UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO DE ESPECIALIZAÇÃO EM SISTEMAS EMBARCADOS
PARA A INDÚSTRIA AUTOMOTIVA

LUCIANA CAMINSKI

ISO 26262: SEGURANÇA FUNCIONAL NO DESENVOLVIMENTO DE SISTEMAS AUTOMOTIVOS

MONOGRAFIA DE ESPECIALIZAÇÃO

CURITIBA
2018

LUCIANA CAMINSKI

ISO 26262: SEGURANÇA FUNCIONAL NO DESENVOLVIMENTO DE SISTEMAS AUTOMOTIVOS

Monografia de Especialização, apresentado ao Curso de Especialização em Sistemas Embarcados para a Indústria Automotiva, do Departamento Acadêmico de Eletrônica, da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Especialista.

Orientador: Prof. Dr. Kleber Kendy Horikawa Nabas

CURITIBA
2018

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Curitiba

Diretoria de Pesquisa e Pós-Graduação
Departamento Acadêmico de Eletrônica
Curso de Especialização em Sistemas Embarcados para Indústria Automotiva

TERMO DE APROVAÇÃO

ISO 26262: SEGURANÇA FUNCIONAL NO DESENVOLVIMENTO DE SISTEMAS AUTOMOTIVOS

por

LUCIANA CAMINSKI

Esta Monografia foi apresentada em 08 de junho de 2018 como requisito parcial para a obtenção do título de Especialista em Sistemas Embarcados para a Indústria Automotiva. A candidata foi arguida pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

_____
Prof. Dr. Kleber Kendy Horikawa Nabas
Orientador e Coordenador do Curso

_____
Prof. Dr. Edenilson José da Silva
Membro titular

_____
Prof. MSc. Omero Francisco Bertol
Membro titular

- O Termo de Aprovação assinado encontra-se na Coordenação do Curso -

RESUMO

CAMINSKI, Luciana. **ISO 26262:** segurança funcional no desenvolvimento de sistemas automotivos. 2018. 37 f. Monografia (Curso de Especialização em Sistemas Embarcados para a Indústria Automotiva), Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

Os avanços tecnológicos dos últimos anos resultaram em sistemas embarcados automotivos cada vez mais complexos. O uso de tecnologias mais avançadas, aliado ao desenvolvimento e integração de novas funcionalidades, proporciona diversos benefícios para os clientes, a sociedade e o meio ambiente. Por outro lado, essa complexidade cada vez maior também leva ao aumento da criticidade das aplicações e ao aumento da probabilidade de falhas de *hardware* e de *software*. São conhecidos diversos casos de acidentes que resultaram em prejuízos materiais e vítimas fatais que foram atribuídos a problemas em sistemas automotivos, ou que poderiam ter sido minimizados e até evitados através do uso de melhores sistemas veiculares de segurança. Considerando todos esses fatos, a segurança funcional dos sistemas embarcados automotivos tornou-se um grande desafio para o desenvolvimento automotivo atual. Nesse contexto, a ISO 26262 estabelece uma série de diretrizes a serem utilizadas no projeto de sistemas elétricos e/ou eletrônicos relacionados à segurança que estão instalados em veículos rodoviários. A norma propõe um ciclo de vida de segurança automotiva desde o gerenciamento até a desativação e também fornece uma série de recomendações a serem usadas ao longo da fase conceitual e do desenvolvimento dos produtos a nível de sistema, *hardware* e *software*. Durante a fase conceitual é realizado um dos processos mais importantes do ciclo de vida de segurança, a análise de perigos e avaliação de riscos, a qual identifica os riscos em potencial e estima a probabilidade de exposição, a controlabilidade e a severidade dos eventos perigosos causados por um mau funcionamento de cada um dos itens em desenvolvimento, sendo que a junção desses parâmetros determina o nível de integridade de segurança automotiva, chamado de ASIL. Através dessa análise, são então determinados os objetivos de segurança, os quais são detalhados em requisitos de segurança funcional e sucessivamente refinados durante as fases subsequentes até os requisitos técnicos de segurança de *hardware* e *software*. O desenvolvimento de produto a nível de *software* baseia-se no modelo “V”, sendo que a especificação dos requisitos, projeto e implementação estão no ramo esquerdo e a integração e teste e verificação de requisitos no ramo direito. A norma sugere vários métodos a serem usados para o teste de unidade de *software* e de integração de *software*, tais como teste baseado em requisitos, de interface, de injeção de falha, de uso de recursos e de comparação *back-to-back*. O método de teste mais adequado para um certo nível de integridade do produto e que representa o grau de rigor que deve ser aplicado na verificação, a fim de evitar um risco residual no produto final, é determinado através do ASIL correspondente. A ISO 26262 recomenda o uso da técnica de injeção de falha como método de teste de *software*, sendo altamente recomendado para atender aos ASILs mais críticos C e D. Um teste de injeção de falha tem como objetivo introduzir falhas em um item utilizando meios específicos e testar se os mecanismos tolerantes a falhas são eficientes o suficiente para manter o sistema de acordo com os objetivos de segurança esperados.

Palavras chave: ISO 26262. Segurança funcional automotiva. Teste de *software*.

ABSTRACT

CAMINSKI, Luciana. **ISO 26262:** functional safety on development of automotive systems. 2018. 37 f. Monografia (Curso de Especialização em Sistemas Embarcados para a Indústria Automotiva), Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

The technological advances of recent years have resulted in embedded automotive systems increasingly complex. The use of more advanced technologies, combined with the development and integration of new functionalities, provides several benefits for customers, society and the environment. On the other hand, this increasing complexity also leads to an increase in the criticality of the applications and an increase in the probability of hardware and software failures. Several cases of accidents are known that have resulted in material damage and fatalities that have been attributed to problems in automotive systems or that could have been minimized or even avoided through the use of improved safety vehicle systems. Considering all these facts, the functional safety of automotive embedded systems has become a major challenge for the current automotive development. In this context, the ISO 26262 establishes several guidelines to be used in the design of electrical and/or electronic safety-related systems that are installed in road vehicles. The standard proposes an automotive safety lifecycle from management to decommissioning and also provides several recommendations to be used along the concept phase and product development at system, hardware and software level. During the concept phase, it is realized one of the most important processes of the safety lifecycle, the hazard analysis and risk assessment, which identifies the potential risks and estimate the probability of exposure, the controllability and the severity of the hazardous events caused by malfunctioning behavior of the items under development, and the junction of these parameters determines the automotive safety integrity level, called ASIL. With this analysis, the safety goals are then determined, which are detailed in functional safety requirements and successively refined during subsequent phases up to hardware and software technical safety requirements. The product development at the software level is based on the “V” model, with the requirements specification, design and implementation on the left branch and the integration and testing and the verification of requirements on the right branch. The standard suggests several methods to be used for the software unit and integration testing, such as requirements-based test, interface test, fault injection test, resource usage test and back-to-back comparison test. The most appropriate test method for a certain level of product integrity and that represents the degree of rigour that must be applied in the verification, in order to avoid a residual risk in the final product, is determined by the corresponding ASIL. ISO 26262 recommends the use of the fault injection technique as a software test method, and is highly recommended to meet the most critical ASILs C and D. A fault injection test aims to introduce faults into the item using specific means and test if the fault-tolerant mechanisms are efficient enough to keep the system according to the expected safety goals.

Keywords: ISO 26262. Automotive functional safety. Software testing.

LISTA DE FIGURAS

Figura 1 – Estrutura geral da ISO 26262 16

Figura 2 – Ciclo de vida de segurança 19

Figura 3 – Estrutura dos requisitos de segurança 21

Figura 4 – Determinação do ASIL 25

Figura 5 – Etapas do desenvolvimento de *software* 27

LISTA DE TABELAS

Tabela 1 – Classes de severidade 23
Tabela 2 – Classes de probabilidade de exposição 24
Tabela 3 – Classes de controlabilidade.....24
Tabela 4 – Determinação do ASIL 25

LISTA DE SIGLAS

ABS	*Anti-lock Braking System*
ACC	*Adaptive Cruise Control*
ADAS	*Advanced Driver Assistance System*
AIS	*Abbreviated Injury Scale*
ASIL	*Automotive Safety Integrity Level*
EBS	*Emergency Brake System*
ECU	*Electronic Control Unit*
ESP	*Electronic Stability Program*
E/E	Elétrico e/ou Eletrônico
FMEA	*Failure Mode and Effect Analysis*
HIL	*Hardware-in-the-loop*
IEC	*International Electrotechnical Commission*
ISO	*International Organization for Standardization*
LOC	*Lines of Code*
MIL	*Model-in-the-loop*
OEM	*Original Equipment Manufacturer*
PIL	*Processor-in-the-loop*
QM	*Quality Management*
SIL	*Software-in-the-loop*

SUMÁRIO

1 INTRODUÇÃO 9

1.1 PROBLEMA 9
1.2 OBJETIVOS 10
1.2.1 Objetivo Geral 11
1.2.2 Objetivos Específicos 11
1.3 JUSTIFICATIVA 11
1.4 ESTRUTURA DO TRABALHO 12

