KLEBER NOGUEIRA HODEL

**Planejamento e Estruturação de Testes de Software em Sistemas
Eletrônicos Embarcados Automotivos**

São Paulo

2018

KLEBER NOGUEIRA HODEL

**Planejamento e Estruturação de Testes de Software em Sistemas
Eletrônicos Embarcados Automotivos**

Versão Corrigida

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção do título de
Doutor em ciências.

Área de Concentração: Engenharia Elétrica –
Sistemas Eletrônicos

Orientador: Prof. Dr. João Francisco Justo Filho

São Paulo

2018

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, _____ de _____ de _____

Assinatura do autor: _____

Assinatura do orientador: _____

Catálogo-na-publicação

Hodel, Kleber Nogueira

Planejamento e Estruturação de Testes de Software em Sistemas Eletrônicos Embarcados Automotivos / K. N. Hodel -- versão corr. -- São Paulo, 2018.

150 p.

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos.

1.Sistemas embutidos 2.Indústria automobilística 3.Redes complexas 4.Softwares I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Sistemas Eletrônicos II.t.

AGRADECIMENTOS

Antes de tudo, sou grato a Deus, por ajudar e manter-me firme neste propósito que é a idealização de mais este trabalho na minha carreira, no qual mais uma vez superei grandes desafios para a elaboração desta tese.

Agradeço imensamente a minha querida esposa Tamiris que esteve o tempo todo ao meu lado, incondicionalmente, com muito amor e paciência. Nos momentos mais difíceis, que não foram raros neste último ano, sempre me fez acreditar que chegaria ao final deste difícil, porém gratificante, trabalho. Também quero agradecer ao nosso filho Noah, que apareceu na fase final desse trabalho para motivar e trazer muito mais sentido para tudo o que estou fazendo, mesmo sem ainda ter nascido me trouxe muita alegria e motivação.

Se estou concluindo esta tese é porque um dia meus Pais, Celeste e Alberto, me orientaram com maestria desde o meu nascimento, trazendo muito protagonismo a minha vida e ao mesmo tempo ensinando a ser um ser humano cada dia melhor. Deixando de lado seus sonhos para transformar os sonhos dos filhos em realidade, tudo que consegui só foi possível graças ao amor, apoio e dedicação que eles sempre tiveram por mim. Sempre me ensinaram agir com respeito, simplicidade, dignidade, honestidade e amor ao próximo.

Ao Prof. Dr. João Francisco Justo Filho, por ter-me dado a oportunidade de desenvolver este trabalho, além de sua orientação, total apoio e disponibilidade, pelo saber que transmitiu, pelas opiniões e críticas, total colaboração no solucionar de dúvidas e problemas que surgiram ao longo da realização deste trabalho e por todas as palavras de incentivo.

Aos Professores e Doutores com quem estudei durante o período de desenvolvimento da pesquisa, especialmente ao Prof. Dr. Armando Antônio Maria Laganá e Prof. Dr. Max Mauro Dias Santos, que sempre me incentivou e motivou a concluir esse trabalho e ao Prof. Dr. José Reinaldo Silva contribuiu com recomendações para definição da tese.

Aos muitos colegas da Mercedes Benz do Brasil que direta e indiretamente me ajudaram a ser o profissional e aluno que sou, contribuindo com conhecimento, amizade, apoio, consultorias, entre outras coisas, em especial ao Dr. Daniel Cardia e ao Dr. Dirceu Fernandes.

*“Confia no Senhor de todo o teu coração, e não te
estribes no teu próprio entendimento”.
Provérbios 3.5*

RESUMO

A indústria automotiva vem mudando seu perfil drasticamente na última década, tornando-se mais intensiva em sistemas computacionais, e consequentemente em *software*, do que em sistemas mecânicos. A maioria das funcionalidades dos veículos modernos é controlada por *software* distribuído, dentro de muitas Unidades de Controle Eletrônicas (ECUs) interconectadas, de modo que o veículo pode ser considerado como um conjunto de sistemas complexos. Como resultado da integração, os testes para as funções de *software* distribuído se tornaram um grande desafio. Muitas estratégias têm sido propostas para organizar o multinível de teste de *software* em sistemas embarcados automotivos, a fim de reduzir custos e melhorar sua eficácia e robustez. Esta investigação pretende estender este conceito de teste de *software* multinível baseado no modelo em V, projetando uma metodologia sistemática que mapeie cada função dentro do sistema embarcado do veículo, sem a existência de uma especificação detalhada de cada componente e possibilitando a definição de uma estratégia e um planejamento de teste de *software* antecipadamente. Esta metodologia foi totalmente desenvolvida incorporando as características da aplicação automotiva, incluindo os requisitos de segurança funcional ISO 26262. Desta forma e dentro de rigorosas condições de segurança, a nova metodologia melhora a eficiência dos processos de desenvolvimento de sistemas embarcados, possibilitando a antecipação do planejamento de teste de *software*. Sendo este, baseado em um mapeamento e codificação da função e das atividades de teste proposto nesta tese, que funciona também como base para futuros desenvolvimento de soluções de rastreabilidade do sistema embarcado.

ABSTRACT

The automotive industry has been changing its profile drastically in the last decade, becoming more intensive in computer systems, and consequently in software, than in mechanical systems. Most of the functionality of modern vehicles is controlled by distributed software, within many interconnected Electronic Control Units (ECUs), so that the vehicle can be considered as a set of complex systems. As a result of integration, testing for distributed software functions has become a major challenge. Many strategies have been proposed to organize the multilevel software testing in embedded automotive systems in order to reduce costs and improve their effectiveness and robustness. This research intends to extend this concept of multilevel software testing based on the V development model, designing a systematic methodology that maps each function within the embedded system of the vehicle, without the existence of a detailed specification of each component, allowing the definition of a strategy and software testing planning in advance. This methodology was fully developed incorporating the characteristics of the automotive application, including the ISO 26262 functional safety requirements. In this way, the new methodology improves the efficiency of the embedded systems development processes, within strict security conditions, allowing the anticipation of the planning based on a mapping and encoding of the function and test activities proposed in this thesis, which also serves as a basis for future development of embedded system traceability solutions.

LISTA DE ILUSTRAÇÕES

Figura 1 – Contexto da tese.	18
Figura 2 – Modelo <i>Stage Gate</i> proposto por Cooper em 2000.	22
Figura 3 – Modelo genérico para processo de desenvolvimento de produto.	23
Figura 4 – Modelo de processo de desenvolvimento com os <i>quality gates</i> definidos e linhas dos esforços para cada departamento em cada uma das etapas.	25
Figura 5 – Elementos para especificações	26
Figura 6 – Lista de funções.	31
Figura 7 – Exemplo da configuração das funções por veículos.	31
Figura 8 – Sistemas elétricos com suas respectivas funções.	32
Figura 9 – Representação de um conjunto de sistemas eletroeletrônicos de um veículo, com suas funções, componentes e subsistemas.	33
Figura 10 – Processo desenvolvimento de Engenharia de Sistemas – SIMILAR	35
Figura 11 – Exemplo de uma configuração de um veículo baseado em funções, sistemas e componentes.	45
Figura 12 – Cronograma de desenvolvimento de um componente eletrônico.	46
Figura 13 – Sequência de causas onde o <i>software</i> é uma potencial fonte de riscos.	54
Figura 14 – Níveis de testes do sistema eletroeletrônico de um veículo.	63
Figura 15 – Modelo V do sistema embarcado automotivo.	64
Figura 16 – Dimensões do teste de sistemas embarcados.	65
Figura 17 – Partes do nível de teste do componente do sistema.	70
Figura 18 – Ilustração da área de estudo da tese.	74
Figura 19 – Modelo V aplicado na Scania para desenvolvimento dos sistemas embarcados distribuídos	77
Figura 20 – Estruturação do teste de software em multinível.	78
Figura 21 – Metodologia proposta para Tese.	79
Figura 22 – Correlação de sistema, componente e função.	82
Figura 23 – Sequência de causas em que o <i>software</i> é uma fonte potencial de riscos.	83
Figura 24 – Mapeamento da função.	86

Figura 25 – Principais fatores que precisam ser avaliados em sistemas embarcados para definição da estratégia de testes.	88
Figura 26 – Estrutura do sistema eletrônico de veículos.	92
Figura 27 – Parâmetros para definição do plano e estratégia dos testes de multinível para aplicação no domínio da indústria automotiva.	98
Figura 28 – Fluxo de trabalho para verificar a relevância da segurança funcional. ..	99
Figura 29 – Fluxograma para definição da estratégia de teste para funções relevantes à segurança funcional.	100
Figura 30 – Fluxograma para definição da estratégia de teste para funções não relevantes à segurança funcional.	102
Figura 31 – Quantidade de testes por nível.	104
Figura 32 – Exemplo das interações entre sistemas e componentes de uma função.	106
Figura 33 – Ilustração da identificação da função em cada nível.	108
Figura 34 – Tipos de testes para os componentes e sistemas de um veículo em ordem cronológica.	110
Figura 35 – Estrutura de um caso de teste multinível.	111
Figura 36 – Abordagem do caso de teste para reutilizar e adaptar para outro nível de teste.	113
Figura 37 – Arquitetura da Função	116
Figura 38 – Fluxograma de definição dos níveis de teste da função ALC.	120
Figura 39 – Reutilização do <i>test case</i> no teste da função ALC.	124
Figura 40 – Adaptação do método proposto na tese para o domínio da indústria aeronáutica.	130

LISTA DE TABELAS

Tabela 1 – Comparativo da classificação do nível de segurança.....	58
Tabela 2 – Especificação da função veicular.	84
Tabela 3 – Modelo básico para especificação do sistema.	84
Tabela 4 – Mapeamento da função dentro do sistema do veículo.	86
Tabela 5 – Matriz dos sistemas, componentes e funções de um veículo.....	87
Tabela 6 – Requisitos de teste da ISO 26262 para funções e sistemas relevantes à segurança funcional.	90
Tabela 7 – Fatores de classificação de funções.....	97
Tabela 8 – Tabela de decisão dos níveis de Teste	103
Tabela 9 – Estrutura para identificação da função no nível de teste.	107
Tabela 10 – Modelo do código de cada atividade de teste.	109
Tabela 11 – Descrição das funções	115
Tabela 12 – Descrição dos sistemas que participam da função ALC.....	115
Tabela 13 – Contribuição da Função	117
Tabela 14 – Mapa da Função ALC no sistema do veículo.	118
Tabela 15 – Definição dos parâmetros da função	118
Tabela 16 – Níveis de teste requeridos para a função ALC.	121
Tabela 17 – Níveis de teste requeridos para a função ALC antes da reutilização do TCC.....	122
Tabela 18 – <i>Test Case</i> para a função ALC no nível de veículo.....	123
Tabela 19 – Níveis de teste requeridos para a função ALC considerando a reutilização do TCC.....	125

