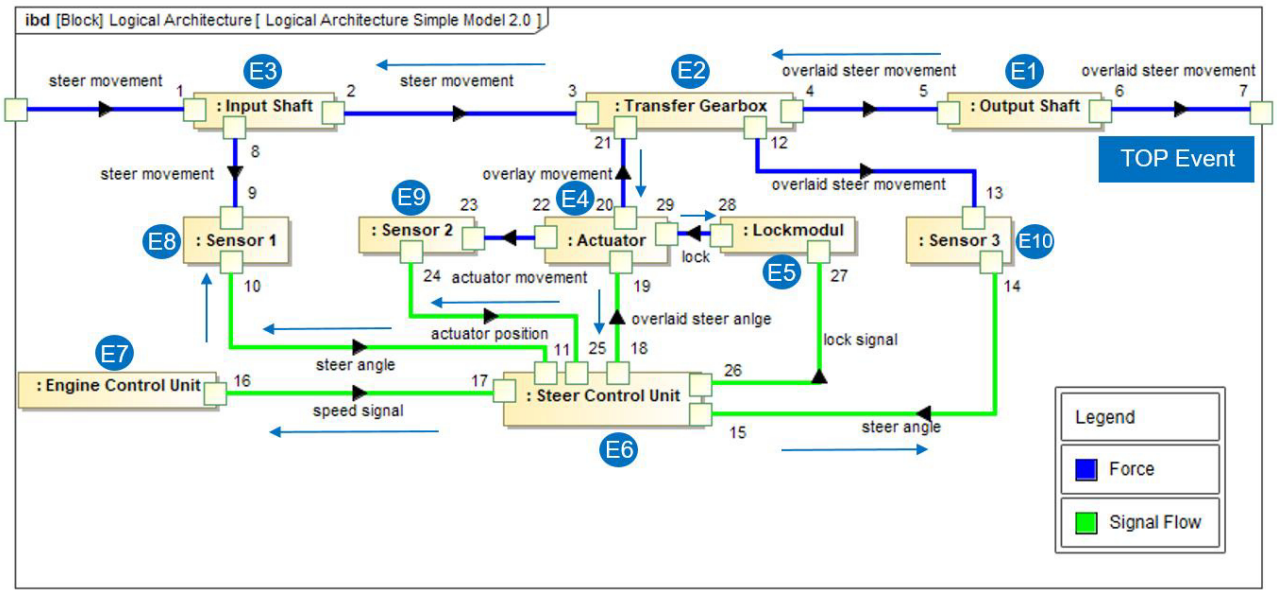


Abbildung 59: Extended Logical Architecture

Der modifizierte Aufbau ist mit einem neuen Fehlerpfad versehen und der Fehlerbaum wird im Folgenden in Abbildung 60 dargestellt.

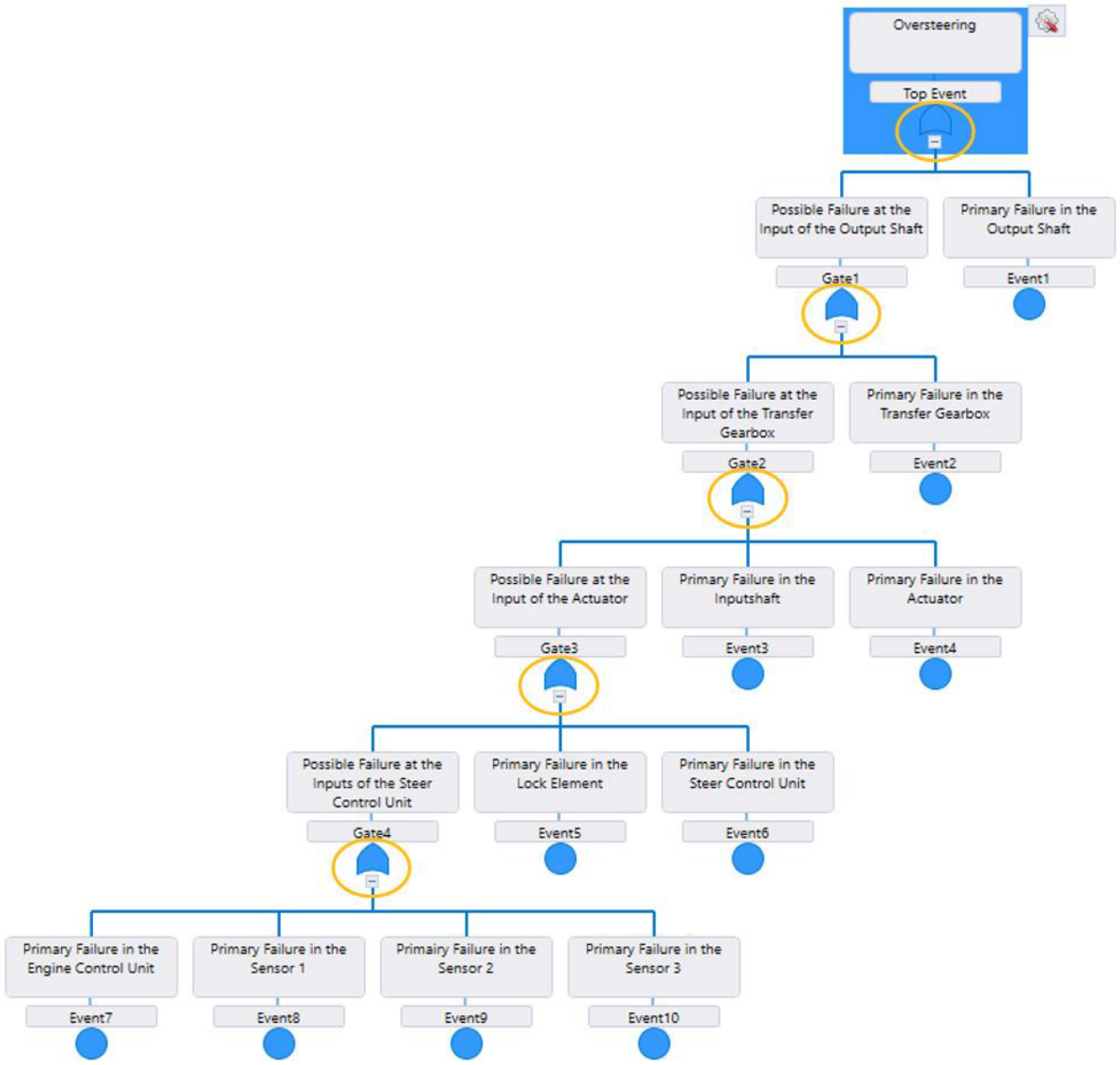


Abbildung 60: Adapted Fault Tree

Der modifizierte Aufbau ist dargestellt. Alle Sensorsignale laufen in die Steer Control Unit ein. Fällt ein Sensor aus, ist die Regelung nicht mehr gewährleistet und das System kann die geforderte Funktion nicht mehr zur Verfügung stellen. Um das System weiter abzusichern, können Redundanzen in die Systemstruktur integriert werden.

Für den Fall, dass die Systemarchitektur Redundanzen aufweisen soll, müssen weitere Bauteile zur Absicherung in die Systemstruktur integriert werden.

Redundanzen in der Systemarchitektur zeichnen sich durch UND-Verbindungen im Fehlerbaum ab. Damit das TOP-Event eintritt, müsste somit in allen Strängen einer UND-Verbindung ein Versagen auftreten. Somit ist eine Architektur mit redundanten Systemsträngen nicht nur durch die Ausfallwahrscheinlichkeit der jeweiligen Bauteile abgesichert. Im Fehlerbaum wirkt sich die doppelte Sensorauslegung als Redundanz aus, die sich als UND-Verknüpfungen, welche grün markiert sind, im Fehlerbaum darstellen. In Abbildung 61 ist der modifizierte Aufbau des Systems exemplarisch über den Fehlerbaum dargestellt.