2 FUNDAMENTAÇÃO TEÓRICA 14

2.1 NORMA DE SEGURANÇA FUNCIONAL PARA A INDÚSTRIA AUTOMOTIVA 14
2.2 ASPECTOS GERAIS DA ISO 26262 14
2.2.1 Escopo 14
2.2.2 Principais componentes 15
2.2.3 Gerenciamento da segurança funcional 18
2.3 NÍVEIS DE INTEGRIDADE DE SEGURANÇA AUTOMOTIVA (ASIL) 22
2.4 SOFTWARE 26
2.5 TESTE DE SOFTWARE 28
2.5.1 Teste de unidade de software 29
2.5.2 Teste de integração de software 29
2.5.3 Teste de injeção de falha 30

3 CONSIDERAÇÕES FINAIS 33

REFERÊNCIAS 34

ANEXOS 36

1 INTRODUÇÃO

Na indústria automotiva, vários avanços nos últimos anos resultaram em uma transição de sistemas puramente mecânicos para sistemas controlados eletronicamente. Hoje em dia, um carro moderno pode conter dezenas de unidades de controle eletrônico (ECU - *Electronic Control Unit*) que hospedam milhões de linhas de código (LOC - *Lines of Code*) e estão interconectadas por diversos barramentos e redes. Há ainda diversos sensores aquisitando dados e atuadores executando inúmeras ações [5].

A tendência de substituir sistemas mecânicos tradicionais por sistemas embarcados modernos permite a implantação de estratégias de controle mais avançadas, que proporcionam benefícios adicionais para o cliente e o meio ambiente tais como economia de combustível, redução da emissão de poluentes, confiabilidade, conforto entre outros [8].

Por outro lado, a crescente complexidade dos sistemas e o alto grau de integração e criticidade também pode levar ao aumento da probabilidade de falhas e, com isso, novos desafios emergiram no gerenciamento da segurança funcional. *Anti-lock Braking Systems* (ABS), *Electronic Stability Program* (ESP), *Adaptive Cruise Control* (ACC) e *Emergency Brake System* (EBS) são alguns dos exemplos de sistemas críticos de segurança em automóveis hoje em dia, que consistem em uma arquitetura sistêmica de interação e interfaces complexas [5][8].

1.1 PROBLEMA

É conhecido o caso do problema da Toyota com o pedal do acelerador que matou uma pessoa. Por causa disso, a Toyota fez o recall de aproximadamente 8,5 milhões de carros e também teve que ressarcir pessoas afetadas. Isso causou uma grande perda de tempo e dinheiro, tanto para a companhia como para os clientes. Outro caso que ocorreu foi o de um chip defeituoso em um conversor DC/DC de um Honda Civic, o qual foi causado por um curto-circuito para a terra que danificou o fusível principal e fez parar o motor. Na Europa, aproximadamente 3751 carros foram afetados e foram alvos de recall para o fabricante. Outro caso bem conhecido

é o da perda de função do sistema de direção de carros Mazda. Três acidentes são conhecidos e analisados atualmente por agentes de investigação [6].

Em vista da evidência de milhares de vítimas ao redor do mundo causadas por acidentes de trânsito e outras devido a problemas em sistemas automotivos, a indústria automotiva está sob pressão a fim de fornecer novos e melhores sistemas veiculares de segurança, que vão desde sistemas de *airbag* a sistemas avançados de assistência ao condutor (ADAS - *Advanced Driver Assistance System*) extremamente complexos, com capacidade de previsão e prevenção de acidentes [2].

Os ADAS são sistemas que estão se desenvolvendo rapidamente. Eles contam com o sistema de percepção do ambiente para ajudar o motorista a evitar acidentes, sendo que os limites do sistema usados na avaliação incluem o condutor do veículo dentro da situação de tráfego. Este desenvolvimento exige a necessidade de métodos e ferramentas de teste para a avaliação dos diferentes aspectos desses sistemas. Isso significa que além da verificação das funções, também é necessário validar os efeitos positivos destes sistemas na segurança do trânsito. Para identificar os requisitos para estas avaliações mais globais, todo o sistema do veículo, o motorista e o ambiente, tem que ser levado em conta. Além disso, também precisa ser provado que os benefícios superam as desvantagens potenciais de tais sistemas, como perigos devido a falhas ou reações falsas [7].

A segurança é uma das questões-chave do desenvolvimento automotivo futuro. O desenvolvimento e integração de novas funcionalidades irão fortalecer cada vez mais a necessidade de processos de desenvolvimento de sistemas seguros, como os propostos pela norma ISO 26262, e a necessidade de fornecer evidências de que todos os objetivos razoáveis de segurança do sistema são satisfeitos [1].

1.2 OBJETIVOS

Nesta seção é apresentado o objetivo geral e os objetivos específicos do trabalho, relativos ao problema anteriormente apresentado.

1.2.1 Objetivo Geral

Apresentar os conceitos e processos recomendados pela norma de segurança funcional ISO 26262 para o desenvolvimento de sistemas automotivos.

1.2.2 Objetivos Específicos

Para atender ao objetivo geral neste trabalho de conclusão de curso, os seguintes objetivos específicos serão abordados:

- Identificar o escopo, a aplicabilidade e a abrangência da ISO 26262 na indústria automotiva;
- Levantar os principais procedimentos e conceitos estabelecidos pela norma a serem usados no gerenciamento da segurança funcional e nos processos de desenvolvimento;
- Apresentar os métodos propostos pela ISO 26262 para o desenvolvimento e teste de *software* em sistemas automotivos.

1.3 JUSTIFICATIVA

Com a tendência de crescente complexidade tecnológica de sistemas embarcados, conteúdo de *software* e implementação mecatrônica, existem cada vez mais riscos de falhas sistemáticas, de *software* e de *hardware*, tornando o projeto de veículos modernos em nível funcional ainda mais desafiador para os fabricantes de carros, a fim de fornecer sistemas com alto nível de segurança. A ISO 26262 inclui orientação para evitar esses riscos, fornecendo processos e requisitos apropriados [1] [7].

Assim, com o propósito de fornecer uma diretriz para o projeto adequado e validação de segurança de novos veículos no nível funcional, a ISO 26262 para segurança funcional em veículos foi derivada da norma geral IEC 61508 destinada à segurança funcional em sistemas eletrônicos relacionados à segurança [1] [7].

Um aspecto chave da ISO 26262 é a análise de perigos e a classificação resultante das metas de segurança derivadas em termos de *Automotive Safety Integrity Levels* (ASILs). Entre outros, esses níveis determinam limites superiores

para a taxa de falha aceitável da função investigada. No caso de possíveis falhas, testes são necessários para obter a aprovação de segurança funcional do sistema de acordo com a ISO 26262 [1] [7].

Por conta de a publicação se tratar de uma norma e ser pública, os advogados tratam a ISO 26262 como o estado da arte técnico. O estado da arte técnico é o nível mais alto de desenvolvimento de um dispositivo ou processo em um determinado tempo. De acordo com a lei alemã os produtores de carro em geral são responsáveis pelos danos causados a uma pessoa pelo mau funcionamento de um produto. No entanto, se o mau funcionamento não puder ser detectado pelo estado da arte técnico, a responsabilidade é excluída (lei alemã sobre responsabilidade de produto (§ 823 Abs. 1 BGB, § 1 ProdHaftG)) [4].

Toda a indústria automotiva está ativa para aprender as ideias e o conteúdo da norma e adotá-la corretamente em seus produtos. Não é apenas um desafio para as OEMs, mas também para todos os fornecedores no desenvolvimento do produto e na cadeia de produção. No futuro, todos os fabricantes automotivos deverão demostrar que todos os sistemas estão alinhados com a ISO 26262 desde o conceito até as fases do processo de desenvolvimento do produto atuais. Por ter a certificação da ISO 26262, promoverá uma alta confiança aos clientes para comprarem automóveis, nos quais a prevenção de acidentes e a redução de riscos sejam aceitáveis. Isso ajuda a evitar erros na implementação, prevenir recalls caros e proteger o nome e a reputação da marca de qualquer dano [2] [5].

A norma é bastante volumosa com aproximadamente 400 páginas e dividida em 10 partes. Como a ISO 26262 é bem complexa e trata de vários assuntos distintos, o presente trabalho se concentrará nos pontos principais da norma e no desenvolvimento e teste de *software*.

1.4 ESTRUTURA DO TRABALHO

O trabalho terá a estrutura abaixo apresentada.

Capítulo 1 - INTRODUÇÃO: será abordado o tema, o problema, os objetivos da pesquisa, a justificativa e a estrutura geral do trabalho.

Capítulo 2 – FUNDAMENTAÇÃO TEÓRICA: serão apresentados os aspectos gerais da ISO 26262, gerenciamento da segurança funcional, explanação sobre os

níveis de integridade de segurança automotiva propostos pela norma e metodologia para desenvolvimento e teste de *software*.