LISTA DE ABREVIATURAS E SIGLAS

AC	Componente Auxiliar
ASIL	Níveis de integridade a segurança funcional
ASME	<i>American Society of Mechanical Engineers</i>
CAN	<i>Controller Area Network</i>
COTS	<i>Commercial Off the Shelf</i>
DGP	Desenvolvimento Global de Produtos
DP	Desenvolvimento de Produto
ECU	<i>Eletronic Control Unit</i>
EE	Elétrica e Eletrônico
FFBD	Fluxo Funcional de Diagrama de Blocos
HIL	<i>Hardware in the Loop</i>
IDP	<i>Integrated Product Development</i>
IEC	<i>International Electrotechnical Commission</i>
IO	<i>Input and Output</i>
ISO	<i>International Standard Organization</i>
ITA	<i>Input Test Adapter</i>
MC	Componente Principal
MDB	<i>Model Based Design</i>
OEM	<i>Original Equipment Manufacturer</i>
OTA	<i>Output Test Adapter</i>
PDP	Processo de Desenvolvimento de Produto
PG	<i>Phase Gate</i>
PPO	<i>Product and Process Organization</i>
PTA	<i>Parameter Test Adapter</i>
QFD	<i>Quality Function Deployment</i>
QG	<i>Quality Gate</i>
QM	<i>Quality Management</i>
SAE	<i>Society of Automotive Engineers</i>
SST	Software sob teste
ST	Sistema em Teste
SW	<i>Software</i>

TCC	<i>Test Case Core</i>
TNC	<i>Transnational Companies</i>
UML	<i>Unified Modeling Language</i>
IoT	<i>Internet of things</i>

SUMÁRIO

1	INTRODUÇÃO	12
1.1	MOTIVAÇÃO	15
1.2	OBJETIVOS.....	16
1.3	CONTEÚDO E ORGANIZAÇÃO	17
2	REVISÃO BIBLIOGRÁFICA	17
2.1	PROCESSOS DE DESENVOLVIMENTO DE PRODUTO	18
2.1.1	PROCESSO DE DESENVOLVIMENTO DE PRODUTO AUTOMOTIVO ...	22
2.2	PROCESSO DE DESENVOLVIMENTO DE SISTEMAS ELÉTRICO- ELETRÔNICOS AUTOMOTIVOS.....	30
2.2.1	LISTA DE FUNÇÕES	30
2.2.2	SISTEMA ELETRÔNICO AUTOMOTIVO	31
2.2.3	COMPONENTE ELETROELETRÔNICO AUTOMOTIVO	32
2.2.4	ARQUITETURA ELÉTRICA.....	33
2.2.5	MODELO DE REFERÊNCIA PARA DESENVOLVIMENTO DE SISTEMAS 34	
2.2.5.1	ENGENHARIA DE SISTEMAS.....	34
2.2.5.2	AUTOSAR.....	38
2.2.5.3	MODELO EM V	40
2.2.5.4	“DESIGN” BASEADO EM MODELOS	41
2.2.5.5	MODELO DE DESENVOLVIMENTO POR FUNÇÕES.....	42
2.3	DESAFIOS PARA DESENVOLVIMENTO DE SISTEMAS ELÉTRICO- ELETRÔNICOS AUTOMOTIVOS.....	43
2.3.1	NÚMERO DE VARIANTES E CONFIGURAÇÕES.....	44
2.3.2	DESENVOLVIMENTO GLOBALIZADO	47
2.3.3	HETEROGENEIDADES DE SOFTWARE.....	49
2.3.4	DISTRIBUIÇÃO DAS ATIVIDADES DE DESENVOLVIMENTO DE “SOFTWARE”	52
2.3.5	REQUERIMENTOS DE SEGURANÇA	53
2.3.5.1	SAFWARE AUTOMOTIVO	54
2.3.5.2	SEGURANÇA FUNCIONAL EM ELETRÔNICA AUTOMOTIVA	55

2.3.5.3 ASIL – NÍVEL DE INTEGRIDADE DA SEGURANÇA AUTOMOTIVA, PROCESSO DE AVALIAÇÃO.	56
2.3.6 GESTÃO DO CONHECIMENTO	59
2.3.7 INTEGRAÇÃO DOS SISTEMAS E SUBSISTEMAS	60
2.3.8 TEMPO DE DESENVOLVIMENTO	61
2.3.9 GERENCIAMENTO DOS TESTES PARA VALIDAÇÃO DO SISTEMA ELÉTRICO-ELETRÔNICO	62
2.4 TESTE DE SOFTWARE EM SISTEMAS EMBARCADOS AUTOMOTIVOS	63
2.4.1 DIMENSÕES DE SISTEMAS EMBARCADOS AUTOMOTIVOS	63
2.4.1.1 OBJETIVO DO TESTE	65
2.4.1.2 PLATAFORMAS DE TESTE	67
2.4.1.3 NÍVEIS DE TESTE OU ESCOPO DE TESTE	69
2.4.2 DESAFIOS PARA TESTES DE SOFTWARE NA INDÚSTRIA AUTOMOTIVA.....	70
2.4.3 TESTE DE <i>SOFTWARE</i> EM APLICAÇÕES AUTOMOTIVAS	74
2.4.4 RASTREABILIDADE DOS TESTES DE <i>SOFTWARE</i>	79
3 DESENVOLVIMENTO DA METODOLOGIA DE DEFINIÇÃO DA ESTRATÉGIA DE TESTES DE <i>SOFTWARE</i> NA INDÚSTRIA AUTOMOTIVA	81
3.1 ESPECIFICAÇÃO DA FUNÇÃO	83
3.2 ESPECIFICAÇÃO DO SISTEMA	84
3.3 MAPEAMENTO DOS SISTEMAS, COMPONENTES E FUNÇÕES.....	85
3.4 FATORES/ PARÂMETROS DO DOMÍNIO DA INDÚSTRIA AUTOMOTIVA	88
3.4.1 SEGURANÇA FUNCIONAL	89
3.4.2 RELEVÂNCIA DO CLIENTE.....	91
3.4.3 COMPLEXIDADE DO SISTEMA	92
3.4.4 INFLUÊNCIA DO MOTORISTA NA FUNCIONALIDADE DO VEÍCULO	94
3.4.5 INFLUÊNCIA DO AMBIENTE EXTERNO NA FUNCIONALIDADE DO VEÍCULO.....	94
3.4.6 RESUMO DAS PRINCIPAIS CARACTERÍSTICAS DO DOMÍNIO AUTOMOTIVO	96
3.5 METODOLOGIA PARA DEFINIÇÃO DA ESTRATÉGIA DE TESTES	97
3.5.1 ESTRATÉGIA DE TESTE PARA FUNÇÕES RELEVANTES À SEGURANÇA FUNCIONAL	99

3.5.2	ESTRATÉGIA DE TESTE PARA FUNÇÕES NÃO RELEVANTES À SEGURANÇA FUNCIONAL.....	101
3.5.3	CONSOLIDAÇÃO DA ESTRATÉGIA DE TESTES	102
3.6	RASTREABILIDADE DAS ATIVIDADES DE TESTE	104
3.6.1	IDENTIFICAÇÃO DAS ATIVIDADES DE TESTE	104
3.6.2	ALINHAMENTO DO <i>TEST CASE</i> ENTRE OS NÍVEIS	109
4	APLICAÇÃO DA METODOLOGIA DE ESTRUTURAÇÃO E PLANEJAMENTO DOS TESTES	114
4.1	ESPECIFICAÇÃO DA FUNÇÃO	114
4.2	ESPECIFICAÇÃO DO SISTEMA.....	115
4.3	DEFINIÇÃO DE ARQUITETURA DO SISTEMA E MAPEAMENTO DA FUNÇÃO DENTRO SISTEMA.....	116
4.4	PARÂMETROS DA APLICAÇÃO	118
4.5	DEFINIÇÃO DOS NÍVEIS DE TESTE	119
4.6	RASTREABILIDADE.....	121
4.6.1	IDENTIFICAÇÃO DAS ATIVIDADES DE TESTE DA FUNÇÃO ALC.....	121
4.6.2	ALINHAMENTO DO <i>TEST CASE</i> OU CASO DE TESTE EM TODOS OS NÍVEIS DE TESTE.....	122
4.7	ANÁLISE DA APLICAÇÃO DA METODOLOGIA DE ESTRUTURAÇÃO E PLANEJAMENTO DOS TESTES	125
5	CONCLUSÃO.....	126
5.1	ESTRUTURAÇÃO E PLANEJAMENTO DOS TESTE DE SOFTWARE	126
5.2	RASTREABILIDADE.....	128
5.3	TRABALHOS FUTUROS	129
	REFERÊNCIAS	131

1 INTRODUÇÃO

Mais de um século depois que Carl Benz, Wilhelm Maybach e Gottlieb Daimler desenvolveram e produziram seus primeiros veículos a motor, o conceito geral básico dos carros não mudou muito. Embora os componentes tenham sido consideravelmente otimizados desde então, os automóveis no século XXI ainda são movidos por motores de combustão, em alguns casos recentes movidos também por motores elétricos, que transmitem a potência propulsora à superfície da estrada através de caixas de câmbio, eixos de transmissão e rodas, que juntamente com as unidades amortecedores, permitem estabilidade e conforto (WEBER, 2009).

No entanto, o que mudou drasticamente são os processos envolvidos no desenvolvimento de veículos, principalmente pela introdução de sistemas eletrônicos distribuídos que comandam grande parte das funcionalidades do veículo. O processo de desenvolvimento de veículos tornou-se uma interação complexa de sub processos descentralizados, que são gerenciados em um nível relativamente alto dentro das empresas.