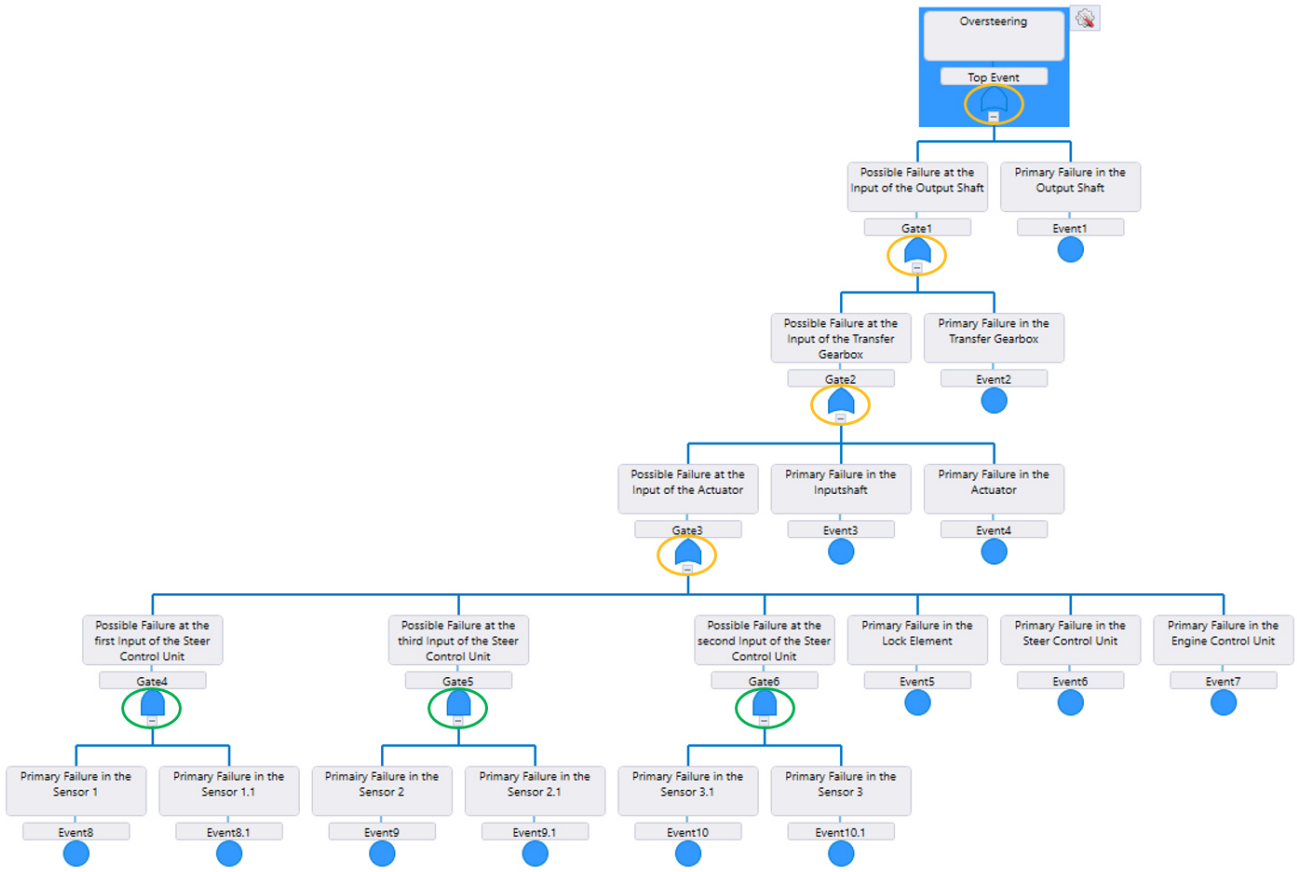


Abbildung 61: Adapted Fault Tree 2

Die Redundanzen zeichnen sich am Eingang der „Steer Control Unit“ ab. Die Ausfallwahrscheinlichkeit des Systems muss nach einer Änderung der Architektur ebenfalls neu berechnet werden. Anhand der Neuberechnung wird entschieden, ob die Bauteilausfallwahrscheinlichkeit kombiniert mit der Redundanz der Architektur ausreicht oder weitere Maßnahmen eingeleitet werden müssen.

Durch die Änderungen in der Systemarchitektur sind die technischen Sicherheitsanforderungen zu beschreiben.

In Abbildung 62 werden die Anforderungen mit technischen Sicherheitsanforderungen exemplarisch ergänzt.

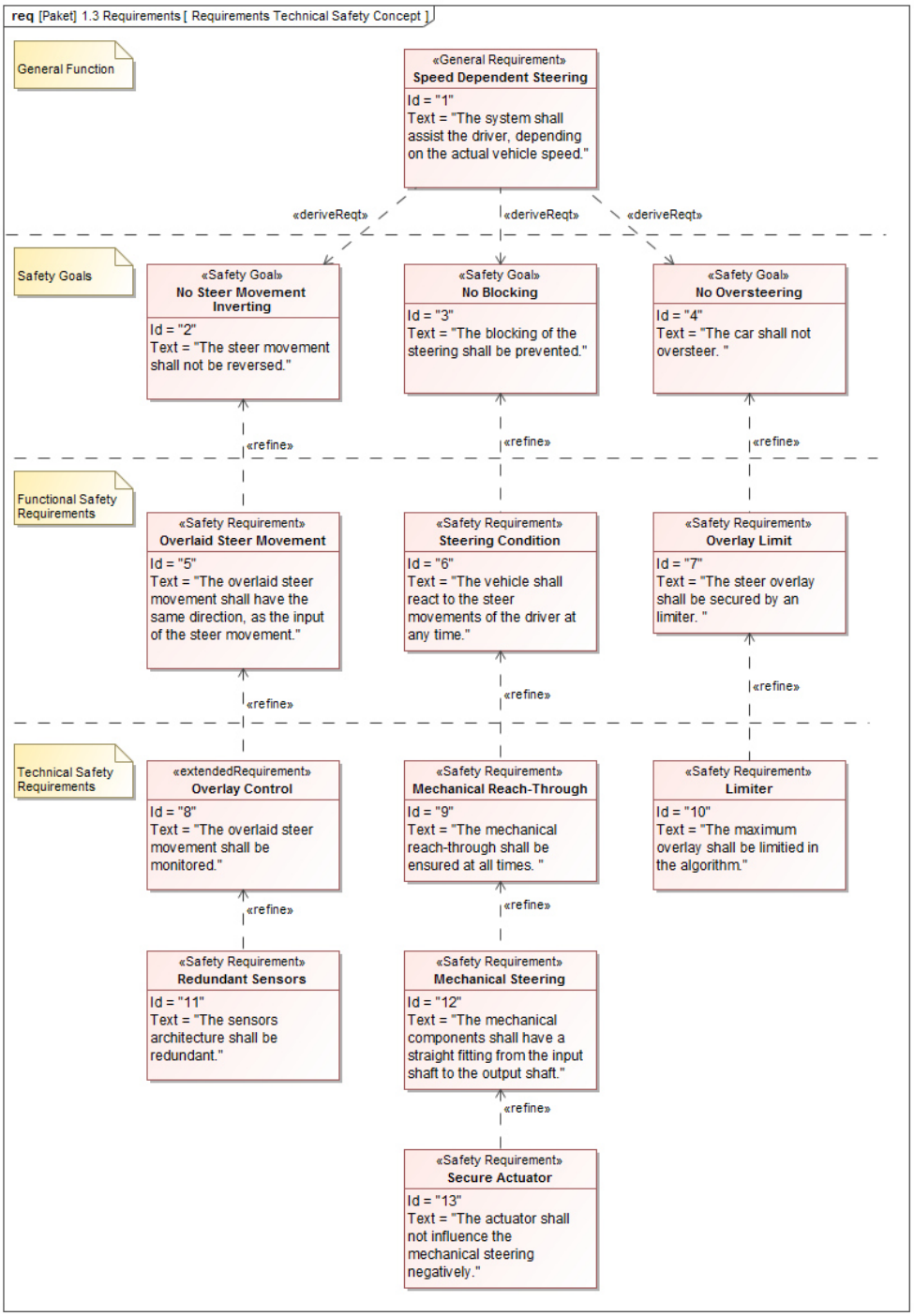


Abbildung 62: Requirements Technical Safety Concept

Die Hierarchie der Anforderungen ist in Abbildung 62 mit aufgezeigt. Je tiefer der Prozessschritt, desto feingranularer werden die einzelnen Anforderungen beschrieben. Im nächsten Schritt werden die technischen Anforderungen unterteilt und auf Software- bzw. Hardware Ebene beschrieben.