Capítulo 3 – CONSIDERAÇÕES FINAIS: será apresentada uma conclusão e reflexão a respeito do conteúdo apresentado. Além disto, serão sugeridos trabalhos futuros que poderiam ser realizados a partir do estudo realizado.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 NORMA DE SEGURANÇA FUNCIONAL PARA A INDÚSTRIA AUTOMOTIVA

A IEC 61508, "*Functional safety of electrical/electronic/programmable electronic safety-related systems*", é designada pelo IEC como uma norma genérica e uma publicação básica de segurança. Isso significa que os diversos setores industriais irão basear suas próprias normas para segurança funcional nos requisitos da IEC 61508. Na indústria automotiva, há uma série de problemas em se aplicar a IEC 61508 diretamente [1].

A ISO 26262, "*Road vehicles – Functional safety*" é a adaptação da IEC 61508 para atender às necessidades específicas do setor de aplicação de sistemas Elétricos e/ou Eletrônicos (E/E) em veículos rodoviários. Essa adaptação se aplica a todas as atividades durante o ciclo de vida de segurança de sistemas relacionados à segurança compostos de componentes elétricos, eletrônicos e de *software* [1].

2.2 ASPECTOS GERAIS DA ISO 26262

2.2.1 Escopo

A ISO 26262, publicada em novembro de 2011, aborda a segurança funcional de veículos rodoviários e irá tornar-se a norma internacional de segurança mais importante da indústria automotiva. Ela destina-se a ser aplicada a sistemas relacionados à segurança que incluem um ou mais sistemas Elétricos e/ou Eletrônicos e que estão instalados em carros de passageiros produzidos em série de até 3500 kg. A norma não aborda sistemas E/E exclusivos em veículos de finalidade especial, como os veículos concebidos para motoristas com deficiência [1][3].

Mesmo que a ISO 26262 ainda não seja mandatória para os veículos pesados, várias montadoras deste tipo de veículo já estão investigando e se preparando para a adoção da norma. Estima-se que começará a ser aplicada para caminhões e ônibus por volta de 2018.

Sistemas e seus componentes liberados para produção ou que já estão em desenvolvimento antes da data de publicação da ISO 26262 estão isentos do

escopo. Para desenvolvimentos posteriores ou alterações, baseados em sistemas e componentes liberados pra produção antes da publicação da norma, apenas as modificações serão desenvolvidas em conformidade com a ISO 26262 [1].

A ISO 26262 aborda possíveis perigos causados pelo mau funcionamento de sistemas E/E relacionados à segurança, incluindo a interação destes sistemas. Ela não aborda os perigos relacionados a choque elétrico, fogo, fumaça, calor, radiação, toxicidade, inflamabilidade, reatividade, corrosão, liberação de energia e perigos semelhantes, a menos que diretamente causados pelo mau funcionamento de tais sistemas. A norma também não aborda o desempenho nominal dos sistemas E/E, mesmo que existam para eles normas específicas de desempenho funcional [1].

2.2.2 Principais componentes

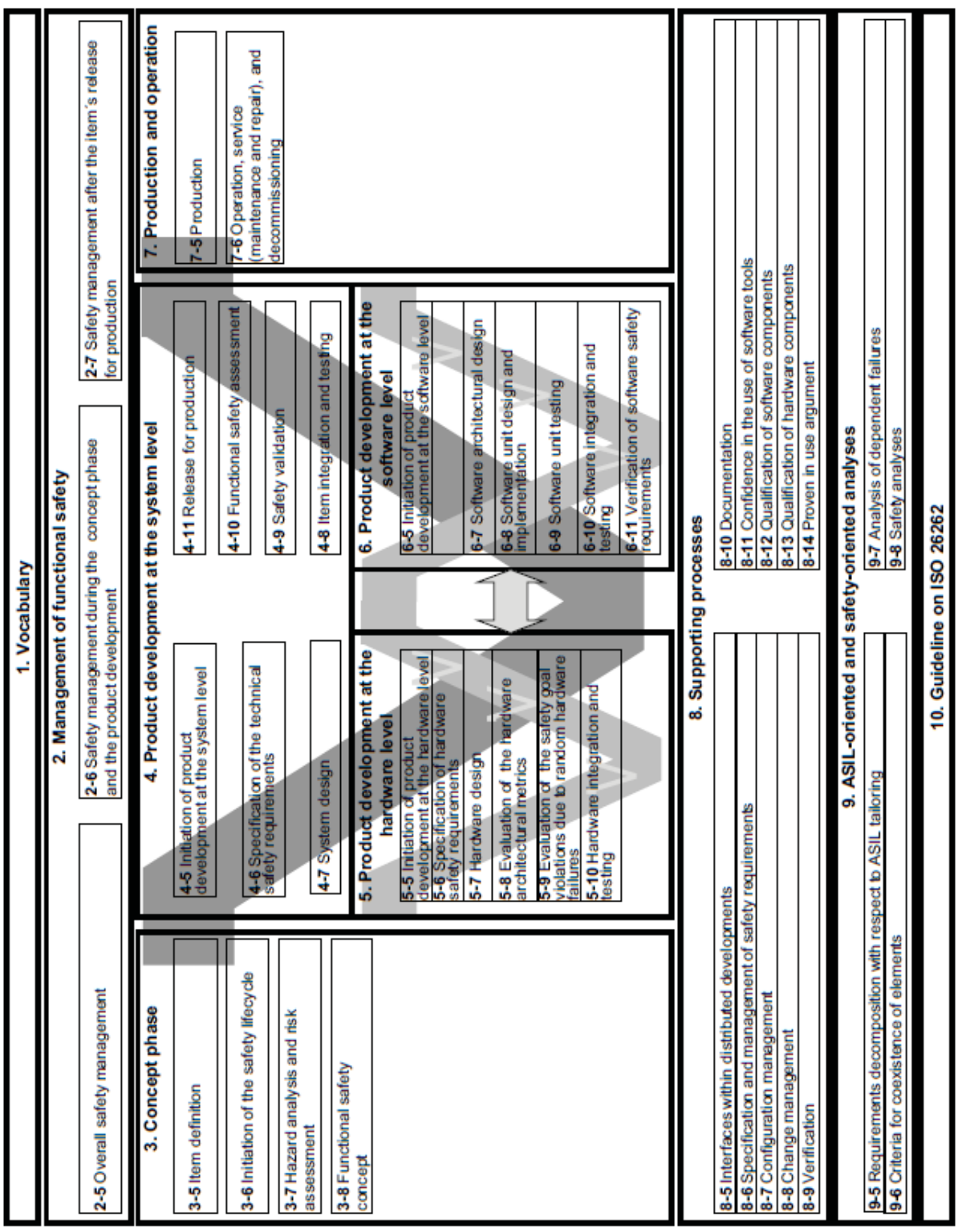
A norma ISO 26262 usa um sistema de passos para gerenciar a segurança funcional e regular o desenvolvimento de produtos em nível de sistema, *hardware* e *software* [4]. Em geral, a ISO 26262 fornece:

- Um ciclo de vida de segurança automotiva (gerenciamento, desenvolvimento, produção, operação, serviço, desativação) e suporta a adaptação das atividades necessárias durante as fases desse ciclo de vida. Diversas regulamentações e recomendações são fornecidas ao longo do processo de desenvolvimento do produto, desde o desenvolvimento conceitual até a desativação [4].
- Uma abordagem baseada nos riscos específicos para o setor automotivo, com o objetivo de determinar as classes de riscos (níveis de integridade de segurança automotiva, ASILs - *Automotive Safety Integrity Levels*). Os ASILs então são usados para especificar os requisitos de segurança necessários para alcançar um risco residual aceitável para um sistema ou componente. Com base nisso, a norma também fornece requisitos para as medidas de confirmação e validação para garantir que um nível de segurança suficiente e aceitável está sendo alcançado [4].

A Figura 1 mostra a estrutura geral da ISO 26262, a qual é baseada no modelo “V” como um modelo de processo de referência para as diferentes fases do

desenvolvimento de produto. Os “V”s sombreados representam a interconexão entre as partes ISO 26262-3, ISO 26262-4, ISO 26262-5, ISO 26262-6 e ISO 26262-7 [1].

Figura 1 – Estrutura geral da ISO 26262



Fonte: ISO 26262 (2011).

Os 10 volumes que constituem a edição atual da norma são descritos brevemente a seguir [2]:

- **Parte 1: Vocabulário**

Esta parte especifica os termos, definições e termos abreviados para aplicação em todas as partes da ISO 26262.

- **Parte 2: Gerenciamento da segurança funcional**

Esta parte especifica os requisitos no gerenciamento da segurança funcional para aplicações automotivas. Estes requisitos abrangem as atividades de gerenciamento de projeto de todas as fases do ciclo de vida de segurança e consistem em requisitos independentes do projeto, em requisitos dependentes do projeto a serem seguidos durante o desenvolvimento e em requisitos que se aplicam após o lançamento para a produção.

- **Parte 3: Fase conceitual**

Esta parte especifica os requisitos da fase conceitual para aplicações automotivas. Esses requisitos incluem a definição de item, o início do ciclo de vida de segurança, a análise de perigos e avaliação de risco e o conceito de segurança funcional.