Os modelos de desenvolvimento utilizados nas empresas automotivas são reconhecidos por possuírem eficácia e robustez, resultado de muitos anos de experiência, sempre passando por revisões e adaptações para que as empresas se mantenham competitivas, acompanhando a evolução e as exigências do mercado.

Na atualidade, o desenvolvimento de produtos continua a ser um dos maiores custos e investimentos da indústria de OEM (*Original Equipment Manufacturer*) automotiva. É considerado um ponto importante não só por causa dos enormes recursos e custos envolvidos, mas também devido à grande aceitação dos produtos novos no mercado (TOWNSEND *et al.*, 2013).

A indústria automotiva modificou seus veículos tecnologicamente partindo de um conceito de veículo historicamente dominado pela engenharia mecânica, com alguns sistemas auxiliares de engenharia elétrica e eletrônica. Passando para um modelo onde os sistemas mecânicos são os itens auxiliares e o universo dos sistemas eletrônicos torna-se o foco principal do desenvolvimento de produto

(LARRUCEA *et al.*, 2017, DAJSUREN *et al.*, 2017), determinando os limites de evolução de indústria automotiva.

Mais recentemente, o foco passou mais ao desenvolvimento de *software* do que de *hardware* (WEBER, 2009; ABELEIN *et al.*, 2012, LARRUCEA *et al.*, 2017). Essa evolução tem sido turbinada no mercado por algumas soluções, tendo como tecnologias propulsoras soluções para veículos autônomos (BERTOZZI *et al.*, 2011; LAUGIER *et al.*, 2014; BROGGI *et al.*, 2010; BARTELS *et al.*, 2014), veículos elétricos (híbridos completos, híbridos plug-in, veículos elétricos a bateria (KAHN *et al.*, 2007; ROS *et al.*, 2009; LUTSEY *et al.*, 2012; THOMAS, 2009) e tecnologias de IoT (*Internet of things*).

A primeira geração de funções envolvendo *software* na indústria automotiva era estritamente individual, controlada por apenas um componente; funcionalmente isolada de outras funções controladas por *software* no veículo (BROY *et al.*, 2007, LARRUCEA *et al.*, 2017). Atualmente, funções processadas por *software* são generalizadas no sistema dos veículos e variam desde funções básicas, como o acionamento de uma lâmpada ou ler o sinal de um interruptor, até funcionalidades de segurança avançada como funções que controlam veículos autônomos (DAJSUREN *et al.*, 2017).

A evolução tecnológica impulsiona o aumento de funções dentro do veículo, tornando a arquitetura eletrônica muito complexa, distribuída entre várias Unidades de Controle Eletrônico (ECU) através de uma rede de comunicação de dados. Para trabalhar com essa grande quantidade de componentes, é necessário um número significativo de pessoas e exige muita interação entre elas, o que torna a complexidade geral do processo de desenvolvimento do produto maior.

Nas últimas décadas, a quantidade de *software* em veículos evoluiu de zero a centenas de milhões em linhas de código, e esse crescimento deverá continuar no futuro (PRETSCHNER *et al.*, 2007; BROY *et al.*, 2007; MARO *et al.*, 2018). Isto é evidenciado por veículos de luxo, que contêm atualmente cerca de 300 funções de usuário, compostas por até 2500 funções de unidades de *software* individuais e implantadas em mais de 70 plataformas embutidas (PRETSCHNER *et al.*, 2007). A razão para este rápido desenvolvimento é impulsionada pela grande disponibilidade

de recursos tecnológicos, com custo muito atrativo (BROY *et al.*, 2007). Além disso, toda essa complexidade não pode comprometer a segurança do veículo, o que é regulamentado pela introdução da norma ISO 26262 (NÖRENBORG *et al.*, 2010, MAUBORGNE, *et al.*, 2016), com o objetivo de estabelecer um padrão de desenvolvimento de segurança unificado para todos os sistemas elétricos e eletrônicos automotivos.

Devido ao aumento da complexidade, a engenharia de *software* automotivo atraiu atenção tanto de estudiosos quanto de profissionais, e houve progressos significativos, especialmente nas duas últimas décadas em pesquisas com relevância industrial, mas é relativamente baixo o seu rigor científico (HAGHIGHATKHAH *et al.*, 2017), dificultando o avanço rápido e profundo desse conhecimento. No entanto, são necessárias mais contribuições analíticas e teóricas para avançar a pesquisa e apoiar os projetos de sistemas automotivos, que cada vez se tornam mais complexos (HAGHIGHATKHAH *et al.*, 2017).

Dentro desse contexto, diversos estudos apontam que o teste constitui mais da metade dos custos globais do desenvolvimento de *software* (HARROLD, 2000) e que, com a crescente complexidade do *software*, a proporção do custo do teste provavelmente continuará aumentando, a menos que sejam encontrados métodos de teste mais eficientes (AFZAL *et al.*, 2016).

Existem poucas investigações focadas em sistemas embarcados no domínio de aplicações da indústria (RANA *et al.* 2014). Estas geralmente estão orientadas à redução do tempo, ciclos de testes, diminuição dos custos de desenvolvimento e, ao mesmo tempo, aumento da qualidade do *software* final, que na maioria dos casos está na melhoria do seu processo de testes (COLLOFELLO *et al.*, 1996). O ciclo completo de testes do sistema eletrônico do veículo possui vários níveis e o grande desafio é identificar uma estratégia de teste na qual grande parte das falhas sejam detectadas no início do projeto (BRIGMANN *et al.*, 2008).

O foco deste trabalho é estudar e desenvolver uma metodologia para planejar e estruturar testes de *software* de sistemas eletrônicos embarcados de forma sistemática dentro do contexto automotivo, estendendo o conceito de teste multinível. Para isso, serão analisadas as características da aplicação e mapeadas

todas as etapas de teste, facilitando, assim, uma característica fundamental dos processos automotivos, a rastreabilidade geral do desenvolvimento das funções do veículo. Adicionalmente, a metodologia inclui os requisitos de segurança funcional, descritos e regulamentados pela ISO 26262 (ISO, 2011).

1.1 MOTIVAÇÃO

A indústria automotiva possui um complexo processo de desenvolvimento de produtos. Nos últimos vinte anos, tem aumentado tremendamente a aplicação de *software* nos veículos (BROY, 2006), com tendências tecnológicas desafiadoras, como, por exemplo, veículos autônomos, onde *software*, além de complexo, controla sistemas críticos de segurança (MARO *et al.*, 2018, KIM *et al.*, 2017, MAUBORGNE, *et al.*, 2016). Dentro desse contexto, a atividade de teste representa uma das mais importantes e custosas partes do desenvolvimento dos veículos. Pois, devido ao elevado risco de acidentes, um alto nível de maturidade e validação é exigido antes do produto chegar ao mercado. Embora isso tenha garantido o desenvolvimento de veículos com alto nível de qualidade e confiabilidade, com o aumento da introdução de novas tecnologias, surgiram desafios adicionais que não são tradicionais na indústria automotiva. Dentre eles a validação de *software* em um ambiente complexo de desenvolvimento de produto (MAUBORGNE, *et al.*, 2016, KIM *et al.*, 2017, PENDLETON *et al.*, 2017).

A literatura possui diversas abordagens relacionadas aos desafios das tecnologias de *software* na indústria automotiva (GRIMM *et al.*, 2003, DAJSUREN, *et al.*, 2017). Abordagens essas focadas em pontos chaves como a engenharia de requisitos (WEBER *et al.*, 2002), desenvolvimento ágil (KATUMBA *et al.*, 2014) e testes (KASOJU *et al.*, 2013; SUNDMARK *et al.*, 2011), embora a abordagem de teste de integração no nível mais alto ainda seja bastante limitada (CHUNDURI *et al.*, 2016). Além disso, devido à crescente complexidade, dependência e compartilhamento de *software* em veículos automotores, os desafios relacionados a todos os aspectos do *software*, incluindo testes, deverão crescer ao longo dos anos na indústria automotiva (GRIMM *et al.*, 2003; KATUMBA *et al.*, 2014; FRANCO *et al.*, 2014, DAJSUREN, *et al.*, 2017). Isso torna o estudo muito dinâmico, considerando a contínua evolução tecnológica, junto com novas regulamentações e padronizações

para indústria automotiva. Como por exemplo, a introdução do padrão de segurança funcional definido na norma ISO 26262 (ISO, 2011, LARRUCEA *et. al*, 2017), que afeta diretamente os testes de *software*. Dentro desse contexto, existe um desafio muito grande em planejar, estruturar e executar os testes de *software* dentro da indústria automotiva, quando se considera o sistema eletrônico embarcado completo do veículo, devido ao seu grande número de funções, sistemas e componentes.

1.2 OBJETIVOS

Este trabalho pretende estender o conceito de teste multinível, projetando uma metodologia sistemática para avaliar antecipadamente o sistema de aplicação, a fim de definir a estratégia do plano de teste de *software*, gerando transparência para as etapas de testes. Para isso, utiliza mapeamento das funções dentro do sistema embarcado do veículo, que possibilita definição de responsabilidade e rastreabilidade das atividades. Esta proposta foi totalmente desenvolvida incorporando as características da aplicação automotiva, incluindo os requisitos de segurança funcional ISO 26262. Dado o objetivo geral, os principais objetivos da pesquisa são:

- Identificar as características do desenvolvimento de produto e sistemas embarcados que influenciam a definição da estratégia e planejamento dos testes;
- Criar um mapeamento da complexidade das funções dentro dos sistemas embarcados;
- Criar um método para identificação, codificação e contribuições das funções de *software* dentro dos sistemas embarcados;
- Criar um método para definir a estratégia de testes para cada função dentro do conceito de teste de multinível;

- Definir uma estrutura básica para rastreabilidade das atividades de testes.