4.5.3 Hardware & Software Safety Requirements

Durch Hardware- und Softwaresicherheitsanforderungen (HSR und SSR) wird das System auf Bauteilebene weiter spezifiziert. Diese Anforderungen fließen in die Architektur ein und beschreiben diese auf einem hohen Detaillevel. Das Systemmodell wächst somit kontinuierlich weiter. Im Fokus ist dabei die Sensorik mit der Regeleinheit. Um das System weiterhin abzusichern und die funktionale Sicherheit zu gewährleisten, müssen die Komponenten detaillierter beschrieben werden. In Abbildung 63 auf der nächsten Seite ist das hierarchische Anforderungsdiagramm noch einmal abgebildet. Die Hardware und Softwaresicherheitsanforderungen wurden exemplarisch ergänzt. Die Architektur des Systems ist an die neuen Hard- und Softwaresicherheitsanforderungen anzupassen.

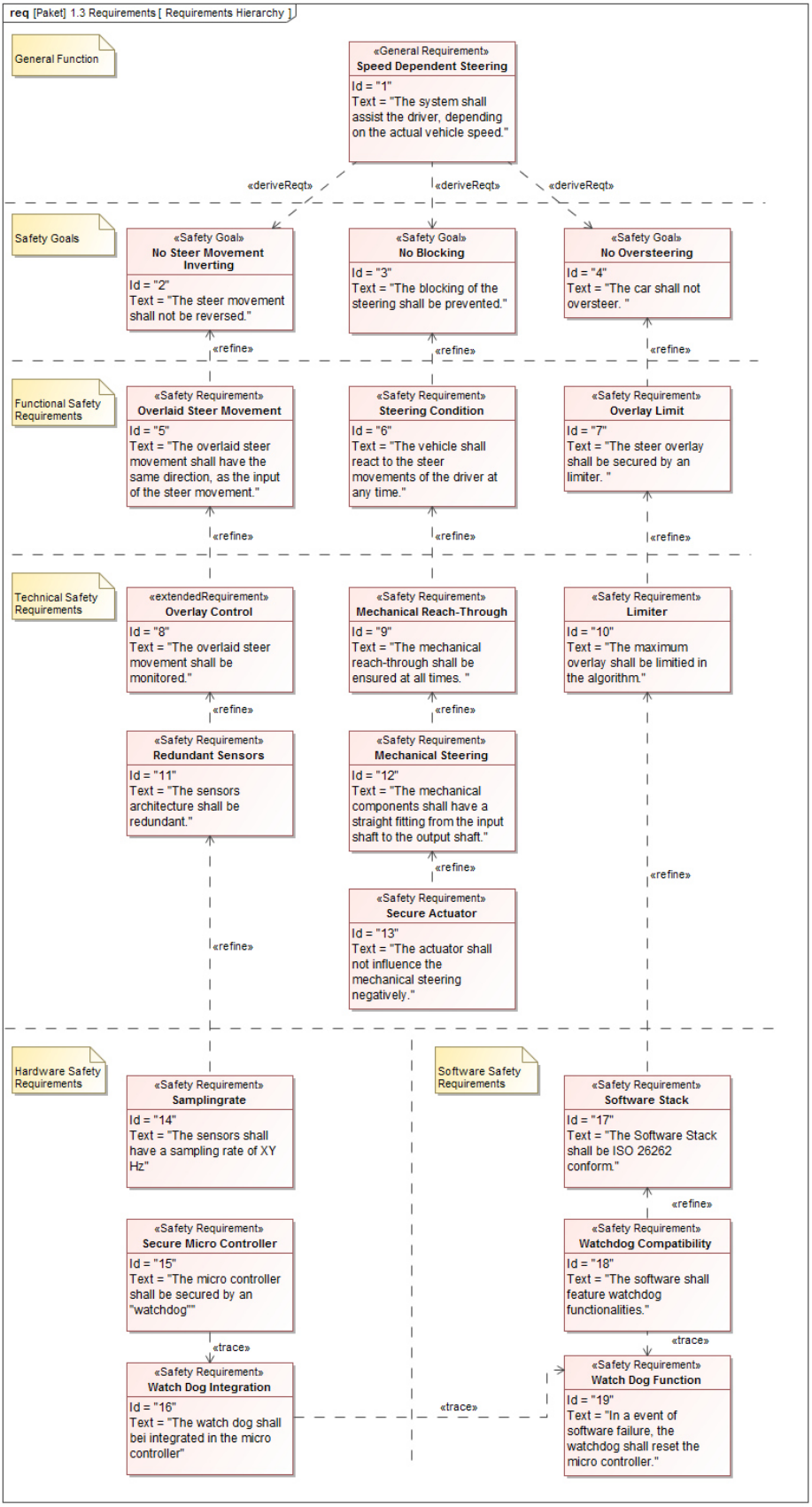


Abbildung 63: Requirements Hierarchy

An dieser Stelle wird die Methodik nicht weiter validiert. Die Software und Hardware Ebene werden nicht weiter behandelt.

5 Ergebnisdiskussion

In diesem Kapitel werden die Erkenntnisse der entwickelten Methodik und die Ergebnisse aus der Umsetzung bzw. der Validierung diskutiert und bewertet.

Resümee

Die Methodik aus Kapitel 3 stellt die modell-basierte Aufschlüsselung der normativen Vorgaben der ISO 26262 verknüpft mit der systematischen Systemmodellierung im Systemmodell als durchgängigen Entwicklungsprozess dar. Die Erarbeitung dieser Aktivitäten wird über die Schnittstellen mit dem Systemmodell durchgeführt. Der Mehrwert besteht in der systematisch modellierten Darstellung der Norm als Aktivitäten. An dieser Stelle ist die Begrifflichkeit des *Modellierens* sehr wichtig, da die Abgrenzung zur dokumenten-zentrierten Erarbeitung der normativen Vorgaben den wesentlichen Mehrwert der Methode ausmacht. Die normativen Vorgaben werden somit greifbar und auch sehr schnell verständlich, ohne die Norm als Textdokument vorliegen zu haben. Jeder Arbeitsschritt innerhalb der Norm wird als Aktivität dargestellt. Damit wird das Abarbeiten der einzelnen möglichen Arbeitspakete verständlich und überschaubar. Außerdem können die Verantwortlichkeiten leichter zugeordnet werden. Die Nachvollziehbarkeit innerhalb des Systemmodells stellt einen weiteren erheblichen Mehrwert dar. Überarbeitete Entwicklungsstände können aufgrund der atomaren Beschreibung mit semantischer Bedeutung und den Verbindungen zwischen den Beschreibungselementen schneller und eindeutiger erfasst werden als bei der text-basierten Entwicklung.

Die Aufgliederung der Hauptaktivitäten: ITEM Definition, HARA, FSC und TSC, letztere wird weiter untergliedert in HSR und SSR, machen die normativen Vorgaben und Umfänge verständlich und greifbar. Die einzelnen Arbeitsumfänge zur Erarbeitung der Inhalte sind in der jeweiligen Aktivität iterativ feingranular darstellbar.

Sowohl bei der Methodik als auch bei der Validierung werden bewusst nicht alle Aktivitäten der ISO 26262 modelliert und validiert. Der Fokus in dieser Masterarbeit liegt auf der Darstellung der Hauptaktivitäten, der Ableitung der Sicherheitsanalysen und der systematischen Darstellung dieser durch das Systemmodell.

ITEM Definition

Die ITEM Definition ist mit der aufgezeigten Methodik systematisch, durchgängig und komplett über das Systemmodell abbildbar. Die Nachvollziehbarkeit und Erweiterbarkeit ist über das Systemmodell gegeben. Die einzelnen Diagramme zur Erarbeitung der Inhalte sind leicht zu erfassen und stellen einen erheblichen Mehrwert zur rein dokumenten-basierten Entwicklung dar. Inhalte und Erkenntnisse können schnell aufgegriffen und nachvollzogen werden. Die Erarbeitung der Anforderungen ist mit dem Anforderungsdiagramm und der Stereotypenbildung von Anforderungstypen spezifisch darstellbar. Die Verknüpfung von Anforderungen mit Anwendungsfällen, deren Zusammenhänge in text-basierten Dokumenten oft nicht nachvollziehbar sind, verdeutlicht deren Zusammenhang.

HARA

Die HARA ist über die modell-basierte Darstellung systematisch und durchgängig abbildbar und die Inhalte bauen nahtlos auf der ITEM Definition auf. Durch die systematische Weiterverwendung der ITEM Definition ergibt sich eine nahtlose Beschreibung. Die HARA ist praktisch über eine Matrix/ Tabelle durchgängig abbildbar. Die Integration der ASIL Bewertung in das Systemmodell ist ein essenzieller Schritt. Durch diese Art der Darstellung ist auf einen Blick die Sicherheitsrelevanz des Systems in den jeweiligen Gefahrensituationen ersichtlich und es kann eine quantitative Aussage über das System in Hinblick auf die funktionale Sicherheit gemacht werden. Des Weiteren können die einzelnen Elemente und Artefakte im Systemmodell mit den jeweiligen ASIL Einstufungen verknüpft werden.