- **Parte 4: Desenvolvimento de Produto: Nível de sistema**

Esta parte especifica os requisitos no desenvolvimento de produtos no nível do sistema. Estes incluem requisitos no início do desenvolvimento do produto a nível de sistema, a especificação dos requisitos técnicos de segurança, conceito técnico de segurança, projeto do sistema, integração e teste de itens, validação de segurança, avaliação da segurança funcional e lançamento do produto.

- **Parte 5: Desenvolvimento de Produto: Nível de *Hardware***

Esta parte especifica os requisitos no desenvolvimento de produtos no nível do *hardware*. Estes incluem requisitos para o início do desenvolvimento do produto a nível de *hardware*, a especificação dos requisitos de segurança do *hardware*, projeto de *hardware*, métricas arquitetônicas de *hardware*, integração e testes de *hardware* e avaliação da violação de metas de segurança devido a falhas de *hardware* aleatórias.

- **Parte 6: Desenvolvimento de Produto: Nível de *Software***

Esta parte especifica os requisitos no desenvolvimento de produtos no nível do *software*. Estes incluem requisitos para o início do desenvolvimento do produto a

nível de *software*, a especificação dos requisitos de segurança do *software*, projeto de arquitetura de *software*, projeto e teste de unidade de *software* e implementação, integração e teste de *software* e verificação dos requisitos de segurança do *software*.

- **Parte 7: Produção e Operação**

Esta parte especifica os requisitos para a produção, operação, serviços e desativação.

- **Parte 8: Processos de apoio**

Esta parte especifica os requisitos para os processos de apoio. Estes incluem interfaces em desenvolvimentos distribuídos, gerenciamento geral dos requisitos de segurança, gerenciamento de configurações, gerenciamento de mudanças, verificação, documentação, qualificação de ferramentas de *software*, qualificação dos componentes de *hardware* e *software* e argumento “comprovado na prática”.

- **Parte 9: Análises orientadas à segurança e ao ASIL**

Esta parte especifica os requisitos para análises orientadas à segurança e ao ASIL. Estes incluem decomposição do ASIL, critérios para a coexistência de elementos de diferentes ASIL, análise de falhas dependentes e análises de segurança.

- **Parte 10: Orientação sobre a ISO 26262**

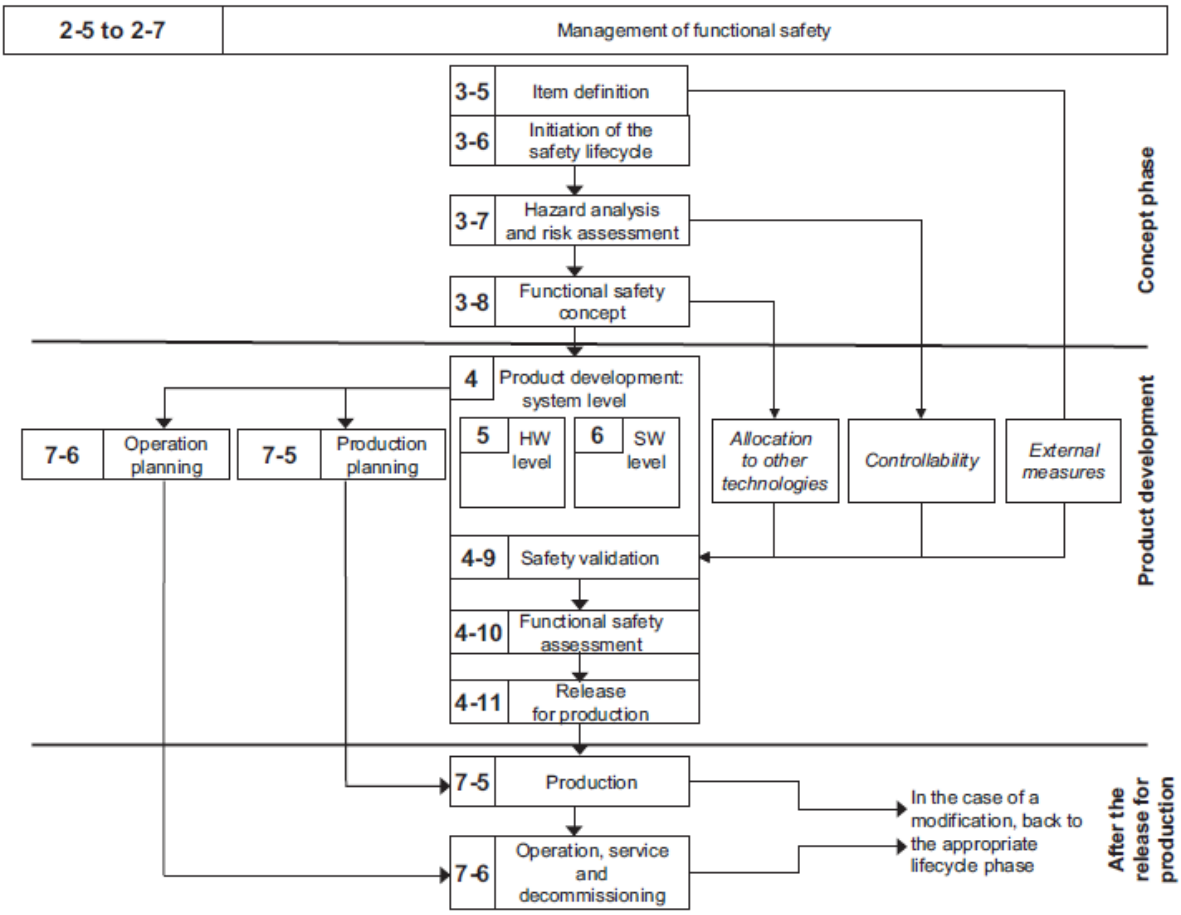
Esta parte tem caráter apenas informativo. Ela fornece um panorama geral informativo da ISO 26262, bem como explicações adicionais destinadas a melhorar a compreensão das outras partes da ISO 26262. Esta parte descreve os conceitos gerais da ISO 26262 a fim de facilitar o entendimento. A explanação se expande de conceitos gerais para conteúdos específicos.

2.2.3 Gerenciamento da segurança funcional

O ciclo de vida de segurança automotiva apresentado pela ISO 26262 (Figura 2) abrange as principais atividades de segurança durante a fase conceitual, desenvolvimento do produto, produção, operação, serviço e desativação. Planejamento, coordenação e documentação das atividades de todas as fases do ciclo de vida de segurança são tarefas chave de gerenciamento. O gerenciamento

do ciclo de vida de segurança automotiva inclui a necessidade de um gerente de segurança, o desenvolvimento de um plano de segurança e a definição de medidas de confirmação incluindo auditoria, avaliação e revisão de segurança [1][4].

Figura 2 – Ciclo de vida de segurança



Fonte: ISO 26262 (2011).

A tarefa inicial do ciclo de vida de segurança, dando início à fase conceitual, é desenvolver uma definição e descrição dos itens a serem desenvolvidos em relação à sua funcionalidade, interfaces, condições ambientais, requisitos legais, restrições operacionais e ambientais, perigos conhecidos entre outros. Essa definição serve para fornecer informação suficiente sobre o item a ser usada nas fases seguintes. Na sequência, é feita uma distinção entre o desenvolvimento de um novo item e a modificação de um já existente. Se for o caso de um novo desenvolvimento, o desenvolvimento continua com a análise de perigos e avaliação de riscos. Caso se trate da modificação de um item, antes de prosseguir com o

desenvolvimento é feita uma análise de impacto, a fim de identificar as áreas, condições e características afetadas pela modificação [1].

Após o início do ciclo de vida de segurança, é realizada a análise de perigos e avaliação de riscos, a qual identifica os riscos em potencial e estima a probabilidade de exposição, a controlabilidade e a severidade dos eventos perigosos causados por mau funcionamento de cada um dos itens, sendo que a junção desses parâmetros determina os ASILs dos eventos perigosos. Posteriormente, a análise de perigos e avaliação de riscos determina os objetivos de segurança para cada evento perigoso dos itens. Os objetivos de segurança não são expressos em termos de soluções tecnológicas, mas sim em termos de objetivos funcionais. Os ASILs que foram determinados para os eventos perigosos são atribuídos aos objetivos de segurança correspondentes [1].