1.3 CONTEÚDO E ORGANIZAÇÃO

O capítulo 1 apresenta uma introdução da pesquisa, assim como a motivação e seus objetivos. O capítulo 2 é focado no estado da arte da tese, inicia-se com o processo de desenvolvimento geral de um veículo e desafios para desenvolvimento de sistemas eletroeletrônicos automotivos e termina com o tema de teste de *software* de sistemas embarcados em aplicações automotivas. O capítulo 3 apresenta o desenvolvimento da metodologia sistemática para planejamento e estruturação dos testes de *software* em sistemas eletrônicos embarcado na indústria automotiva. O capítulo 4 apresenta um estudo de caso com a aplicação da metodologia proposta e uma análise geral dos respectivos resultados. O capítulo 5 apresenta as conclusões do trabalho, detalhando suas contribuições e os principais pontos que podem ser utilizados para futuras pesquisas.

2 REVISÃO BIBLIOGRÁFICA

Existem quatro pontos importantes que precisam ser abordados nesta tese. O primeiro está relacionado ao processo de desenvolvimento de produto na indústria automotiva. O segundo, a entender as características do desenvolvimento de sistemas embarcados automotivos. Terceiro, apresentar os desafios da indústria automotiva que tenham impactos no processo geral do desenvolvimento de sistemas eletrônicos embarcados. E para finalizar dentro do contexto dos tópicos anteriores, o tema de teste de *software* é estudado, verificando e analisando as características gerais do teste de *software* e seus desafios dentro do contexto da indústria automotiva, conforme a Figura 1.

Figura 1 – Contexto da tese.

Fonte: Elaborada pelo autor.

2.1 PROCESSOS DE DESENVOLVIMENTO DE PRODUTO

Existe um número elevado de abordagens sobre o processo de desenvolvimento de produto, que tem origens em diferentes áreas do conhecimento e valoriza diferentes aspectos do processo. A indústria automotiva também possui esse mesmo cenário, no qual a maioria das empresas estabelece um modelo diferente de processo de desenvolvimento, mas seguindo em muitos casos a base dos modelos clássicos que serão apresentados nesta secção.

Em Harvard e MIT, entre o final da década de 80 e início de 90, vários estudos foram desenvolvidos, com importantes projetos de pesquisa relacionados à manufatura enxuta e a gestão do processo de desenvolvimento de produto. Estes primeiros trabalhos, puramente analíticos, tornaram-se clássicos e são comumente referenciados na literatura sobre desenvolvimento de produto (CLARK *et al.*, 1991; WOMACK *et al.*, 1994). Essas investigações foram a base de uma abordagem para gerenciar processos, apresentada nos livros de Clark e Wheelwright (1993). Nesta abordagem, os autores dividem o processo de desenvolvimento de produto em três etapas maiores: a estratégia de desenvolvimento, onde se apresenta uma estrutura para o planejamento e gerenciamento do portfólio dos projetos em andamento; o gerenciamento do projeto específico, abordando o gerenciamento, liderança e tipos de interação entre atividades; e a aprendizagem, apresentando formas para garantir

a melhoria do processo e gerenciamento organizacional a partir da experiência com o projeto.

Nesta abordagem, os autores dividem o processo de desenvolvimento de produto em três etapas maiores: Estratégia de Desenvolvimento, que apresenta uma estrutura para o planejamento e gerenciamento do portfólio dos projetos em andamento; Gerenciamento do Projeto Específico, abordando o gerenciamento, liderança, tipos de interação entre atividades e outros assuntos relacionados com um projeto específico; e Aprendizagem apresentando formas para garantir a melhoria do processo e a aprendizagem organizacional a partir da experiência com o projeto.

A abordagem proposta por Pugh et al (1996) possui uma forte influência da experiência prática oriunda dos anos trabalhando como projetista e gerente de projetos em diversas indústrias. Sua principal preocupação era a busca de uma visão global da atividade de projeto, ou seja, a superação das visões parciais presentes em cada setor tecnológico específico. Para atingir este objetivo, ele dedicou uma grande ênfase ao desenvolvimento de um modelo que ficou conhecido como *Total Design*. Este modelo possui um conjunto de seis etapas, todas interativas e aplicáveis a qualquer tipo de projeto, independente da disciplina tecnológica envolvida. Cada etapa é representada por um cilindro, significando que nela é empregado um conjunto específico de conhecimentos, compostos por diversas visões tecnológicas parciais.

O autor Don Clausing (1994) influenciado pelo trabalho de Pugh (1991) e Taguchi (1983), adicionando sua própria experiência, criou uma abordagem a qual denominou *Total Quality Development*. O trabalho teve como foco as técnicas QFD (*Quality Function Deployment*), método Taguchi (1983), matriz de Pugh (1991) e conceitos sobre gerenciamento das equipes de desenvolvimento de produto. Uma de suas principais contribuições foi mostrar a integração entre o QFD e o método Taguchi. As fases em que ele divide o processo de desenvolvimento de produto são: Conceito, que foca na metodologia do QFD; *Design*, dividido em projeto dos subsistemas e projetos das partes; e Preparação/produção, dividida em verificação do sistema, prontidão e produção piloto.

Prasad (1996) propõe uma sofisticada abordagem para engenharia simultânea, que engloba diversos fatores em uma estrutura bastante independente das fases de um processo de desenvolvimento de produto. Ele divide a engenharia simultânea em duas rodas, denominadas Organização do Produto e Processo (*Product and Process Organization Wheel* - PPO) e Desenvolvimento de Produto Integrado (*Integrated Product Development Wheel* - IPD). Ambas possuem no seu centro a descrição dos quatro elementos de suporte desta metodologia, que são: modelos, métodos, métricas e medidas.

As duas rodas possuem também anéis intermediários idênticos que representam as equipes de trabalho, ou a estrutura organizacional, que dirige as ações dentro do processo de engenharia simultânea. A primeira roda, PPO, aborda os fatores que determinam o grau de complexidade do gerenciamento do desenvolvimento de produto e os fatores organizacionais. A segunda roda, IPD, define de uma maneira bastante flexível a integração do processo de desenvolvimento de produto.

Outro modelo muito utilizado pelas indústrias é o “*Stage Gate*” que foi desenvolvido pelos autores Robert G. Cooper e Scott J. Edgett, na segunda metade da década de 80, a partir do lançamento do livro: “*Winning at New Products*”, em 1986 (COPPER, 1986). É um processo operacional e conceitual utilizado para definir, em cada “*gate*”, o que precisa ser entregue dentro do projeto, desde o início do desenvolvimento até o seu lançamento. Este processo divide os esforços em fases (*sages*) distintas, separadas por pontos de decisões gerenciais (*gates*). Equipes multifuncionais devem completar uma série de atividades e tarefas pré-estabelecidas durante cada fase antes de obter a aprovação no *gate* e, assim, avançar para a próxima etapa (COOPER, 2000).

Entre as contribuições de Cooper, podem-se destacar alguns importantes elementos para esse trabalho. O “*Stage*”, também conhecido como fase, é o ponto de controle, onde uma equipe multifuncional, coordenada pelo líder do projeto, reporta as atividades do projeto; o “*Gate*”, ou ponto de verificação do projeto, é aplicado no final de cada “*stage*” ou fase. Quando o status do projeto é verificado e

caso as atividades estejam dentro do planejado, principalmente as atividades daquela fase, o projeto é aprovado para a próxima fase.

A análise e alocação de recursos devem ser balanceadas para os novos desenvolvimentos e para a resolução dos problemas diários. Os autores mencionam três áreas ou etapas críticas em que tipicamente se percebe a escassez de recursos: “*up-front homework activities*”, ou atividades ligadas à pesquisa de mercado; na formação de verdadeiras equipes multifuncionais e nas decisões sobre continuar ou parar o projeto, nos “*gates*”.

Algumas funções e responsabilidades de cada participante precisam ser estabelecidas para esse processo, dentre elas destacam-se: “*gate-keepers*”, ou tomadores de decisões, que atuam nos “*gates*” avaliando as atividades realizadas na fase anterior e decidindo sobre o futuro dos projetos. Normalmente são elementos da alta administração da empresa: “*owners*” ou gerentes de projeto, que têm a função de comandar a equipe de trabalho e controlar todas as fases; “*sponsors*” ou patrocinadores, elementos da alta direção da empresa que apoiam e cobram o bom andamento do projeto; adoção de uma plataforma de comunicação, focado em informar e educar; sistema para avaliar e entender o impacto do novo processo na cultura e nos atuais sistemas da organização; gerenciador efetivo e eficaz de gerenciamento de mudanças (*change management*).

Cooper (2008) apresentou algumas conclusões importantes, após a aplicação do processo de “*Stage gates*” em algumas empresas, destacam-se: o “*Stage-Gate*” não é um processo funcional, sequencial e departamental, as etapas são multifuncionais, sem domínio de nenhuma área funcional; não é um livro de regras, usando as palavras do autor; é um mapa direcional e flexível, que deve ser adaptado a cada tipo de projeto, com a possibilidade de pular tarefas e/ou etapas; não é um sistema linear, as atividades podem ser realizadas em paralelo, com “*loopings*” e iterações; não é um mecanismo de controle de executivos e auditores, é um processo para organizar recursos para utilização nos projetos, de forma a apressar a introdução do projeto no mercado; não é um processo estagnado, é um processo dinâmico, compreensivo, maleável e integrado que sempre evolui; não é um processo burocrático, o objetivo é criar uma sistemática em que qualquer reunião,

atividade ou procedimento que não agregue valor deve ser eliminado; não é o mesmo que “gerenciamento de projetos”, *Stage-gate* é um macroprocesso, gerenciamento de projetos é um microprocesso, método que pode ser utilizado dentro do *Stage-gate*.

A Figura 2 mostra o modelo clássico “*Stage Gate*”, dividido em seis fases: fase da ideia, coleta das ideias para o projeto; fase do estágio 1, avaliação preliminar técnica e de mercado; fase estágio 2, identificação do conceito no mercado, geração do conceito do produto e processo, justificativa do negócio e plano de ação para próximas fases; fase estágio 3, desenvolvimento do produto (projeto de engenharia e protótipos), mapeamento do processo de produção, desenvolvimento de plano de marketing e definição do plano de testes para a próxima fase; fase estágio 4, testes para verificação e validação do produto proposto, do mercado e do processo produtivo; fase estágio 5, início da produção e comercialização do produto; fase de revisão e pós-lançamento, avaliação e controle do produto no mercado e avaliação das lições aprendidas no projeto. No final de cada fase existem “*gates*”, que servem como pontos de controle para avaliação do status do projeto e aprovação da fase para continuidade do projeto.