FSC

Das FSC wird ebenfalls durchgängig über das Systemmodell abgebildet. Die ASIL Dekomposition ist komplett, durchgängig und kompakt durchführ- und abbildbar. Der Mehrwert besteht auch hier in der modell-basierten Erarbeitung und Darstellung über eine Matrix. Die Nachvollziehbarkeit bietet ähnliche Mehrwerte, wie schon bei der ASIL Bewertung in der HARA. Über die Möglichkeit der Stereotypenbildung der SysML können die einzelnen ASIL den jeweiligen Funktionen zugeordnet werden. In der funktionalen Architektur ist somit auf einen Blick ersichtlich, wo die sicherheitsrelevanten Funktionen im funktionalen Sicherheitskonzept integriert sind.

TSC

Die Erarbeitung der FMEA und der FTA im TSC, sind mit dem Systemmodell unter Anwendung der beschriebenen Methodik möglich. Jedoch sind einzelne Arbeitsumfänge, der Erstellung des FMEA Formblattes und der Erstellung eines FTA-Fehlerbaumes nicht innerhalb des Systemmodells umsetzbar. Hierzu müssen externe Schnittstellen genutzt werden. Die Daten der FMEA welche in das FMEA Formblatt eingetragen werden, müssen händisch angepasst bzw. übertragen werden. Die Erstellung des FTA-Fehlerbaumes geschieht zwar systematisch anhand des Systemmodells, jedoch wird der Fehlerbaum mit einer externen Software erstellt. Die Daten des Fehlerbaumes bzw. des FMEA Formblattes sind somit nicht im eigentlichen Systemmodell integriert.

Bei der FTA wäre es ebenfalls sinnvoll, die einzelnen Komponenten der logischen Architektur mit den Aktivitäten zu mappen. Dadurch wären die Bauteilfunktionen in einem Schritt auch in der FTA mit der logischen Architektur verbunden und die Aktivitäten könnten in die Fehleranalyse des FTA-Fehlerbaumes mit einbezogen werden. Dies würde die Methodik weiter schärfen und die Durchgängigkeit verbessern.

Bei der FMEA würden durch eine modell-basierte Erweiterung mit einem „Relation Map Diagram“ weitere Mehrwerte entstehen. Durch diese Erweiterung könnte ein Funktions- bzw. Fehlernetz innerhalb des Systemmodells dargestellt werden.

Die Vor- und Nachteile zur systematischen Systementwicklung durch die Methodik sind in der folgenden Tabelle zusammengefasst.

Prozessschritt	Vorteile	Nachteile
ITEM Definition	• Komplett über das Systemmodell abbildbar • Durchgängig, systematisch und schnell nachvollziehbar • Vernetzung der Informationen durch Verknüpfungen von Anforderungen und Anwendungsfällen im Systemmodell erhöhen das Verständnis	• Übersicht lässt nach mit steigendem Systemumfang und Tiefe da die Darstellungen und Diagramme komplexer werden.
HARA	• Integration der ASIL Bewertung im Systemmodell. • ASIL relevante Artefakte sind im Systemmodell verknüpft • Inhalte der HARA über die Darstellung als Tabelle schnell greifbar und bewertbar.	• Mit Zunahme von potenziellen Gefahrensituationen und Betriebszuständen wächst der Umfang der Darstellungen stark an.
FSC	• ASIL Dekomposition über zwei Tabellen durchgängig und vollständig abbildbar. • ASIL modell-basiert durch Stereotypen auf Funktionen und Bauteile übertragbar • ASIL Dekomposition im Systemmodell vernetzt integriert	• Mehraufwand durch Modellierung
TSC	• Das Systemmodell leistet einen großen Beitrag zur Erarbeitung der nötigten Strukturen um die Analysen durchzuführen. • FMEA über Aktivitätsdiagramme und Mappings zwischen Aktionen und Bauteilen visuell gut abbildbar. Durchführung anhand des Systemmodells gut nachvollziehbar. • FTA über die internen-Blockdiagramme sehr gut abbildbar, Aufbau des Fehlerbaums ist systematisch und strukturiert.	• Beide Sicherheitsanalysen sind nicht komplett ins Systemmodell integriert. • Es werden externe Schnittstellen und Software benötigt. • Mehraufwände durch mehrfache Modellierung

Tabelle 9: Ergebnisdiskussion

Die mögliche Umsetzung der Methodik in der Praxis

Um die entwickelte Methodik in der Praxis umzusetzen, sollte diese anhand von weiteren Beispielen weiter validiert werden. Die Anwendung der Methodik auf der Hard- und Software Ebene im TSC bringt weitere Herausforderungen mit sich, da der Detaillierungsgrad auf diesen Ebenen stark ansteigt.

Des Weiteren setzt die Methodik das intensive Erarbeiten der konkreten Inhalte und Vorgaben heraus, die in der Entwicklung durchlaufen und abgebildet werden sollen. Das erarbeitete Detailwissen muss umgesetzt und als Prozess abbildbar modelliert werden. Alle Fehler, die bei der Umsetzung der Norm in Aktivitäten und Aktionen entstehen, werden später auch im Entwicklungsprozess auftreten. Somit ist die Korrektheit der Umsetzung von großer Relevanz.

Das Verstehen und Umsetzten dieser modellierten Prozessschritte setzt wiederum die Kenntnisse des modell-basierten Systems Engineering voraus, ohne welche weder die Erarbeitung noch die Abarbeitung der modellierten Prozessschritte erfolgen kann. Alle Mitarbeiter, die zur Erarbeitung der Methode und in die spätere Entwicklung unter Anwendung der Methode mit einbezogen werden, müssen somit Kenntnisse im modell-basierten Entwickeln haben und mit dem jeweiligen Modellierungswerkzeug vertraut sein.

Hinzu kommen die benötigten Mittel wie IT-Infrastruktur, spezifische Software, Lizenzen und Hardware, um die modell-basierte Entwicklung einzuführen und umzusetzen.

Das Anwenden von MBSE zur modell-basierten Entwicklung und Nutzung eines zentralen Datenmodels kann zwar iterativ, durchgängig systematisch und nachvollziehbar sein, jedoch steht und fällt das Systemmodell mit der Qualität der Strukturierung und Modellierung. Der Aufbau und die Struktur bilden das Fundament des späteren Modells und der eigentlichen Entwicklung. Eine schlechte Modellstruktur wirkt sich auf alle weiteren Elemente im Systemmodell, die darauf aufbauen, aus. Des Weiteren ist auf die Sinnhaftigkeit des Modells zu achten. Es sollten immer nur die Sichten und Diagramme erstellt und abgebildet werden, die auch tatsächlich benötigt werden. Dies hält das Modell kompakt, sorgt für Überschaubarkeit und hilft bei der Sicht auf den Gesamtumfang. Das Systemmodell ist immer sauber zu halten, unnötige Diagramme oder hinfällige Daten sind zu löschen.

Die Stereotypenbildung kann sehr hilfreich sein, wenn es um die spezifische Darstellung von Artefakten im Systemmodell geht. Stereotypen sollten jedoch nur erzeugt werden, wenn eine Differenzierung von den üblichen SysML Artefakten unbedingt notwendig ist, da sie sonst das Systemmodell unnötig verkomplizieren und der große Mehrwert einer einheitlichen Systemsprache schwindet.

Der Grundstein, um mit der dargestellten Methode zu entwickeln, liegt letztendlich in der Etablierung des modell-basierten Systems Engineering im Unternehmen. Die Akzeptanz dieses interdisziplinären Entwicklungsansatzes muss in die Unternehmenskultur integriert und die Mitarbeiter müssen schrittweise an die Inhalte herangeführt werden.

6 Fazit und Ausblick

In der vorliegenden Arbeit wurde eine Methodik aufgezeigt, mit der sicherheitsrelevante Systeme über ein Systemmodell, welches als zentrale Informationsdatenbasis dient, entwickelt werden können. Das Hauptziel bestand in der Darstellung der übergeordneten Aktivitäten des Entwicklungsprozesses der ISO 26262 und der Ableitung der Sicherheitsanalysen HARA, FMEA und FTA aus dem Systemmodell. Wie die Validierung in Kapitel 4 zeigt, ist die vollständige und durchgängige Ableitung der ITEM Definition und der HARA anhand des Beispiels möglich. Die ITEM Definition und die HARA sind durchgängig, vollständig und nachvollziehbar darstellbar. Die resultierenden Mehrwerte des modell-basierten Entwickelns sind in Kapitel 5 aufgezeigt.