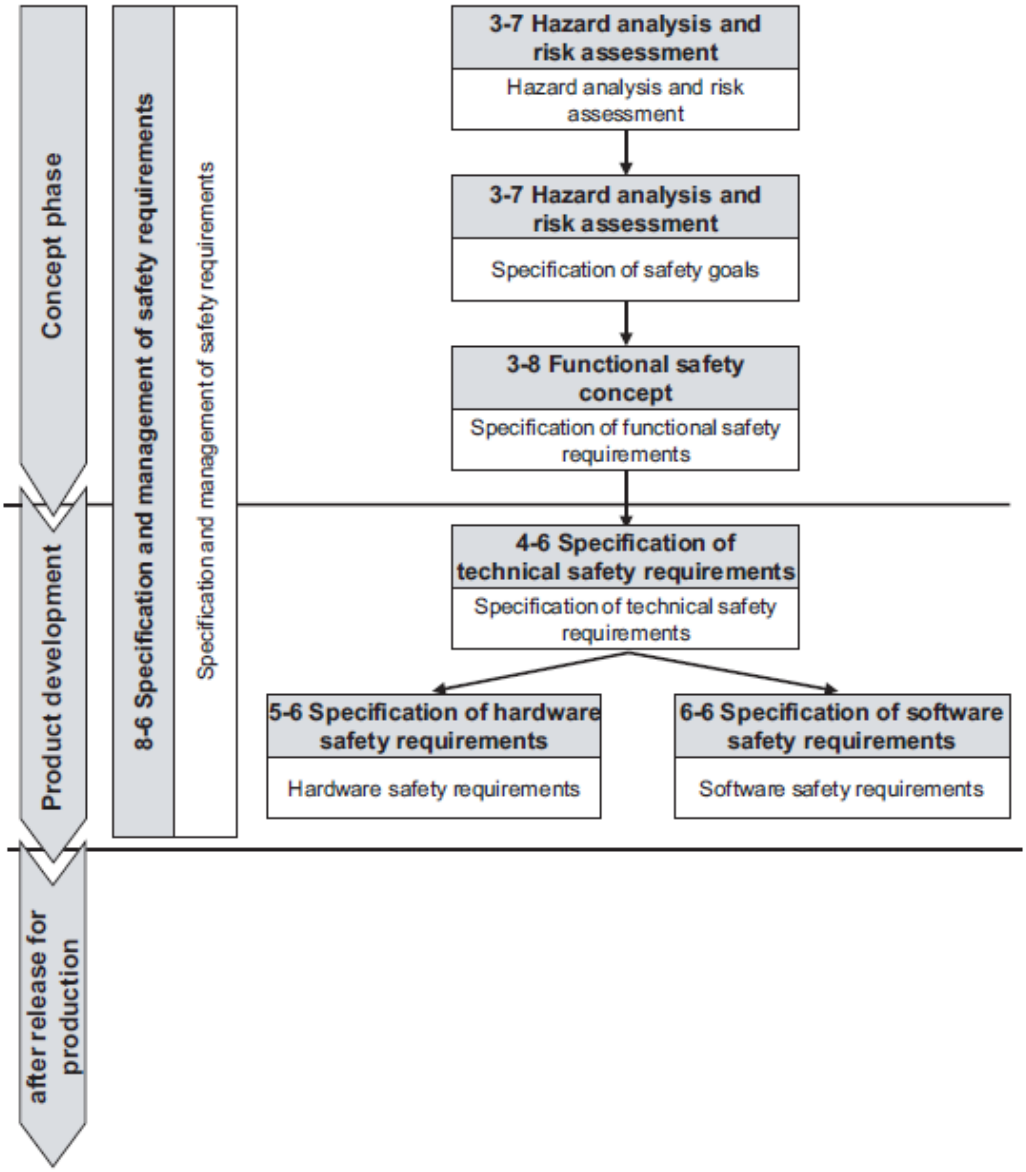
Durante as fases e subfases subsequentes, conforme será descrito a seguir, os requisitos de segurança detalhados são derivados dos objetivos de segurança e sucessivamente refinados, de requisitos de segurança funcional (requisitos de alto nível) em requisitos técnicos de segurança, e ainda mais adiante até os requisitos de baixo nível específicos para a implementação em soluções técnicas (requisitos de *hardware* e/ou *software*) conforme mostra a Figura 3. Esses requisitos de segurança herdam o ASIL dos objetivos de segurança correspondentes [1].

Baseado nos objetivos de segurança, primeiramente o conceito de segurança funcional é então especificado pelos requisitos de segurança funcional que são alocados aos elementos de um item. Após ter especificado o conceito de segurança funcional, inicia-se a fase de desenvolvimento e o item é desenvolvido a partir da perspectiva do nível do sistema. O processo de desenvolvimento do sistema baseia-se no conceito de modelo V com a especificação dos requisitos técnicos de segurança, a arquitetura do sistema, a concepção e implementação do sistema no ramo esquerdo e a integração, verificação, validação e avaliação de segurança funcional no ramo direito [1].

Baseado na especificação de projeto do sistema, o item é então desenvolvido a partir da perspectiva do nível de *hardware* e *software* que, assim como o desenvolvimento a nível de sistema, também seguem o conceito do modelo V. Assim, no processo de desenvolvimento do *hardware* e do *software*, a

especificação dos requisitos e o projeto e implementação estão no ramo esquerdo e a integração e teste e a verificação dos requisitos estão no ramo direito [1].

Figura 3 – Estrutura dos requisitos de segurança



Fonte: ISO 26262 (2011).

O planejamento para produção e operação e a especificação dos requisitos associados iniciam durante o desenvolvimento do produto a nível de sistema. Essa fase endereça os processos produtivos relevantes para os objetivos de segurança do item e o desenvolvimento e gerenciamento de instruções para manutenção, reparo e desativação do item, a fim de assegurar a segurança funcional após a liberação do item para a produção [1].

2.3 NÍVEIS DE INTEGRIDADE DE SEGURANÇA AUTOMOTIVA (ASIL)

O ASIL é um componente chave para o cumprimento da ISO 26262, sendo determinado já no começo do processo de desenvolvimento ao realizar a análise de perigos e avaliação de riscos, conforme citado anteriormente. O ASIL representa o grau de rigor que deve ser aplicado no desenvolvimento, implementação e verificação de um requisito, a fim de evitar um risco residual no produto final. Com isso, fornece orientações para escolher os métodos adequados para alcançar certo nível de integridade do produto, sendo que tais orientações são destinadas a complementar as práticas de segurança atuais. Assim, os automóveis atuais são fabricados em um nível de segurança alto e a ISO 26262 é feita para padronizar algumas práticas ao longo de toda indústria. A norma ISO 26262 especificamente identifica os requisitos mínimos de teste dependendo do ASIL do componente. Isso auxilia na determinação dos métodos que devem ser usados para o teste e as análises a serem feitas [4].

Baseada na definição dos itens, a análise de perigos e avaliação de riscos busca descrever primeiramente as situações e modos de operação em que um comportamento não intencional ou um mau funcionamento de cada item irá resultar em um evento perigoso, tanto quando o veículo é corretamente usado como também quando é utilizado incorretamente de forma previsível. Com isso, os perigos são então determinados sistematicamente em termos de condições ou comportamentos observados a nível de veículo e utilizando técnicas adequadas tais como *brainstorming*, *checklists*, histórico de qualidade, FMEA e estudos de campo [1].

Os eventos perigosos são definidos para combinações relevantes de situações operacionais e perigos, considerando que todos os outros sistemas além do item em análise estão funcionando corretamente. Além disso, eles focam no dano a cada pessoa potencialmente em risco, incluindo o motorista ou passageiros do veículo que causa o evento perigoso, como também ciclistas, pedestres ou ocupantes de outros veículos. Na etapa seguinte, todos os eventos perigosos que foram identificados são classificados em relação à severidade, exposição e controlabilidade. [1].

A classe de severidade é determinada com base na avaliação das possíveis lesões resultantes de um dano em relação ao motorista, aos passageiros, pessoas

ao redor do veículo ou indivíduos em veículos próximos. Estatísticas de acidentes podem ser usadas para determinar as lesões que podem ocorrer em diferentes tipos de acidentes. A severidade do dano em potencial é então estimada e atribuída a uma das classes de S0, S1, S2 ou S3, de acordo com a Tabela 1. A descrição da escala abreviada de lesões (AIS) pode ser usada para caracterizar a severidade. A classe de severidade S0 deve ser atribuída se a análise do perigo determina que as consequências são claramente limitadas ao dano material e não envolvem dano as pessoas, portanto a atribuição do ASIL não é exigida [1].

Tabela 1 – Classes de severidade

	Classes of severity			
	S0	**S1**	**S2**	**S3**
Description	No injuries	Light and moderate injuries	Severe and life-threatening injuries (survival probable)	Life-threatening injuries (survival uncertain), fatal injuries
Reference for single injuries (from AIS scale)	AIS 0 and less than 10 % probability of AIS 1-6 Damage that cannot be classified safety-related	More than 10 % probability of AIS 1-6 (and not S2 or S3)	More than 10 % probability of AIS 3-6 (and not S3)	More than 10 % probability of AIS 5-6

Fonte: ISO 26262 (2011).

A estimativa da probabilidade de exposição requer a avaliação dos cenários em que estão presentes determinados fatores relevantes que contribuem para a ocorrência do perigo. Os cenários a serem avaliados incluem uma ampla gama de situações de direção ou operação. Essas avaliações resultam na classificação da probabilidade de exposição de cada situação operacional, a qual é atribuída a uma das classes de E0, E1, E2, E3 ou E4, de acordo com a Tabela 2. A avaliação da probabilidade de exposição é feita assumindo que cada veículo está equipado com o item, de forma que então não é válido reduzir a probabilidade de exposição porque o item não está presente em todos os veículos. A classe E0 deve ser usada para as situações consideradas como extremamente incomuns (como, por exemplo, desastres naturais) e, portanto, a atribuição do ASIL não é exigida [1].