Figura 2 – Modelo *Stage Gate* proposto por Cooper em 2000.

Fonte: Cooper (2000).

2.1.1 PROCESSO DE DESENVOLVIMENTO DE PRODUTO AUTOMOTIVO

Além dos modelos clássicos de processos de desenvolvimento de produto citados anteriormente, existem outros autores. Nesta tese destaca-se o modelo de Rozenfeld *et al.* (2006), que pode ser considerado um dos modelos com uma abordagem completa, incluindo várias etapas do desenvolvimento de produto.

Rozenfeld *et al.* (2006) divide o processo de desenvolvimento de produto em três fases principais: pré-desenvolvimento, desenvolvimento e pós-desenvolvimento, conforme detalhado na Figura 3.

Figura 3 – Modelo genérico para processo de desenvolvimento de produto.

Fonte: Rozenfeld (2006).

Na fase de pré-desenvolvimento é delineada a estratégia de processo de desenvolvimento de produto (PDP) da empresa, incluindo os objetivos e metas relativos ao portfólio. Esta fase se divide em planejamento estratégico do produto e planejamento do projeto. O objetivo do planejamento estratégico do produto é a definição de um portfólio de produtos para a empresa, ou seja, descrever uma linha de produtos e os projetos a serem desenvolvidos. Este planejamento estratégico pode envolver, inclusive, planos de retiradas de produtos do portfólio. O planejamento do projeto é um elemento essencial desta fase, apontando desde o escopo do projeto e do produto, orçamento, prazos, definição do pessoal, recursos, procedimentos de avaliação, análises de risco e indicadores de desempenho do projeto e produto.

A macro fase de desenvolvimento envolve o processo de projeto, detalhamento e produção do produto. Esta fase divide-se em projeto informacional,

projeto conceitual, projeto detalhado, preparação da produção e lançamento do produto. Na fase de projeto informacional devem resultar as especificações-meta do produto, que orientarão a geração de soluções, fornecendo uma base para elaboração de critérios de avaliação e tomadas de decisão. A fase seguinte trata do projeto conceitual, que objetiva a busca, criação, representação e seleção de soluções para o problema identificado no projeto informacional.

No projeto detalhado, são realizadas todas as especificações e detalhamentos, os protótipos são testados resultando no detalhamento de todos os recursos, manuais de uso, instruções de assistência, além de suporte às equipes de venda. Finalmente, na preparação da produção é produzido um lote de produção piloto, o que permite que o processo produtivo seja mapeado e melhor definido. Por fim, ocorre o lançamento do produto, implicando em projeto e implementação de processos de assistência técnica e auxílio ao consumidor (ROZENFELD, 2006).

Finalmente, a última fase do modelo de Rosenfeld trata do pós-desenvolvimento, que inclui as atividades do pós-venda até o fim do ciclo de vida do produto. Esta fase divide-se em duas partes: acompanhamento e descontinuidade. O acompanhamento do produto inclui a realização de atividades como: avaliação da satisfação do cliente, o monitoramento do desempenho do produto, auditoria pós-projeto e o registro de lições aprendidas. A descontinuidade inclui as atividades de: analisar e aprovar descontinuidade do produto, planejar a descontinuidade do produto, preparar o recebimento, acompanhar o recebimento do produto, descontinuar a produção, finalizar suporte ao produto e a avaliação geral e encerramento do projeto (ROZENFELD, 2006).

Modelos para processos de desenvolvimento do produto vêm sendo propostos nas últimas décadas, considerando a evolução da indústria de bens de consumo e as necessidades de levar ao mercado produtos de sucesso no menor tempo possível e contendo elevado nível técnico nos aspectos de confiabilidade.

Estes modelos de desenvolvimento têm como base o processo de desenvolvimento tradicional de produtos em massa, amplamente utilizado pelas indústrias automobilísticas europeias e americanas, e o enxuto *Lean Development*

que é apresentado como bem-sucedido, principalmente nas indústrias automobilísticas japonesas (LIKER, 2004).

A partir do modelo de Rozenfeld *et al.* (2006), pode-se criar um modelo simplificado (Figura 4), utilizado como base para muitos processos de desenvolvimento da indústria automotiva, definindo um QG para as principais entregas ou pontos de controle do desenvolvimento do veículo. A fase do conceito no *quality gate* seis, a fase de projeto no *quality gate* cinco, a fase de protótipo no *quality gate* quatro, a fase de teste no *quality gate* três, a fase de produção no *quality gate* dois e a fase de pós-vendas no *quality gate* um. A largura da faixa em azul representa a quantidade de esforços de cada tópico na linha do tempo e também apresenta de forma clara que existem trabalhos paralelos em todos tópicos, porém com intensidades diferentes.

Figura 4 – Modelo de processo de desenvolvimento com os *quality gates* definidos e linhas dos esforços para cada departamento em cada uma das etapas.

Fonte: Elaborado pelo autor.

A fase de conceito está totalmente relacionada à especificação de um projeto. Consiste em uma lista de requerimentos a ser cumprida para a produção de um produto de sucesso e é normalmente realizada partindo de uma comunicação formal

entre um vendedor e um comprador (DIETER, 1990). Ao preparar uma especificação detalhada, é importante afirmar se os itens requisitados são uma necessidade (N), que devem ser atendidas em quaisquer circunstâncias, ou se são meramente desejos (D) que devem ser considerados quando possível (PAHL *et al.*, 2013).

Uma especificação inicia-se na pesquisa de mercado, seguida de um comparativo com os concorrentes, busca por patentes, examinando a literatura técnica (DIETER, 1990). Segundo a *Institution of Production Engineers London*, os 28 elementos primários de uma especificação que podem fazer parte de uma lista estão descritos na Figura 5.

Figura 5 – Elementos para especificações

Fonte: Dieter (1990)

Geralmente, engenheiros demonstram dificuldades em elaborar sua primeira especificação, assim, recomenda-se que cada item contenha uma tarefa e algumas características técnicas associadas, pois ajuda a firmar a natureza e escopo do problema (PAHL *et al.*, 2013).

Especificamente para o desenvolvimento dos sistemas eletrônicos, a etapa do conceito é uma das mais importantes, principalmente para os aspectos integração e gestão do conhecimento. Alguns dos requerimentos necessários para o desenvolvimento desses sistemas são descritos abaixo:

- A primeira etapa a ser definida é o desejo do cliente, para criar a de lista funções do veículo, que será a base para todo o desenvolvimento. A descrição precisa ser em alto nível, não devem existir aspectos técnicos, nem nomes de módulos eletrônicos, dando liberdade para que a criatividade e inovação sejam utilizadas nas próximas etapas;
- Com a função descrita de forma genérica, o grupo de especialistas define as condições de contorno da função, questões de operacionalidade e variáveis envolvidas;
- As lições aprendidas nos projetos anteriores precisam ser avaliadas principalmente nesta etapa, mas ocorre de forma organizada, se no processo foram incluídos esses aprendizados. Isso deve ser sistematizado e padronizado;
- Definição de cada subsistema para atender todas as funções solicitadas;
- Definição dos componentes necessários;
- Criação da matriz de integração de todos os subsistemas e componentes;
- Especificação detalhada de todos os componentes para que o departamento de compras possa buscar os fornecedores potenciais no mercado;
- Definir os fornecedores de cada componente;

- Revisão das especificações, juntamente com os fornecedores;
- Definição das características mecânicas do componente, tamanho, peso e fixação, para integração no conceito geral do veículo.

Na Fase de *design*, as premissas definidas na fase de conceito são utilizadas como parâmetros para a solução do projeto que delimita seu escopo. O termo *design* historicamente foi definido para desenvolvimento dos desenhos de peças mecânicas, entretanto, com a introdução dos sistemas eletrônicos, o nome ficou mais amplo, considerando que o *design* precisa incluir o *software* e *hardware*. No caso dos sistemas eletrônicos, a fase de projeto inicia-se com uma especificação detalhada do *software* e *hardware*, com base nessas informações o *design* de *hardware* e *software* pode ser desenvolvido.

Dentro das atividades de especificação e *design*, existem alguns pontos importantes que precisam ser desenvolvidos, como a definição da interdependência entre os componentes e sistemas, desenvolvimento do *software* e *hardware*, definição do número e prazo das amostras, definição do *layout* dos componentes, dimensionamento do balanço energético dos componentes elétricos, criação do diagrama elétrico ou esquema elétrico das ligações dos componentes e criação do desenho do chicote elétrico.

A finalização dessa etapa para as peças mecânicas termina com a construção de uma peça protótipo, que é verificada em relação ao desenho projetado; caso esteja dentro dos parâmetros, é liberada para montagem no veículo de testes. No caso dos sistemas eletroeletrônicos, a verificação do “design” com a peça protótipo exige uma análise mais sofisticada, principalmente pela parte do *software*, exigindo ferramentas de validação adequadas para esse tipo componente, como HiL (*Hardware in the loops*) e outras formas de simulação e testes (STRICHMAN et al., 2017). O processo de desenvolvimento geral do veículo não diferencia os sistemas elétricos dos componentes mecânicos, o que torna o controle e gestão do desenvolvimento mais complexo.

O teste é uma fase importante para qualquer projeto e precisa de um planejamento dos testes para garantir que o produto atenda aos requisitos definidos na fase inicial do projeto, dentre eles as necessidades do cliente, os requisitos dos órgãos regulamentadores, entre outros.

Para a realização do teste, normalmente é gerada uma especificação dos testes que define como o produto ou projeto precisa ser testado, em que são definidos a metodologia, as ferramentas, a rotina dos testes e os critérios de aprovação. A base da especificação de teste é a documentação do projeto e as normas dos órgãos regulamentadores.

No caso dos sistemas eletrônicos embarcados, os testes são divididos em diversas etapas e, especificamente para os sistemas eletrônicos, a etapa de teste no veículo representa uma etapa de integração e também de comprovação da eficácia dos testes realizados previamente nos componentes.