Das TSC ist in der Validierung nicht vollständig in das Systemmodell integrierbar. Die FMEA und FTA sind mit dem modell-basierten Ansatz methodisch und systematisch darstellbar, die Durchführung ist aber nur über externe Schnittstellen möglich. In der FMEA wurde die Struktur über das Systemmodell aufgebaut. Die Analyse und die Dokumentation fand jedoch in einem externen FMEA Formblatt statt. Die Attribute mussten händisch in das Formblatt übertragen werden. Bei der FTA wurde auf eine externe Fehlerbaumsoftware zurückgegriffen. Der Fehlerbaum wurde systematisch mit Hilfe der modell-basierten Diagramme erstellt, durch die Nutzung einer externen Software befindet sich dieser jedoch nicht im eigentlichen Systemmodell. Die komplette Abbildung der Sicherheitsanalysen über das Systemmodell ist somit momentan, unter den gegebenen Rahmenbedingungen nicht vollständig durchführbar.

An dieser Stelle kann auf der Arbeit aufgebaut werden. Weitere Validierungen der Methodik, bezüglich der Ableitung von Hardwaresicherheitsanforderungen und Softwaresicherheitsanforderungen aus dem Systemmodell währen sinnvoll. Des Weiteren wäre die Validierung der Ableitung von Testfällen aus dem Systemmodell ein weiterer möglicher Ansatz, um die Methodik in der Tiefe weiter zu verifizieren.

Mögliche Untersuchungen können auch in Richtung der Modellierungswerkzeuge führen. Ein Vergleich aller bestehender vollwertigen SysML-Modellierungswerkzeuge mit Hinblick auf die komplette durchgängige Ableitung der Sicherheitsanalysen wäre sinnvoll.

Literaturverzeichnis

[Adl12] Adler, Rasmus; Kemmann, Sören; Schurius, Markus; Allmann, Christian *Modellbasierte Sicherheitsanalysen im BMBF-Förderprojekt e performance.* In: Plödereder, E.; Gesellschaft für Informatik -GI-, Bonn: Automotive - Safety & Security 2012. Proceedings : Sicherheit und Zuverlässigkeit für automobile Informationstechnik, Tagung, 14.-15.11.2012 in Karlsruhe S.(179-194)

[Adl15] Adler, N.: *Modellbasierte Entwicklung funktional sicherer Hardware nach ISO 26262*. Dissertation, KIT Karlsruhe, 2015

[Duo13] Duo S.; Li S.: *A practcable safety modeling methodology for aircraft systems using Altarica*. 3rd International Symposium on Aircraft Airworthiness, ISAA 2013

[Edl15] Edler, F.; Soden, M.; Hankammer, R.: *Fehlerbaumanalyse in Theorie und Praxis, Grundlagen und Anwendung der Methode.* Berlin, Heidelberg: Springer-Verlag, 2015

[Eig14] Eigner, M.; Roubanov D.; Zafirov R. (HRSG.) : *Modellbasierte virtuelle Produktentwicklung.* Berlin, Heidelberg: Springer Vieweg, 2014

[Fre15] Freese, W.: *Erstellung von Fehlerbäumen, Eine strukturierte und systematische Methode.* München: Carl Hanser Verlag, 2015

[For91] Forsberg, K.; Mooz H. 1991: *The Relationship of System Engineering to the Project Cylce*, http://www.damiantgordon.com/Videos/ProgrammingAndAlgorithms/Papers/The%20Relationship%20of%20System%20Engineering%20to%20the%20Project%20Cycle.pdf, Wissenschafltiche Veröffentlichung, Chattanooga: Abruf am 05.12.2017

[Hab12] Haberfellner, R.: *Systems Engineering, Grundlagen und Anwendung.* Zürich: Orell Füssli Verlag AG, 2012

[Hei06] Heimdahl, M.; Joshi, A.; Miller, S.; Whalen, M.: *Model-Based Safety Analysis.* Minneapolis, Minnesota; Cedar Rapids, Iowa, 2006

[Hil12] Hillenbrand, M.: *Funktionale Sicherheit nach ISO 26262 in der Konzeptphase der Entwicklung von Elektrik/ Elektronik Architekturen von Fahrzeugen.* Dissertation, KIT Karlsruhe, 2012

[IAT16] Technischer Überwachungsverein Süd, 2017: https://www.tuev-sued.de/akademie-de/campaigns/die-iatf-16949-2016 Abruf am 14.12.2017

[ISO11] International Organization for Standardization : ISO 26262 *Roadvehicles-Functional Safety*, Part 1 – 10, 15.11.2011

[IEC615] International Electrotechnical Comission, 2017: http://www.iec.ch/about/brochures/pdf/technology/functional_safety.pdf Abruf am 14.12.2017

[Jas17] Jastram M.; 2017: http://se-trends.de/was-ist-eigentlich-das-v-modell/ Abruf am 04.12.2017

[Kle13] Kleiner S.: 2013: Systems Engineering mit dem RFLP-Ansatz. :em engineering methods AG. http://www.em.ag/downloads/Flyer/flyer_systems_engineering.pdf Abruf am 14.12.2017

[Nom17] No Magic, 2017: https://www.nomagic.com/products/cameo-systems-modeler Abruf am 14.12.2017

[Pfe15] Pfeufer, H-J.: *FMEA – Fehler- Möglichkeits- und Einfluss- Analyse.* München: Carl Hanser Verlag, 2015

[Thu04] Thums A.: *Formale Fehlerbaumanalyse*, Dissertation, Universität Augsburg: 2004, https://opus.bibliothek.uni-augsburg.de/opus4/frontdoor/deliver/index/docId/17/file/eVeroeffentlichung.pdf Abruf am 05.12.2017

[Top17] TopEvent FTA, 2017: https://www.fault-tree-analysis.com/ ,Abruf am 04.12.2017

[Wal15] Walden, D.; Roedler, G.; Forsberg, K. et al. (HRSG) : *INCOSE Systems Engineering Handbuch.* GfSE e. V., 2017

[Wal06] Wallentowitz, H.; Reif, K.: *Handbuch Kraftfahrzeugelektronik, Grundlagen, Komponenten, Systeme, Anwendungen.* Wiesbaden: Friedr. Vieweg & Sohn Verlag, GWV Fachverlage GmbH, 2006

[Wei14] Weilkiens, T.: *Systems Engineering mit SysML/UML.* Heidelberg: dpunkt.verlag GmbH, 2014

[Wei17] Weilkiens T., 2017: https://model-based-systems-engineering.com/sysml-tools/ ,Abruf am 30.11.2017

[Wer12] Werdich, M.: *FMEA – Einführung und Moderation*. Wiesbaden: Vieweg+Teubner Verlag, Springer Fachmedien, 2011, 2012

[Zha17] Zhao, X.: *The Integration of Model-Based Safety Analysis and Model-Based Systems Engineering at an Early Stage.* The Woodlands School, Peel District School Board, Mississauga, Ontario Kanada, 2017