Tabela 2 – Classes de probabilidade de exposição

	Classes of probability of exposure				
	E0	**E1**	**E2**	**E3**	**E4**
Description	Incredible	Very low probability	Low probability	Medium probability	High probability
Duration (% of average operating time)		Not specified	<1 % of average operating time	1 % to 10 % of average operating time	>10 % of average operating time
Frequency of situation		Occurs less often than once a year for the great majority of drivers	Occurs a few times a year for the great majority of drivers	Occurs once a month or more often for an average driver	Occurs during almost every drive on average

Fonte: ISO 26262 (2011).

A controlabilidade do evento perigoso é atribuída a uma das classes de C0, C1, C2 ou C3, de acordo com a Tabela 3. A avaliação da controlabilidade é uma estimativa da probabilidade de que o motorista ou outras pessoas potencialmente em risco possam obter o controle suficiente do evento perigoso, de modo que possam evitar o dano específico. Supõe-se que o motorista está em condições apropriadas de dirigir (por exemplo, não está cansado (a)), tem o treinamento adequado (possui carteira de motorista) e está cumprindo todos os regulamentos legais aplicáveis, incluindo os devidos requisitos de cuidados para evitar riscos aos outros participantes do tráfego [1].

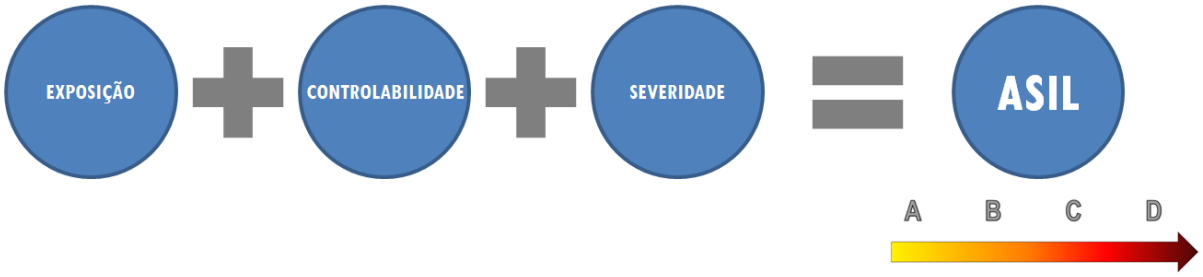
Tabela 3 – Classes de controlabilidade

	Classes of controllability			
	C0	**C1**	**C2**	**C3**
Description	Controllable in general	Simply controllable	Normally controllable	Difficult to control or uncontrollable
Driving factors and scenarios	Controllable in general	99 % or more of all drivers or other traffic participants are usually able to avoid harm	90 % or more of all drivers or other traffic participants are usually able to avoid harm	Less than 90 % of all drivers or other traffic participants are usually able, or barely able, to avoid harm

Fonte: ISO 26262 (2011).

Assim, a estimativa de um risco, baseada em uma combinação da probabilidade de exposição, da possível controlabilidade por um motorista e a possível severidade do resultado se ocorrer um evento crítico, leva ao ASIL (Figura 4). Através da combinação desses parâmetros de acordo com a Tabela 4, é atribuído um ASIL de A, B, C ou D, sendo D os processos de segurança mais críticos e as regulamentações de teste mais restritas [4].

Figura 4 – Determinação do ASIL



Fonte: National Instruments (2013). Disponível em: <http://www.ni.com/white-paper/13647/pt>. Acesso em: 20 dez. 2017.

Tabela 4 – Determinação do ASIL

Severity class	Probability class	Controllability class		
		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

Fonte: ISO 26262 (2011).

A classe QM indica que não existem requisitos de conformidade com a ISO 26262. Maus funcionamentos classificados como QM não são considerados como relevantes à segurança de acordo com a ISO 26262, o que não significa que o sistema como um todo não é relevante para a segurança. A classificação QM pressupõe que haja um sistema funcional QM em funcionamento implementado de acordo com a ISO/TS 16949, por exemplo, e que isso é suficiente para desenvolver tal sistema classificado como QM [3].

O ASIL não leva em conta as tecnologias usadas no sistema, ele é puramente focado no dano ao motorista e aos outros usuários da estrada. Além disso, o esquema da ISO 26262 para a classificação dos perigos reconhece que um perigo em um sistema automotivo não leva necessariamente a um acidente. O resultado dependerá se as pessoas em risco estão realmente expostas ao perigo da situação em que ele ocorre, e se eles são capazes de controlar o resultado do perigo [3][4].

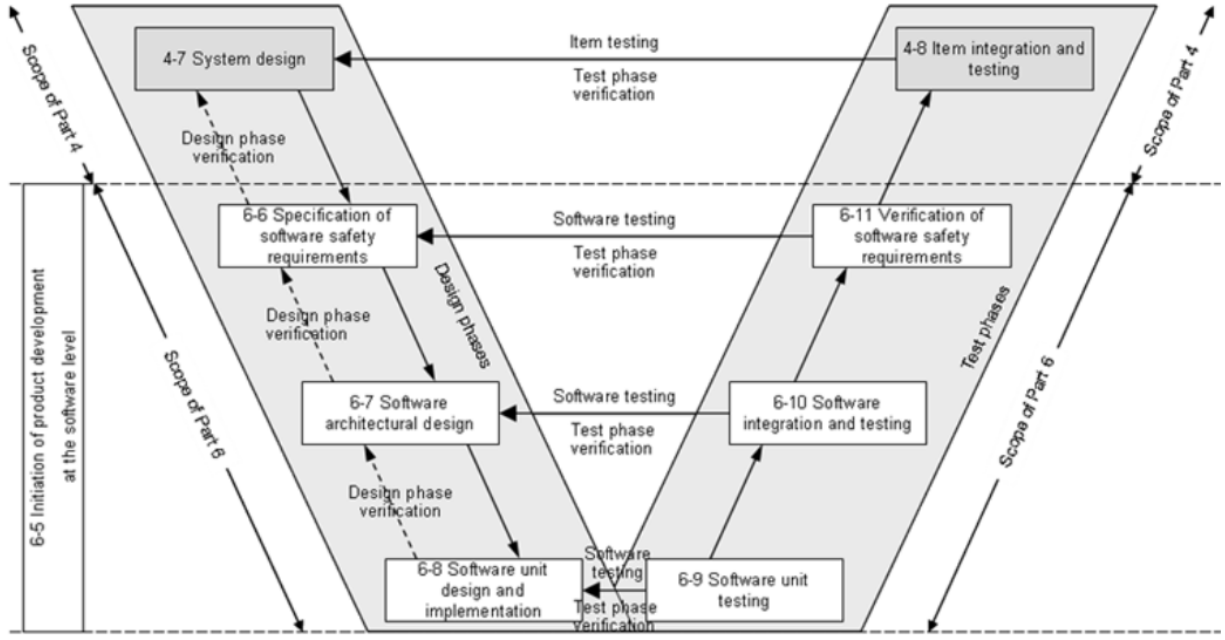
Uma vez que o ASIL foi determinado, um objetivo de segurança para o item é formulado, conforme mencionado anteriormente. Isso define o comportamento necessário do sistema para garantir a segurança, de modo que seja evitado um risco indesejável, e servirá de ponto de partida para o desenvolvimento futuro. [4].

2.4 SOFTWARE

O início do desenvolvimento de *software* é uma atividade de planejamento, em que as subfases do desenvolvimento e seus processos de suporte são determinados e planejados de acordo com a extensão e complexidade do desenvolvimento do item. Para cada subfase e processo são determinados também os métodos e ferramentas apropriados para cumprir os requisitos e seus respectivos ASILs. O planejamento do desenvolvimento de *software* inclui a coordenação com o desenvolvimento de produto a nível de sistema e a nível de *hardware* [1].

As fases do desenvolvimento de produto a nível de *software*, o qual baseia-se no modelo V, são mostradas na Figura 5 [1].

Figura 5 – Etapas do desenvolvimento de *software*



Fonte: ISO 26262 (2011).

Primeiramente são especificados os requisitos de segurança do *software*, derivados dos requisitos técnicos de segurança, e com base no conceito de segurança técnico e na especificação do projeto do sistema. Essa especificação leva em consideração as restrições do *hardware* e o impacto dessas restrições no *software*, configurações especificadas para o sistema e o *hardware*, restrições de tempo, interfaces externas, entre outros. Nessa subfase também é feito o detalhamento da especificação de interface entre *hardware* e *software*, descrevendo cada dependência entre *hardware* e *software* relacionada à segurança [1].