A fase de produção pode ser dividida em duas etapas: a produção dos componentes do veículo e a montagem do veículo. A produção do componente pode ser realizada dentro do OEM quando, além da montagem do veículo, a empresa fabrica os componentes. Mas existe a opção da terceirização, quando os componentes são fabricados por outras empresas.

Antes do início da produção do veículo, um processo chamado de industrialização é realizado, para preparar toda a infraestrutura da fábrica para fabricação do veículo e também para validação dos fornecedores de componentes e sistemas. Dentro dessa etapa de preparação da produção, existe uma etapa piloto, com a produção completa de um ou mais veículos, na qual o processo detalhado com as atividades de produção é executado por equipes responsáveis pela orientação e disseminação do conhecimento para todos os funcionários da produção.

Além da montagem, os sistemas eletrônicos em grande parte dos veículos exigem tanto um processo de carregamento de *software* e parâmetros que também

precisam ser validados, fechando com o teste ou análise da diagnose do veículo, quanto um processo bastante preciso de verificação da qualidade de fabricação e montagem do veículo.

2.2 PROCESSO DE DESENVOLVIMENTO DE SISTEMAS ELÉTRICO-ELETRÔNICOS AUTOMOTIVOS

O desenvolvimento dos sistemas eletrônicos está muito integrado com o processo de desenvolvimento geral do veículo, e o desenvolvimento de um veículo inicia-se com uma identificação de demanda para o mercado, que se origina de algumas características específicas que diferenciam o produto a ser desenvolvido dos produtos que existem no mercado. Essas características podem ser diversas, como mecânicas, estéticas, conforto, segurança, entretenimento, entre outras, que geram as funções ou características do veículo.

Como citado no capítulo anterior, em torno de 90% das inovações, ou seja, a maior parte das novas características dos veículos está ligada aos sistemas eletroeletrônicos. Esse fato transformou o desenvolvimento desses sistemas em um dos assuntos mais estudados e discutidos dentro do desenvolvimento de produto automotivo (BROY *et al.*, 2001; COULON, 2014; HUDI, 2016; HAGHIGHATKHAH *et al.*, 2017).

2.2.1 LISTA DE FUNÇÕES

O veículo possui diversas funções que são facilmente compreendidas pelo seu “cliente”. Uma função pode ser desde uma simples ação de ligar o motor, até um controle de velocidade cruzeiro adaptativo do veículo. Cada função precisa ser atrelada a um sistema para implementação. O conjunto geral de funções do veículo é chamado de lista de funções. A Figura 6 mostra um exemplo de uma lista de funções configurada de acordo com a necessidade de cada veículo, em que cada letra representa uma função específica.

Figura 6 – Lista de funções.

Funções disponíveis																		
A	B	C	D	E	F	G	H	I	J	L	M	N	O	P	Q	R	S	...

Fonte: Elaborada pelo autor.

As contínuas inovações e tecnologias geram um aumento das funções e um grande número de combinações entre elas. Algumas funções podem ser de série, ou seja, farão parte de todos os modelos de veículos, como o caso das funções H e M da Figura 7, ou funções opcionais, que nem todos os veículos possuem.

Figura 7 – Exemplo da configuração das funções por veículos.

Veículos	Configuração dos veículos																
1	A	B	D	E	G	H	I	J	L	M	N	O	P	Q	R	S	
2	B	C	E	F	G	H	I	J	L	M	N	O	P	Q	R	S	
3	A	B	C	E	H	J	L	M	N	O	P	Q	R	S			
4	D	E	F	G	H	I	J	L	M	N	O	P	Q	R	S		
5	B	D	E	G	H	I	M	N	O	P	Q	R	S				
6	A	B	D	E	G	H	I	J	M	N	P	Q	R	S			
7	C	D	E	F	G	H	I	J	L	M	N	O	P	Q	R	S	
8	G	H	I	J	L	M	N	O	P	Q	R	S					
9	A	B	C	D	E	F	G	H	I	J	L	M	N	O			
10	A	B	D	E	G	H	I	J	L	M	P	Q	R	S			
11	B	D	E	G	H	I	J	L	M	N	O	P	Q	R			
...	E	G	H	I	J	L	M	N	O	P	Q						

Fonte: Elaborada pelo autor.

Independentemente da quantidade de funções dentro do veículo, ou do tipo de combinações e configuração, em série ou opcional, o sistema precisa funcionar apropriadamente.

2.2.2 SISTEMA ELETRÔNICO AUTOMOTIVO

O veículo é composto por diversos sistemas, e cada sistema possui um conjunto de elementos que representa uma parte da funcionalidade do veículo, conforme Figura 8. O sistema representa o conjunto das funções ligadas ao seu domínio (WINNER *et al.*, 2018). A forma como o sistema é definido é diferente de

uma empresa para outra, e normalmente é um nome correlacionado com um pacote de funções.

Figura 8 – Sistemas elétricos com suas respectivas funções.

Fonte: Elaborada pelo autor.

2.2.3 COMPONENTE ELETROELETRÔNICO AUTOMOTIVO

Componente é uma parte do sistema que realiza ou contribui com algumas funções. Um componente também é chamado de subsistema. A Figura 9 mostra a representação de um conjunto de sistemas de um veículo, na qual cada componente faz parte de pelo menos um sistema, com o intuito de realizar as funções e/ou contribuir com alguma função. Podem ocorrer casos em que o componente contribui com funções de outros sistemas; por exemplo, o componente 8 participa de três sistemas: os sistemas 2,3 e 4.

Figura 9 – Representação de um conjunto de sistemas eletroeletrônicos de um veículo, com suas funções, componentes e subsistemas.

Fonte: Elaborada pelo autor.

2.2.4 ARQUITETURA ELÉTRICA

O conjunto de diversos sistemas eletrônicos implementados e conectados em uma dada aplicação embarcada é chamado de Arquitetura Eletroeletrônica, ou simplesmente Arquitetura Elétrica (RUSHTON *et al.*, 2000; REICHART *et al.*, 2004, STARON *et al.*, 2017). Essa nomenclatura é bastante utilizada pelo setor automotivo.

A definição da arquitetura começa com a lista de funções do veículo, para cada função é necessário um componente ou conjunto de componentes para executá-la. A estrutura que interliga esses componentes forma a arquitetura. Nesta etapa, a rede de comunicação entre os módulos e a matriz de dados que circularão na rede é definida. O conteúdo dos dados é definido em detalhes na etapa de especificações dos sistemas e componentes.

2.2.5 MODELO DE REFERÊNCIA PARA DESENVOLVIMENTO DE SISTEMAS

O processo de desenvolvimento de sistemas eletrônicos na indústria automotiva possui diversas características oriundas de diferentes modelos, conceitos e abordagens utilizadas para sua aplicação. Esta investigação apresenta as principais referências que fundamentaram o modelo existente na indústria automotiva.

2.2.5.1 Engenharia de sistemas

Engenharia de Sistemas é uma abordagem interdisciplinar que define os meios para uma realização de sistemas com sucesso. Inicia-se com as necessidades e funcionalidades do cliente, definição da síntese do projeto e finaliza-se com a validação. Ela integra todas as disciplinas e grupos especializados, formando um processo de desenvolvimento estruturado que procede do conceito à produção, para a operação (INTERNATIONAL COUNCIL OF SYSTEMS ENGINEERING, 2016).

O sistema é o produto da associação de elementos diferentes, que produzem resultados que não podem ser obtidos pelos elementos sozinhos. Os elementos podem incluir pessoas, *hardware*, *software*, instalações, políticas e documentos; ou seja, todos os elementos necessários para produzir resultados em nível de sistemas. Os resultados da combinação desses elementos podem ser medidos através da funcionalidade do sistema, do nível da qualidade do sistema, características construtivas, desempenho do sistema, entre outras formas. O resultado do sistema como um todo é criado principalmente pela relação entre as partes; isto é, como eles são interligados.

Outro ponto bastante importante é que as atividades relacionadas ao desenvolvimento de sistemas normalmente não são sequenciais, as funções são executadas de uma maneira paralela e iterativa (INTERNATIONAL COUNCIL OF SYSTEMS ENGINEERING, 2016).

O processo de combinação entre elementos que cria um sistema e gera resultado através dele faz parte do procedimento corriqueiro dentro das equipes e organizações, de tal forma que essa abordagem da definição de um sistema foi descrita por BAHILL *et al.*, (1998) com a seguinte sequência de atividades: identificar o problema, investigar alternativas, modelar o sistema, integrar, lançar o sistema, avaliar o desempenho e reavaliar (Figura 10). Esse processo foi chamado de “SIMILAR”, e é base de muitos processos de desenvolvimento de sistemas (BAHILL *et al.*, 1998).

Figura 10 – Processo desenvolvimento de Engenharia de Sistemas – SIMILAR

Fonte: Elaborada pelo autor.

- **Identificar o Problema:** começa com uma descrição das funções em um nível superficial do que o sistema vai executar, sob a forma de uma declaração de missão, um conceito de operações ou uma descrição da deficiência que deve ser melhorada. Os requerimentos são focados na funcionalidade e não na implementação. A declaração do problema deve expressar as necessidades dos clientes em termos funcionais ou comportamentais. Ela pode ser composta por palavras ou como um modelo. Entradas vêm de usuários finais, operadores, equipe de manutenção, fornecedores, compradores, proprietários, agências reguladoras, as vítimas, patrocinadores, fabricantes e outras partes interessadas.
- **Investigar Alternativas:** alternativas são criadas e avaliadas com base no desempenho, cronograma, custo e risco. Como nenhum projeto é inquestionavelmente melhor em todas as figuras de mérito, técnicas de

apoio à decisão multicritério devem ser utilizadas para revelar as alternativas mais otimizadas. Essa análise deve ser revisada sempre que mais dados estiverem disponíveis. Por exemplo, figuras de mérito devem ser calculadas inicialmente com base em estimativas dos engenheiros do projeto e, principalmente, utilizando dados de projeto anteriores. Então, ao mesmo tempo, os modelos devem ser construídos e avaliados, dados de simulação devem ser derivados, e protótipos devem ser construídos e medidos e, finalmente, os testes devem ser executados no sistema real. As alternativas devem ser julgadas pelo cumprimento de capacidade em relação aos requisitos.