Anhang

ITEM Definiton

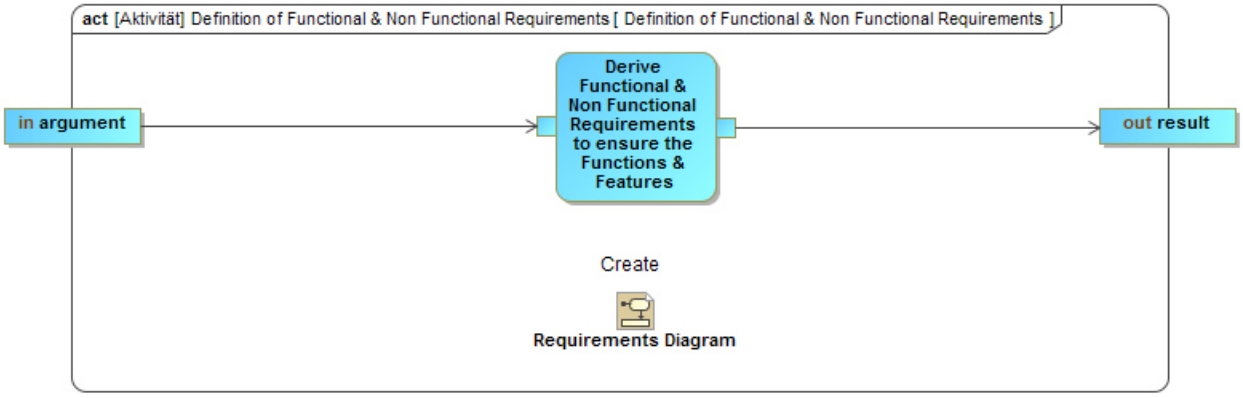


Abbildung 64: Definition of Functional & Non Functional Requirements

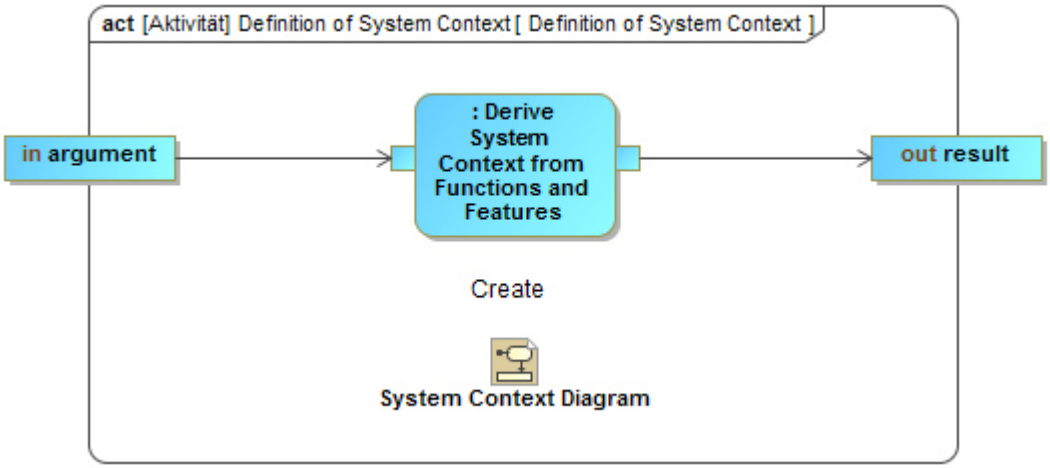


Abbildung 65: Definition of System Context

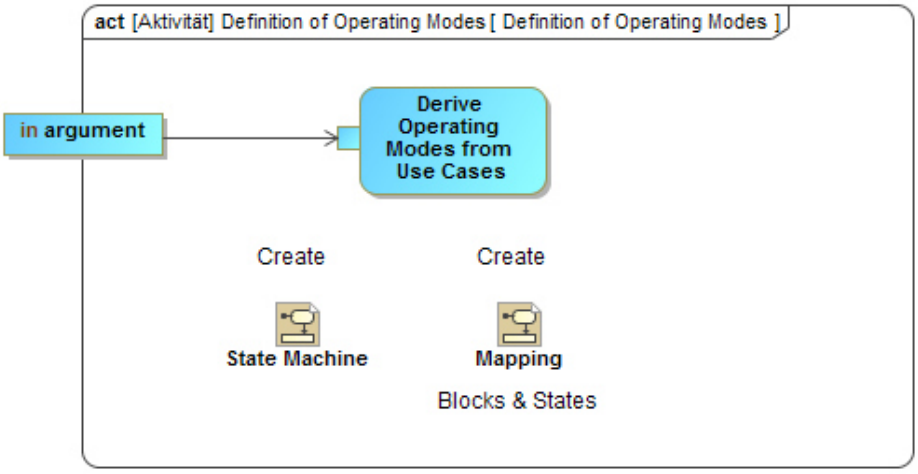


Abbildung 66: Derive Operating Modes from Use Cases

HARA

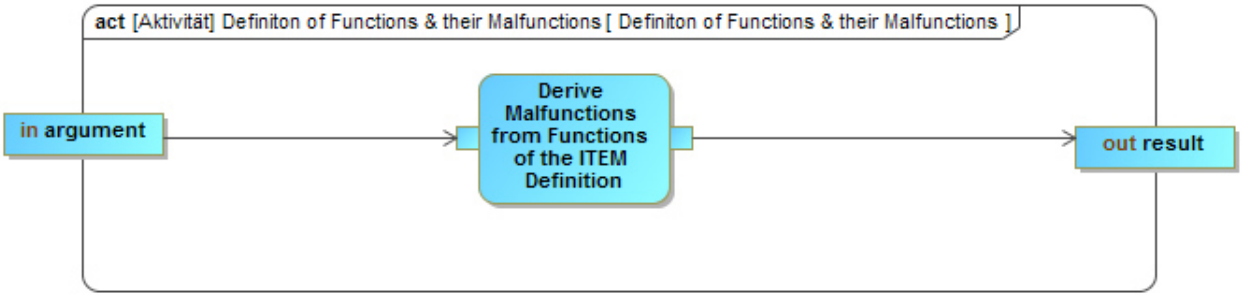


Abbildung 67: Definition of Functions & their Malfunctions

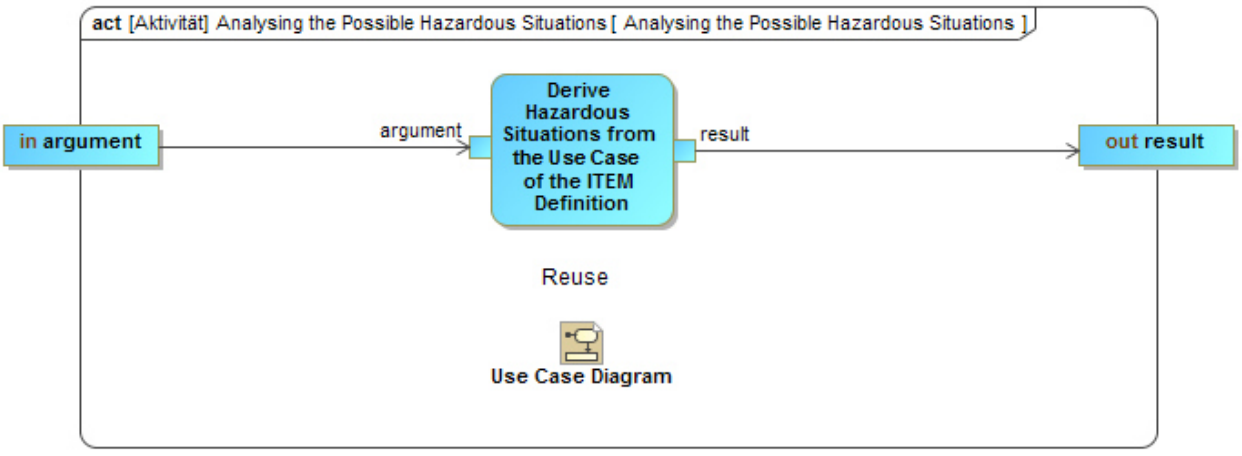


Abbildung 68: Analysing the Possible Hazardous Situations

Abbildung 69: Definition of Safety Goals

TSC

FTA

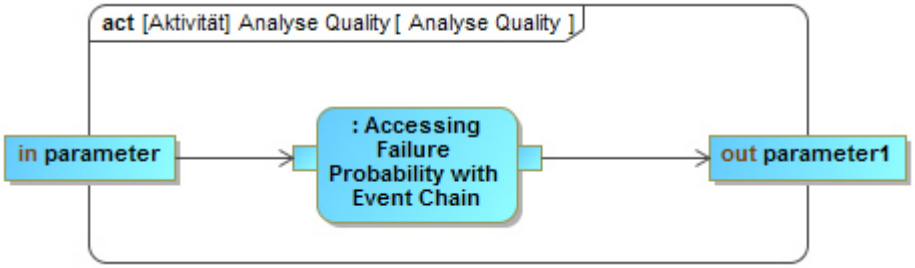


Abbildung 70: Analyse Quality

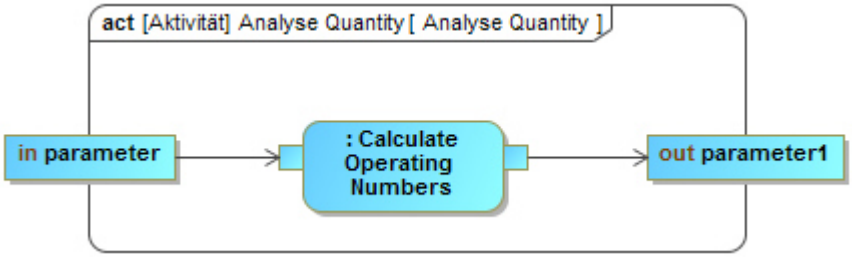


Abbildung 71: Analyse Quantity

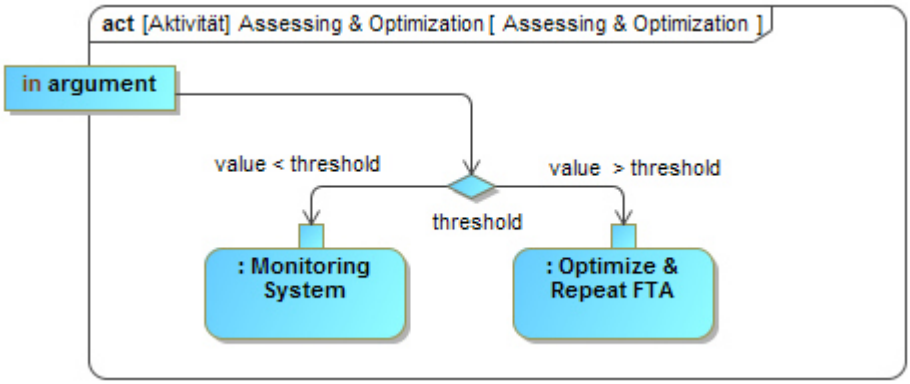


Abbildung 72: Assessing & Optimization

FMEA

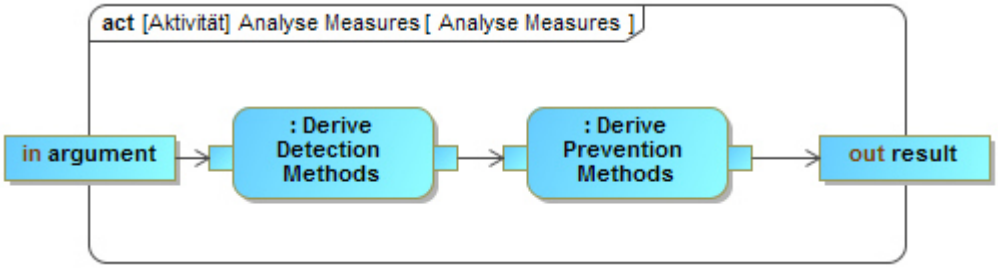


Abbildung 73: Analyse Measures