O objetivo da subfase seguinte é desenvolver e verificar o projeto de arquitetura de *software* que atenda aos requisitos de segurança de *software*. São também implementados os requisitos que não são relacionados à segurança em um mesmo processo de desenvolvimento. O projeto da arquitetura do *software* representa todos os componentes de *software* e suas interações em uma estrutura hierárquica, sendo descritos os aspectos estáticos (tipos de dados e suas características, sequência lógica do processamento de dados, interfaces e dependências externas etc) e os dinâmicos (fluxo de dados, fluxo de controle dos processos, restrições temporais, interrupções etc) [1].

Nessa etapa, são especificados mecanismos de segurança necessários para evitar as falhas de *software*. Para isso, são aplicados mecanismos de detecção de erros tais como checagem dos intervalos de dados de entrada e saída, checagem de plausibilidade, detecção de erros nos dados, monitoramento do fluxo de controle entre outros. Outro ponto importante é que se o *software* embarcado tem que implementar componentes de *software* de diferentes ASILs e também aqueles não relacionados à segurança, então todo o *software* embarcado deve ser tratado de acordo com o ASIL mais alto [1].

A próxima subfase no processo de desenvolvimento de *software* trata de fazer o desenvolvimento detalhado das unidades de *software*, com base no projeto da arquitetura do *software* e nos requisitos de *software* associados, relacionados à segurança ou não. O projeto detalhado será implementado como um modelo ou diretamente em forma de código-fonte, de acordo com as respectivas instruções de modelamento ou codificação aplicáveis [1].

2.5 TESTE DE SOFTWARE

O aumento cada vez maior da complexidade dos *softwares* embarcados e, consequentemente, da probabilidade de falhas, está tornando a atividade de verificação de *software* para detecção de falhas ainda mais difícil e demorada. Assim, durante o desenvolvimento da ISO 26262, o teste torna-se um componente crítico. Os componentes de segurança críticos devem responder apropriadamente a cenários de teste e estar dentro de limites de segurança especificados quando expostos a várias entradas humanas e ambientais. O uso de sistemas de teste de alta qualidade pode melhorar o desempenho de um produto, aumentar a qualidade e confiabilidade e diminuir as taxas de devolução. É estimado que o custo de uma falha diminui em 10 vezes quando o erro é pego na produção em vez do campo e diminui 10 vezes novamente se for pego no projeto em vez da produção. Detectando esses defeitos e coletando dados para melhorar um projeto ou processo, o teste traz valor à organização. Conduzir a inovação neste processo por meio da inserção de tecnologia e metodologias baseadas nas melhores práticas pode gerar grandes ganhos de eficiência e reduções de custos [4].

2.5.1 Teste de unidade de software

O objetivo dessa subfase é demonstrar que as unidades de *software* estão em conformidade com as especificações do projeto, especificações funcionais e de interface *hardware-software*, não contêm funcionalidades indesejadas, apresentam projeto robusto (com ausência de qualquer tipo de inacessibilidade, detecção de erros efetiva, mecanismos de tratamento de erros) e possuem os recursos suficientes para suportar suas funcionalidades [1].

Para o desenvolvimento baseado em modelo, o teste de unidade de *software* pode ser executado no nível de modelo, seguido de testes de comparação entre o modelo e o código, a fim de garantir que o comportamento dos modelos é equivalente ao código gerado automaticamente [1].

No Anexo A são apresentados os métodos recomendados pela ISO 26262 para teste de unidade de *software*, para especificação de sequências de teste e para avaliar a cobertura de requisitos pelas sequências de teste. Para cada método, o grau de recomendação para cada ASIL é representado da seguinte forma: “++” indica que o método é altamente recomendado, “+” indica que o método é recomendado e “o” indica que o método não tem recomendação a favor ou contra o seu uso [1].

2.5.2 Teste de integração de software

Nessa subfase, os níveis de integração e as interfaces entre os elementos de *software* são testados em relação ao projeto da arquitetura do *software*, verificando-se assim que o *software* está em conformidade com as funcionalidades especificadas [1].

O ambiente de teste, tanto para o teste de integração de *software* como também para o teste de unidade de *software*, deve corresponder o mais próximo possível ao ambiente de destino. Se o teste não for realizado no ambiente de destino, as diferenças em relação ao ambiente de teste devem ser analisadas a fim de especificar testes adicionais no ambiente de destino durante as fases de teste subsequentes. Além disso, dependendo do escopo, o ambiente de teste mais apropriado deve ser escolhido para a execução dos testes, tais como *Model-in-the-*

loop (MIL), *Software-in-the-loop* (SIL), *Processor-in-the-loop* (PIL) e *Hardware-in-the-loop* (HIL) [1].

A simulação em HIL é muito utilizada no desenvolvimento automotivo e recomendada pela ISO 26262 para ser usada em todas as fases de desenvolvimento de *software*. Ela se aplica aos vários sistemas do veículo e também aos vários cenários no processo de desenvolvimento tais como testes funcionais (como por exemplo, validação de estratégias de controle), testes de ECUs unitárias (os quais envolvem testes de integração, de aceitação e de liberação de *software*, de funções de diagnóstico, de funcionalidade geral da ECU) e testes em uma rede de ECUs (testes de gerenciamento da rede, comportamento do barramento, interação entre ECUs e funções distribuídas). A simulação em HIL permite testar a ECU em um ambiente de teste realístico, sendo que processos estáticos e dinâmicos, incluindo a comunicação, são simulados em tempo real, formando um circuito fechado entre o objeto sendo testado e o ambiente de simulação. Os testes em HIL podem ser realizados até em estágios de desenvolvimento iniciais, sem comprometer o sistema real ou colocar as pessoas em risco [9].

No Anexo B são apresentados os métodos recomendados pela ISO 26262 para teste de integração de *software*, para especificação de sequências de teste e para avaliar a cobertura de requisitos pelas sequências de teste [1].

Conforme mostrado nos Anexos A e B, os métodos de teste de unidade de *software* e de integração de *software* recomendados pela ISO 26262 são: teste baseado em requisitos, teste de interface, teste de injeção de falha, teste de uso de recursos e teste de comparação *back-to-back*.

2.5.3 Teste de injeção de falha

A ISO 26262 recomenda o uso da técnica de injeção de falha como método de teste no desenvolvimento de sistemas, *hardware* e *software* automotivo. No nível do *software*, a injeção de falha é essencial para teste de unidade de *software* e teste de integração de *software*, sendo altamente recomendado para atender aos ASILs mais críticos C e D [1].

Esse método é frequentemente utilizado para aprimorar a cobertura de teste dos requisitos de segurança, pois durante a operação normal os mecanismos de segurança não são chamados. Além disso, também é utilizado para analisar o comportamento do sistema na presença de falhas e avaliar os efeitos das mesmas [9][12].

Ao desenvolver sistemas tolerantes a falhas em configurações críticas de segurança, a injeção de falhas também tem como objetivo testar se os mecanismos tolerantes a falhas e mecanismos de segurança estão implementados corretamente e são eficientes o suficiente a fim de manter o sistema de acordo com os objetivos de segurança esperados, mesmo que o desempenho funcional seja um pouco diferente do sistema original. Assim, é verificado que os requisitos de segurança não são violados e é determinada a cobertura da detecção de erros e dos mecanismos de recuperação que foram desenvolvidos [10][12].

Um teste de injeção de falha introduz falhas em um item através do *software* em si ou utilizando meios específicos como, por exemplo, uma interface especial de teste, elementos preparados especificamente para esse fim ou certos equipamentos de comunicação. Diferentes tipos de falhas podem ser injetados como, por exemplo, valores e variáveis corrompidos, falhas de memória e registro, condições de erro e sinalizadores, temporizações irregulares, mensagens faltantes, memória corrompida, entre outros [9][12].

Os principais elementos que caracterizam o teste de injeção de falha são: o conjunto de falhas a serem injetadas (o modelo de falha), as atividades do sistema sob as quais as falhas são injetadas (a ativação), as leituras dos resultados do experimento e as medidas avaliadas com base nas leituras. Esses elementos formam a base do modelo FARM (“*Fault model – Activation – Readouts – Measures*”) para injeção de falha encontrado na literatura [13].

Testar sistemas ou componentes para todas as falhas possíveis não é viável, portanto, um modelo de falha restrito e adequado deve ser selecionado. O modelo de falha deve ser baseado no conhecimento profundo do domínio, nos sistemas e na maneira como o sistema interage com o ambiente. Modelos de falha apropriados podem ser baseados em falhas conhecidas, riscos identificados e eventos temidos, bem como requisitos específicos (por exemplo, requisitos não funcionais, requisitos de segurança e confiabilidade) e possíveis defeitos físicos ou

erros operacionais. É então muito importante selecionar e compor os modelos de falha e estes devem desencadear problemas reais que são reconhecidos como tal. Os modelos de falhas também devem ser adaptados às interfaces disponíveis e aos recursos de monitoramento e controle [11].

Um modelo de falha descreve o escopo das falhas consideradas nos experimentos. Esses modelos são a representação de falhas reais e são geralmente limitados pela capacidade da ferramenta em reproduzi-los ou em simular seus efeitos. Modelos de falhas em *softwares* geralmente descrevem erros reais implementados pelos desenvolvedores. Tais modelos podem descrever defeitos comuns ou a manifestação de tais defeitos durante a execução do programa [11].