- Modelar o sistema: Modelos são desenvolvidos para a maioria dos projetos alternativos. O modelo para a alternativa preferida será ampliado e usado para ajudar a gerir o sistema durante todo o seu ciclo de vida. Muitos tipos de modelos de sistema são utilizados, tais como análogos físicos, equações analíticas, máquinas de estado, diagramas de blocos, diagramas de fluxo funcionais, modelos de orientação a objetos, simulações de computador e modelos mentais. A engenharia de sistemas é responsável pela criação de um produto e um processo para a sua produção, assim, modelos devem ser construídos tanto para o produto quanto para o processo. O modelo de processo permite, por exemplo, estudar mudanças de programação, criar gráficos dinâmicos e realizar análises de sensibilidade para mostrar os efeitos de retardar ou acelerar determinados subprojetos. Com a execução dos modelos de processo, as atividades e custos de todo o processo são identificados. Os modelos de produtos ajudam a explicar o sistema, e podem ser usados em estudos de *trade-offs* e gestão de riscos.
- Integrar: os sistemas, as empresas e as pessoas devem ser integrados para que eles interajam entre si. Integração representa a ligação de todos os elementos para a realização de um resultado conjunto. Para isso, as interfaces e as funções de cada subsistema precisam ser definidas e sistematicamente estruturadas. Subsistemas são aplicados com o

objetivo de descentralizar e minimizar a quantidade de informação trocada entre os sistemas, facilitando a definição de responsabilidades.

- Lançar o sistema: significa executar o sistema e produzir resultados. Esta é a fase em que a alternativa preferida é concebida em detalhe; as partes são construídas ou compradas e são integradas e testadas em vários níveis, conduzindo ao produto certificado. Paralelamente, os processos necessários são desenvolvidos e aplicados de modo que a produção aconteça. Na concepção e produção do produto, a devida consideração é dada às suas interfaces com os operadores (humanos, que terão de ser treinados) e outros sistemas com os quais o produto tem interface. O processo de concepção e produção do sistema é iterativo e realimentado, os resultados podem modificar ou melhorar definições realizadas em etapas anteriores do processo.

Produtos dos sistemas de engenharia são uma declaração de missão, um documento de requisitos, incluindo a verificação e validação, uma descrição das funções e objetos, figuras de mérito, plano de teste, um desenho fronteiras do sistema, documento de controle de interface, lista de resultados, modelos, análise de sensibilidade, estudo de troca, análise de risco, análise do ciclo de vida e descrição da arquitetura física. Os requisitos devem ser validados, ou seja, é preciso saber se o sistema está sendo construído de forma correta. As funções do sistema devem ser mapeadas para os componentes físicos. O mapeamento de funções de componentes físicos pode ser um a um ou de muitos para um, se uma função é atribuída a dois ou mais componentes físicos.

- Avaliar o resultado: figuras de mérito, medidas de desempenho técnico e métricas são utilizadas para avaliar o desempenho. Figuras de mérito são usadas para quantificar requisitos, focando no produto, e medidas de desempenho técnico para mitigar o risco durante o projeto e fabricação. Métricas (incluindo comentários de satisfação do cliente, produtividade, número de relatórios de problemas, ou o que é crítico para o negócio) são utilizadas para ajudar a gerenciar processos de uma empresa. A

medição é necessária, ou o controle não pode ser aplicado. Cada subsistema é atribuído a uma parte do orçamento total, e ao gerente do projeto é atribuída uma reserva. Estes orçamentos de recursos são geridos durante todo o ciclo de vida do sistema.

- Reavaliar é sem dúvida a mais importante destas funções. Durante um século, os engenheiros usaram uma realimentação para ajudar os sistemas de controle e melhorar o desempenho. É uma das ferramentas fundamentais de engenharia. A reavaliação deve ser um processo contínuo, com muitos *loops* paralelos. Reavaliar significa observar saídas e utilizar essa informação para modificar o sistema, as entradas, o produto ou o processo.

A engenharia de sistema é uma das bases para o desenvolvimento de sistemas complexos e muitos dos seus conceitos são utilizados dentro dos processos de desenvolvimento de produto, incluindo aplicações com sistemas embarcados.

2.2.5.2 AUTOSAR

Impulsionado pelo surgimento de aplicações automotivas inovadoras, a arquitetura EE automotiva atingiu um nível de complexidade que requer um avanço no processo de desenvolvimento, a fim de gerenciar e controlar de forma satisfatória os avanços tecnológicos, as exigências dos clientes e os requisitos legais (DAJSUREN *et al.*, 2017). Esta necessidade tem particularmente alto grau de dificuldade para os fabricantes de veículos e seus fornecedores, que são confrontados muitas vezes com requisitos conflitantes de:

- Legislação - itens importantes incluem aspectos ambientais e requisitos de segurança;
- Requisitos de conveniência do passageiro e de serviço, nos domínios funcionais de conforto e entretenimento;

- Assistência ao motorista e à dinâmica veicular - itens importantes incluem detecção e controle de estados críticos da dinâmica veicular e navegação em ambiente de tráfego de alta densidade.

Os componentes de software dos sistemas automotivos em geral são desenvolvidos pelos fornecedores, montadoras (OEMs) ou por uma companhia de software independente. Todos reconheceram que este é um desafio de toda a indústria e decidiram trabalhar juntos para enfrentar o desafio (PELZ *et al.*, 2005, FÜRST *et al.*, 2016). O objetivo comum foi criar uma base de desenvolvimento para colaboração da indústria sobre as funções básicas, proporcionando uma plataforma que continua a incentivar a concorrência em funções inovadoras. Assim, uma parceria de desenvolvimento chamada AUTOSAR (*AUtomotive Open System Architecture*) (AUTOSAR, 2018) foi formada, com os objetivos de:

- Cumprir os requisitos de veículos futuros, como disponibilidade e segurança, atualizações e manutenção de *hardware* e *software* embarcados;
- Aumenta a escalabilidade e flexibilidade para integração e transferência de funções;
- Inserir componentes de *hardware* e *software* do tipo COTS (Produtos de Prateleiras - *Commercial Off the Shelf*) em todas as linhas de produtos;
- Contenção do produto, complexidade do processo e o risco melhorado;
- Otimizar custos de sistemas escaláveis.

O padrão AUTOSAR servirá como uma plataforma sobre a qual aplicações em veículos serão implementados e também para minimizar as barreiras existentes entre domínios funcionais. Será possível mapear as funções e redes funcionais para diferentes nós de controle no sistema, quase independentemente de *hardware*

associado (FÜRST *et al.*, 2016, RAN *et al.*, 2017). Principais motivações do padrão AUTOSAR (FENNEL *et al.*, 2006):

- Gestão da complexidade elétrica e eletrônica automotiva, por causa do crescimento de novas funções;
- Flexibilidade para modificação do produto, melhorias e atualizações;
- Escalabilidade de soluções dentro e através de linhas de produtos;
- Melhoria da qualidade e confiabilidade dos sistemas elétricos e eletrônicos automotivos.

O AUTOSAR é uma arquitetura de *software* automotivo aberto e padronizado. É baseado no conceito de componentes de *software*, o qual é desenvolvido e mantido por fabricantes de automóveis, fornecedores e desenvolvedores de ferramentas, que podem estabelecer o melhor caminho para suportar as aplicações da próxima geração de sistemas embarcados automotivos (AUTOSAR, 2016). Essa arquitetura facilita a organização das atividades de teste de software (FÜRST *et al.*, 2016, RAN *et al.*, 2017), principalmente no nível mais baixo dos testes, o componente para teste de *white box*.

2.2.5.3 Modelo em V

O Modelo V (CLAUDIO, 2013) é um modelo conceitual de Engenharia de Sistemas/Desenvolvimento de Produto visto como melhoria ao problema de reatividade do modelo em cascata. Ele permite que, durante a integração de um sistema em seus diversos níveis, os testes sejam baseados nos próprios requisitos do componente ou interface que está sendo testado; em contraste com modelos anteriores, nos quais o componente era testado contra a especificação do componente/interface. É importante ressaltar a diferença entre requisito e especificação.

O Modelo V virou um padrão da indústria de *software* depois de 1980. E após o surgimento da Engenharia de Sistemas, tornou-se um conceito padrão em todos os domínios da indústria, considerando que poucos avanços em processos de desenvolvimento de *software* existiam antes dele (SCHÄUFFELE *et al*, 2006; SPILLNER *et al*, 2007).

2.2.5.4 “Design” baseado em modelos

Model based design (MBD) é um método matemático e visual de resolver os problemas associados principalmente à concepção de controle do complexo, (REEDY *et al.*, 2010), processamento de sinal (AHMADIAN *et al.*, 2005), à sistemas de comunicação, equipamentos industriais e também aplicações automotivas (EWEN, 2006). “Design” baseado em modelo é uma metodologia aplicada na concepção de *software* incorporado (LENNON, 2008; NICOLESCU, 2009).

No caso de desenvolvimento automotivo, a aplicação crescente de *software* incorporado em automóveis resultou no aumento da complexidade, tornando-se um desafio para os processos de *design* convencionais. Além disso, com a inclusão de *software* em funcionalidades de sistemas críticos, como freios e direção, normas de segurança funcional precisam ser atendidas dentro dos projetos.

Processos de desenvolvimento desempenham um papel decisivo na resposta aos desafios da complexidade, produtividade e qualidade. Devido à sua capacidade para enfrentar esses desafios, MBD tem sido amplamente utilizado em toda a comunidade de engenharia de *software*. Mais recentemente, montadoras e fornecedores começaram a considerar e adotar design *Model-Based* para o desenvolvimento de *software* embarcado em aplicações de alta integridade, que precisam atender aos requisitos da norma ISO 26262 (CONRAD, 2012).

Algumas ferramentas de *design*, como “*Simulink*” e “*Stateflow*”, podem ser utilizadas ao longo de múltiplas fases do ciclo de vida do *software*. Aplicando o conceito MBD, permitindo a simulação precoce e testes do comportamento funcional, assim como na geração automática de código. A geração de código em MBD substituiu a codificação manual em vários domínios e cada vez mais vem

sendo implantada em aplicações que exigem alta integridade (JABLONSKI *et al.*, 2008).