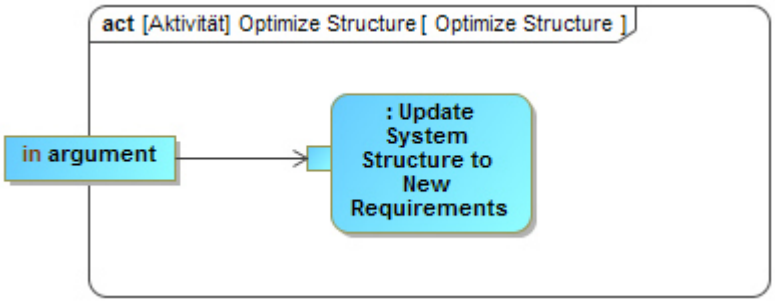


Abbildung 74: Optimize Structure

Hardware Safety Requirements

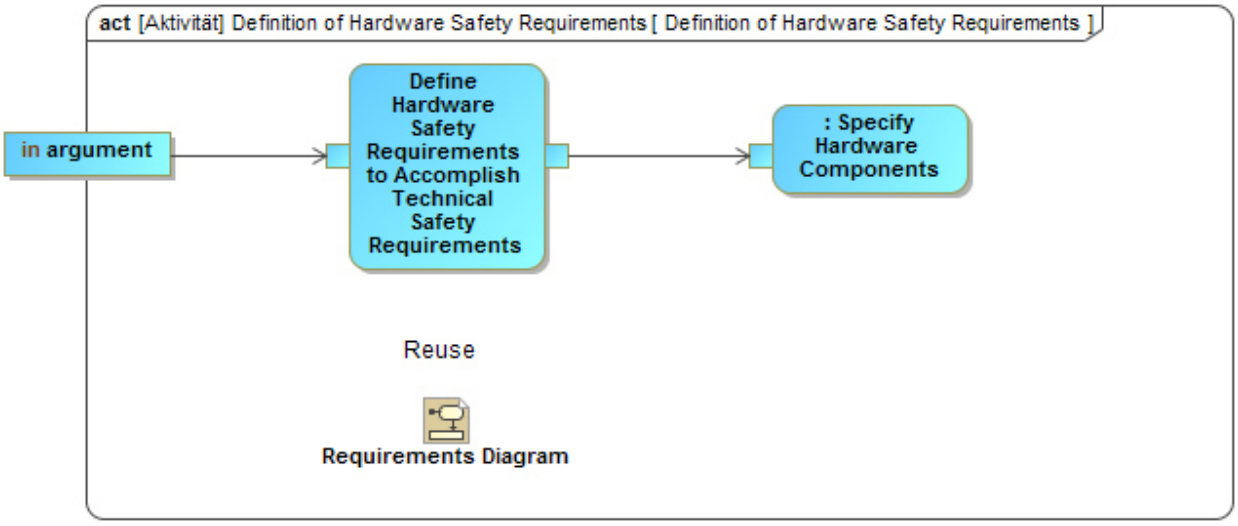


Abbildung 75: Definition of Hardware Safety Requirements

Software Safety Requirements

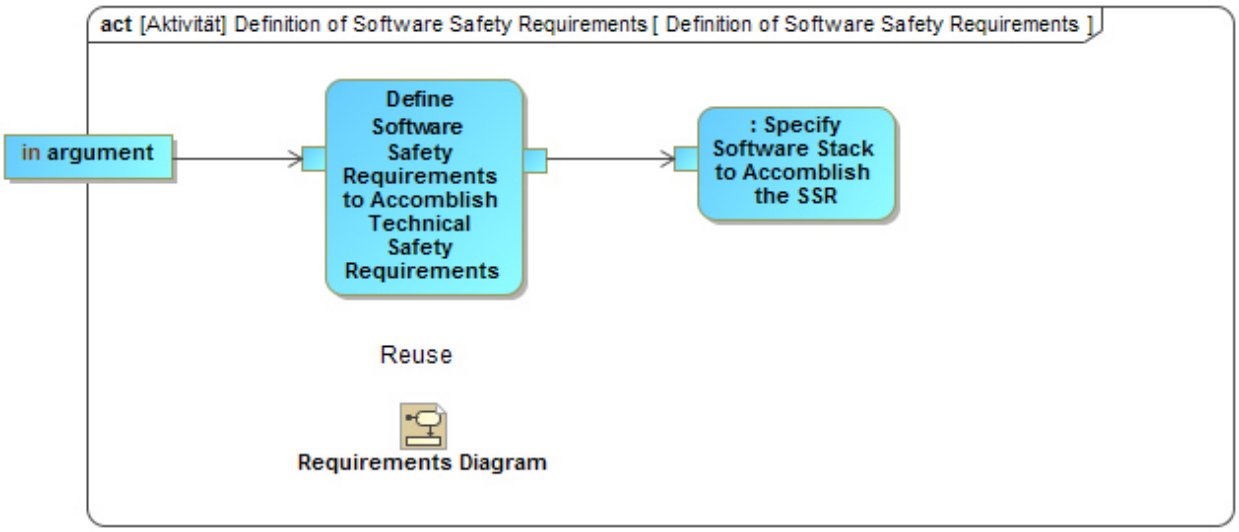


Abbildung 76: Defintion of Software Safety Requirements

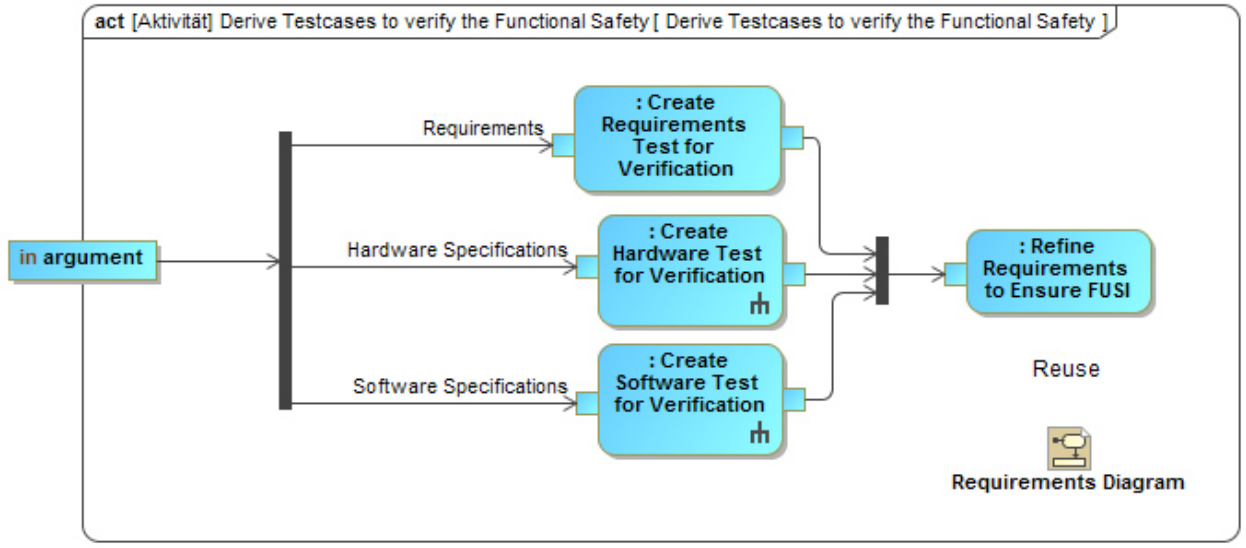


Abbildung 77: Derive Testcases to verify the Functional Safety

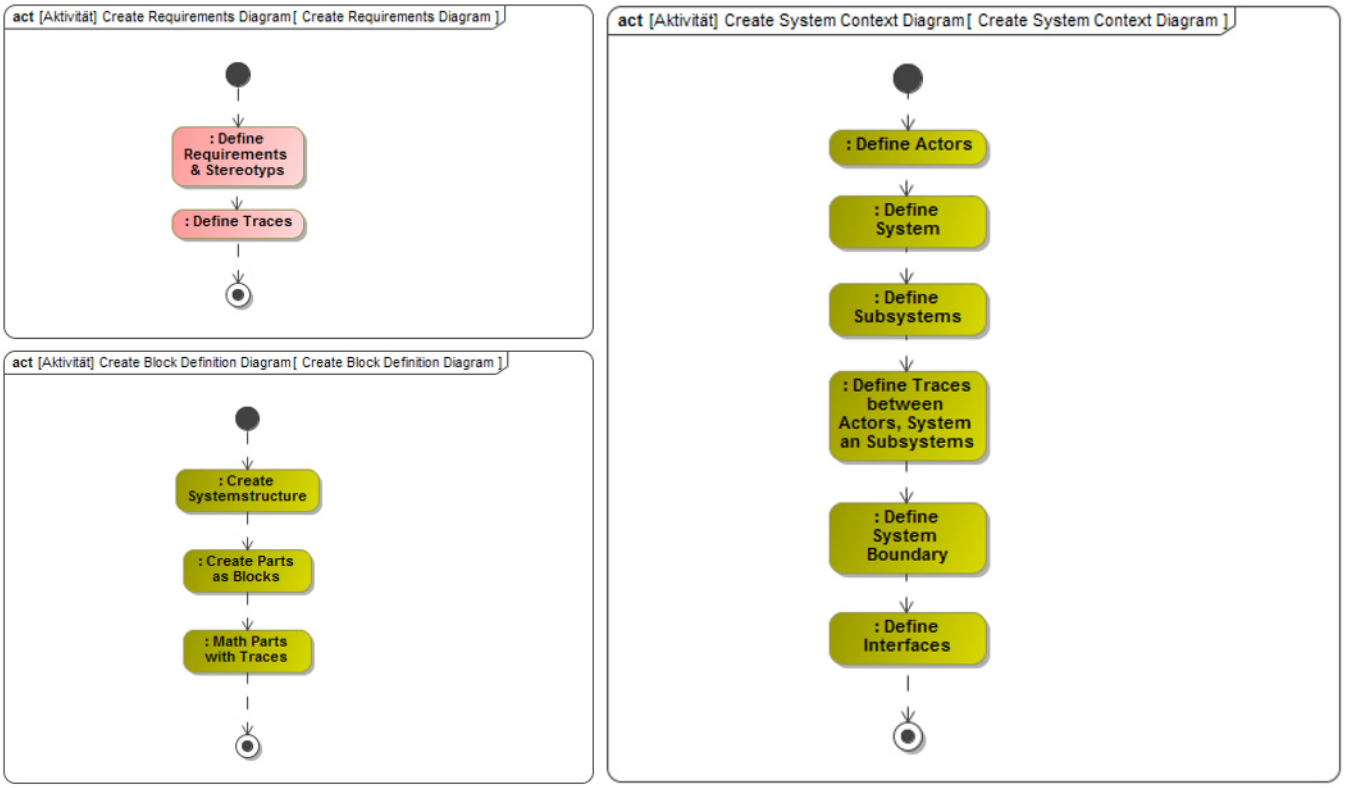


Abbildung 78: Create req, bdd, System Context

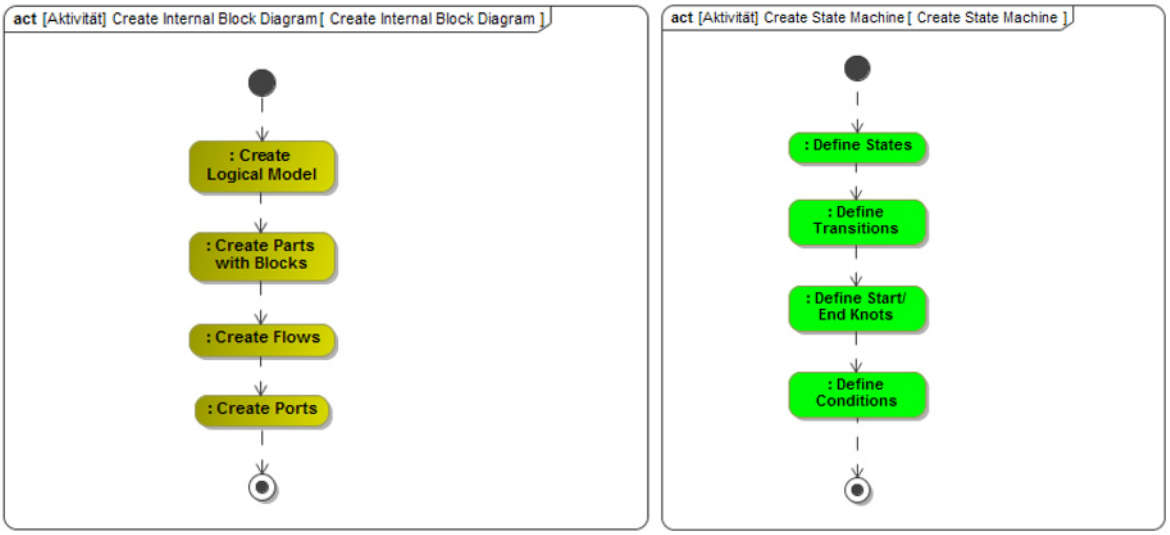


Abbildung 79: Create ibd, Create stm

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „– bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] – ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen."

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Dieses Blatt, mit der folgenden Erklärung, ist nach Fertigstellung der Abschlussarbeit durch den Studierenden auszufüllen und jeweils mit Originalunterschrift als letztes Blatt in das Prüfungsexemplar der Abschlussarbeit einzubinden.
Eine unrichtig abgegebene Erklärung kann -auch nachträglich- zur Ungültigkeit des Studienabschlusses führen.

Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: Hamester

Vorname: Michael

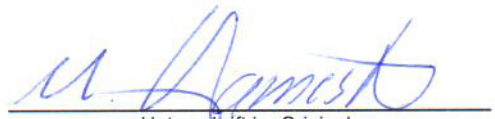
dass ich die vorliegende Masterarbeit bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

Methodik für die systematische Systemmodellierung als Basis für die Entwicklung und Bewertung sicherheitsrelevanter Systeme

ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

- die folgende Aussage ist bei Gruppenarbeiten auszufüllen und entfällt bei Einzelarbeiten -

Die Kennzeichnung der von mir erstellten und verantworteten Teile der -bitte auswählen- ist erfolgt durch:

Darmstadt	31.01.2018	
Ort	Datum	Unterschrift im Original