Ter um modelo realista de falhas é um dos maiores desafios. O problema é que, atualmente, com sistemas mais críticos e mais complexos, bem como com a eficiência das ferramentas de testes automatizados, os modelos de falhas precisam ser constantemente atualizados e adaptados às evoluções tecnológicas (tanto de *hardware* quanto de *software*) [11].

Mutação é uma técnica bastante conhecida para a injeção de falhas em *softwares*, em que são introduzidas pequenas alterações no código-fonte, como sugerido pelos modelos de falhas considerados. O programa alterado resultante é chamado de mutante. A aplicação dessa técnica requer uma biblioteca de falhas a ser considerada e algum mecanismo para injetar as falhas [10].

3 CONSIDERAÇÕES FINAIS

Os avanços tecnológicos implementados em sistemas embarcados automotivos nos últimos anos trouxeram muitos benefícios para a sociedade. Aliado a isso, a complexidade tecnológica dos sistemas também cresceu, tornando a criticidade das aplicações e o risco de falhas cada vez maior. Assim, além de fornecer sistemas veiculares mais eficientes, a preocupação em fornecer sistemas mais seguros também deve fazer parte do desenvolvimento automotivo atual.

A norma ISO 26262 fornece uma série de recomendações e diretrizes a serem aplicadas em novos projetos na indústria automotiva, voltadas à segurança funcional dos sistemas eletro-eletrônicos. Essas recomendações abrangem todo o ciclo de vida dos produtos, desde o conceito até a produção e desativação, o que significa que todos os envolvidos nos processos de desenvolvimento de produto e na cadeia produtiva devem conhecer e aplicar os conceitos e procedimentos estabelecidos pela norma em seus processos atuais.

Assim, sugere-se que em trabalhos futuros sejam realizados estudos detalhados a respeito dos métodos e técnicas propostos pela norma a serem aplicados nos processos de desenvolvimento tanto de *hardware* como também de *software* embarcado. Tais estudos deveriam abordar tanto uma explanação detalhada do que significam e como funcionam os métodos, como também a análise da aplicação em casos reais a fim de aprimorar o entendimento.

REFERÊNCIAS

[1] International Organization for Standardization. **ISO 26262: Road Vehicles – Functional Safety**. 2011.

[2] KAFKA, Peter. **The Automotive Standard ISO 26262, the innovative driver for enhanced safety assessment & technology for motor cars**. In: INTERNATIONAL SYMPOSIUM ON SAFETY SCIENCE AND TECHNOLOGY, 2012, Austria.

[3] CROLLA, David. **Encyclopedia of Automotive Engineering**. Wiley, 2015.

[4] NATIONAL INSTRUMENTS. **A evolução dos sistemas eletrônicos embarcados em veículos e os desafios para garantir a segurança**. 2013. Disponível em: <http://www.ni.com/white-paper/13647/pt>. Acesso em: 20 dez. 2017.

[5] ISMAIL, Azianti; JUNG, Won. **Research Trends in Automotive Functional Safety**. In: INTERNATIONAL CONFERENCE ON QUALITY, RELIABILITY, RISK, MAINTENANCE AND SAFETY ENGINEERING, 2013.

[6] BRICIU, Catalin-Virgil; FILIP, Ioan. **The Challenge of Safety and Security in Automotive Systems**. In: IEEE INTERNATIONAL SYMPOSIUM ON APPLIED COMPUTATIONAL INTELLIGENCE AND INFORMATICS, 9., 2014, Timişoara, Romania.

[7] WINNER, Hermann; MAURER, Markus. **Automotive Systems Engineering**. Springer, 2013.

[8] SPORER, Harald; MACHER, Georg; ARMENGAUD, Eric; KREINER, Christian. **Incorporation of Model-based System and Software Development Environments**. In: EUROMICRO CONFERENCE ON SOFTWARE ENGINEERING AND ADVANCED APPLICATIONS, 41., 2015.

[9] HIMMLER, Andreas; LAMBERG, Klaus; BEINE, Michael. **Hardware-in-the-Loop Testing in the Context of ISO 26262**. In: SAE INTERNATIONAL, 2012.

[10] PILL, Ingo; RUBIL, Ivan; WOTAWA, Franz; NICA, Mihai. **SIMULTATE: A Toolset for Fault Injection and Mutation Testing of Simulink Models**. In: IEEE INTERNATIONAL CONFERENCE ON SOFTWARE TESTING, VERIFICATION AND VALIDATION WORKSHOPS, 9., 2016.

[11] SILVA, Nuno; BARBOSA, Ricardo; CUNHA, João Carlos; VIEIRA, Marco. **A View on the Past and Future of Fault Injection**. In: IEEE, 2013.

[12] RANA, Rakesh et al. **Improving Fault Injection in Automotive Model Based Development using Fault Bypass Modeling**.

[13] PINTARD, Ludovic; LEEMAN, Michel; YMLAHI-OUAZZANI, Abdelillah; FABRE, Jean-Charles et al. **Using Fault Injection to Verify an AUTOSAR Application According to the ISO 26262**. In: SAE INTERNATIONAL, 2015.

ANEXOS

Anexo A – Métodos para teste de unidade de *software*

Table 10 — Methods for software unit testing

Methods		ASIL			
		A	B	C	D
1a	Requirements-based test[a]	++	++	++	++
1b	Interface test	++	++	++	++
1c	Fault injection test[b]	+	+	+	++
1d	Resource usage test[c]	+	+	+	++
1e	Back-to-back comparison test between model and code, if applicable[d]	+	+	++	++

[a] The software requirements at the unit level are the basis for this requirements-based test.

[b] This includes injection of arbitrary faults (e.g. by corrupting values of variables, by introducing code mutations, or by corrupting values of CPU registers).

[c] Some aspects of the resource usage test can only be evaluated properly when the software unit tests are executed on the target hardware or if the emulator for the target processor supports resource usage tests.

[d] This method requires a model that can simulate the functionality of the software units. Here, the model and code are stimulated in the same way and results compared with each other.

Table 11 — Methods for deriving test cases for software unit testing

Methods		ASIL			
		A	B	C	D
1a	Analysis of requirements	++	++	++	++
1b	Generation and analysis of equivalence classes[a]	+	++	++	++
1c	Analysis of boundary values[b]	+	++	++	++
1d	Error guessing[c]	+	+	+	+

[a] Equivalence classes can be identified based on the division of inputs and outputs, such that a representative test value can be selected for each class.

[b] This method applies to interfaces, values approaching and crossing the boundaries and out of range values.

[c] Error guessing tests can be based on data collected through a “lessons learned” process and expert judgment.

Table 12 — Structural coverage metrics at the software unit level

Methods		ASIL			
		A	B	C	D
1a	Statement coverage	++	++	+	+
1b	Branch coverage	+	++	++	++
1c	MC/DC (Modified Condition/Decision Coverage)	+	+	+	++

Anexo B – Métodos para teste de integração de *software*

Table 13 — Methods for software integration testing

Methods		ASIL			
		A	B	C	D
1a	Requirements-based test[a]	++	++	++	++
1b	Interface test	++	++	++	++
1c	Fault injection test[b]	+	+	++	++
1d	Resource usage test[cd]	+	+	+	++
1e	Back-to-back comparison test between model and code, if applicable[e]	+	+	++	++

[a] The software requirements at the architectural level are the basis for this requirements-based test.

[b] This includes injection of arbitrary faults in order to test safety mechanisms (e.g. by corrupting software or hardware components).

[c] To ensure the fulfilment of requirements influenced by the hardware architectural design with sufficient tolerance, properties such as average and maximum processor performance, minimum or maximum execution times, storage usage (e.g. RAM for stack and heap, ROM for program and data) and the bandwidth of communication links (e.g. data buses) have to be determined.

[d] Some aspects of the resource usage test can only be evaluated properly when the software integration tests are executed on the target hardware or if the emulator for the target processor supports resource usage tests.

[e] This method requires a model that can simulate the functionality of the software components. Here, the model and code are stimulated in the same way and results compared with each other.

Table 14 — Methods for deriving test cases for software integration testing

Methods		ASIL			
		A	B	C	D
1a	Analysis of requirements	++	++	++	++
1b	Generation and analysis of equivalence classes[a]	+	++	++	++
1c	Analysis of boundary values[b]	+	++	++	++
1d	Error guessing[c]	+	+	+	+

[a] Equivalence classes can be identified based on the division of inputs and outputs, such that a representative test value can be selected for each class.

[b] This method applies to parameters or variables, values approaching and crossing the boundaries and out of range values.

[c] Error guessing tests can be based on data collected through a “lessons learned” process and expert judgment.

Table 15 — Structural coverage metrics at the software architectural level

Methods		ASIL			
		A	B	C	D
1a	Function coverage[a]	+	+	++	++
1b	Call coverage[b]	+	+	++	++

[a] Method 1a refers to the percentage of executed software functions. This evidence can be achieved by an appropriate software integration strategy.

[b] Method 1b refers to the percentage of executed software function calls.