O uso de MBD para aplicações de alta integridade exige consideração extra e rigor porque as exigências adicionais impostas pelas normas de segurança funcionais, tais como ISO 26262, devem ser cumpridas. Para melhorar a qualidade e segurança funcional, modelos e códigos gerados devem ser submetidos a uma combinação de medidas de garantia de qualidade. Além disso, a conformidade com a norma ISO 26262 necessita ser demonstrada (CONRAD, 2012, LARRUCEA *et. al*, 2017).

2.2.5.5 Modelo de desenvolvimento por funções

O modelo de desenvolvimento por função ou modelo funcional em engenharia de sistemas e engenharia de *software* é uma representação estruturada das funções (atividades, ações, processos, operações), dentro do sistema modelado ou área temática (FIPS, 1993). Atualmente, muitas empresas o substituíram, em muitos casos pelo MBD, principalmente em desenvolvimento de *software*. Com os desafios atuais, especialmente com a exigência do desenvolvimento dos projetos de acordo com a ISO 26262 (MAUBORGNE, *et al.*, 2016), o mapeamento do projeto por uma perspectiva funcional tem auxiliado o desdobramento da função dentro dos sistemas eletrônicos distribuídos. Neste trabalho, esse conceito será integrado ao processo de desenvolvimento de sistemas eletroeletrônicos automotivos.

O modelo de função no campo da engenharia de sistemas e engenharia de *software* teve origem nas décadas de 1950 e 1960, mas a origem da modelagem funcional da atividade organizacional remonta ao final do século XIX. Naquela época, apareceram os primeiros diagramas retratando as atividades dos negócios, ações, processos ou operações, sendo que na primeira metade do século XX, surgiram os primeiros métodos estruturados para documentar as atividades de processos de negócios. Um desses métodos foi o gráfico de fluxo de processo, introduzido por Frank Gilbreth aos membros da *American Society of Mechanical Engineers* (ASME), em 1921 (GHAHAM, 2004).

A necessidade de identificar e manipular as propriedades de um sistema como um todo, que em projetos de engenharia complexos podem diferir muito da soma de propriedades das peças, motivaram várias indústrias a aplicar a disciplina. Um dos primeiros a definir o modelo de função neste campo foi o engenheiro britânico William Gosling. Em seu livro “O design dos sistemas de engenharia”, ele declarou que um modelo funcional deve alcançar dois objetivos: 1) fornecer uma descrição do rendimento mecânico capaz de definir completamente os primeiros e últimos estados da função e, talvez em alguns dos estados intermediários; e 2) deve oferecer algum meio pelo qual qualquer entrada, corretamente descrita em termos mecânicos, possa ser utilizada para gerar uma saída que seja o resultado da descrição da função.

Um dos primeiros modelos funcionais bem definidos foi o Fluxo Funcional de Diagrama de Blocos (FFBD), desenvolvido pela “TRW *Incorporated*” na década de 1950 (WEILKIENS, 2011). Nos anos 1960, ele foi explorado pela NASA para visualizar a sequência temporal dos eventos em sistemas espaciais e missões de voo. Além disso, é amplamente utilizado na engenharia de sistemas clássicos para mostrar a ordem de execução das funções do sistema (DRUFRESNE *et al.*, 2003).

2.3 DESAFIOS PARA DESENVOLVIMENTO DE SISTEMAS ELÉTRICO-ELETRÔNICOS AUTOMOTIVOS

A indústria automotiva passa por uma fase crítica de mudanças de paradigmas, isso devido a vários fatores, como concorrência acirrada, clientes com elevado nível de exigência, margem de lucro apertada, mudanças econômicas constantes e bruscas por todo o mundo, além dos aspectos tecnológicos, motivados principalmente pela eletrificação dos veículos, que também são autônomos e conectados (PELLICCIONE *et al.*, 2017, LARRUCEA *et. al.*, 2017). Desta forma, o processo de desenvolvimento de produto precisa acompanhar essa evolução, a fim de manter a indústria automotiva robusta, eficaz e eficiente.

As próximas seções deste capítulo detalham os principais desafios para o cenário atual da indústria automotiva, que demanda uma nova metodologia para desenvolvimento de sistemas eletroeletrônicos embarcados.

2.3.1 NÚMERO DE VARIANTES E CONFIGURAÇÕES

O mercado automotivo tem como objetivo atender o maior número de clientes com um mesmo modelo de veículo. Consequentemente, o número de variantes e configurações tende a aumentar, adequando os veículos às necessidades dos clientes. O veículo pode possuir uma configuração básica e adicionar opcional, sendo que estas podem ser aplicada de acordo com a solicitação dos clientes. Essa característica tem sido um dos motivadores da complexidade do desenvolvimento dos sistemas eletroeletrônicos (BRAUM *et al.*, 2014; PRETSCHER *et al.*, 2007).

O grande número de opções contribui para a satisfação do cliente e dá mais flexibilidade para a decisão de compra. Entretanto, a combinação desses opcionais aumenta exponencialmente o esforço de desenvolvimento e validação do veículo. Um veículo pode ter um número muito grande de opcionais, e cada conjunto de opcionais aplicado ao veículo forma uma combinação diferente, que precisa ser testada para assegurar a qualidade e confiabilidade do veículo. (WEBER, 2009).

Os sistemas e componentes de um veículo podem ser representados como um conjunto de peças, conforme Figura 11. A combinação de cada componente gera uma configuração diferente no veículo, com um número muito grande de possibilidades. Esse cenário exige que o processo de desenvolvimento tenha uma estratégia para a definição e o controle das atividades de teste *software* do sistema eletroeletrônico automotivo.

Figura 11 – Exemplo de uma configuração de um veículo baseado em funções, sistemas e componentes.

Fonte: Elaborada pelo autor.

A atualização de *software* de um componente automotivo ocorre normalmente em ciclos anuais, desde a fase do desenvolvimento, em que são necessárias alterações e revisões, até depois do lançamento do veículo no mercado; principalmente para: solucionar problemas, melhorar ou igualar o veículo em função da concorrência, por leis ou por oportunidade de inovações. Quanto maior o número de versões e configurações do veículo no qual esse componente será aplicado, maior a frequência de alteração de *software*, isso também implica em diferentes versões de *software* para um mesmo modelo de veículo.

O desafio no contexto do desenvolvimento dos sistemas eletroeletrônico de um veículo, neste aspecto, está no gerenciamento das versões de *software* e *hardware*, pois vários componentes definem um sistema ou uma função. Na maioria dos casos, os componentes são responsabilidade de fabricantes diferentes, o

sincronismo das amostras precisa ser monitorado para garantir a funcionalidade correta do veículo (LARRUCEA *et. al*, 2017). O exemplo a seguir mostra um cronograma de desenvolvimento de um módulo eletrônico, detalhando as amostras de *software*.

Esse componente precisa ser sincronizado com todas as fases de testes, testes de HIL e testes no veículo, onde a integração de todos componentes é necessária para a validação das funções e sistemas. Considerando que um veículo pode ter mais de 50 módulos eletrônicos, o controle e gerenciamento de todas as versões se tornam um ponto chave para um desenvolvimento com qualidade, pois caso um dos módulos esteja com a versão de *software* errada, isso pode comprometer todo o resultado de teste e gerar um risco muito grande.

Figura 12 – Cronograma de desenvolvimento de um componente eletrônico.

Fonte: Elaborada pelo autor.

Um modelo de veículo se mantém em produção em média 10 anos, além de ser necessário manter os serviços e peças de reposição em torno de 15 anos após o fim da produção. Ao se comparar esse período com os prazos de disponibilidade de componentes eletrônicos, o desenvolvimento de sistema eletrônico automotivo se torna muito desafiador, pois muitos módulos eletrônicos são substituídos por outros durante o período de vida deste veículo no mercado. Essa substituição se deve à descontinuidade dos componentes eletrônicos que, na maioria dos casos, são

compartilhados com aplicações que possuem uma renovação mais rápida, a exemplo dos componentes de computadores e celulares.

O elevado número de versões de componentes (HW/SW) exige gerenciamento e controle dos testes de *software*. Qualquer lacuna no planejamento de testes ou falhas no controle das atividades, dentre as versões diferentes dos componentes, pode trazer consequências graves na sua funcionalidade. Atualmente, não existe na literatura um processo que englobe essa atividade de forma sistemática. Para mitigar e gerenciar os riscos de falhas do sistema eletroeletrônico, essa tese definirá uma estratégia de teste que mapeie o sistema de veículo e crie uma estrutura que organize o planejamento, o controle e a execução dos testes do sistema eletroeletrônico.

2.3.2 DESENVOLVIMENTO GLOBALIZADO

Nas últimas décadas, o desenvolvimento de novos produtos, tem representado um papel muito importante para o negócio das empresas, pois a globalização diminui o tempo de transferência de tecnologias e o modelo utilizado para desenvolver o produto pode ajudar a empresa a ser mais competitiva e eficiente, quando é necessário ter um produto inserido globalmente. Dentre as diversas razões estão a crescente internacionalização econômica, o aumento da diversidade e variedade nas ofertas de produtos, bem como a redução do ciclo de vida dos produtos no mercado (OPPAT, 2008).

Esse ambiente de crescente complexidade e competitividade global valoriza a importância da estratégia do desenvolvimento de produto (DP), para que as empresas, dentro das suas estratégias competitivas de organizações transnacionais, intensifiquem o processo de descentralização de suas estruturas de desenvolvimento de produtos (MUDAMBI *et al.*, 2007).

No Brasil, durante os últimos anos, diversos produtos foram desenvolvidos com intensiva participação da engenharia das subsidiárias brasileiras das montadoras automotivas (QUINTELLA *et al.*, 2007). Exemplos de sucesso de mercado incluem o modelo Celta e Meriva (GM), Palio (parcialmente desenvolvido

sob a responsabilidade da engenharia da Fiat Brasil), Polo Sedan (para mercados como Brasil, China e África do Sul) e o Fox da Volkswagen, este último, inclusive, exportado para a Europa. Outros estudos (CHIESA, 2000; SUBRAMANIAM, 2006; HARVEY *et al.*, 2007) mostram ainda que o desenvolvimento de novos produtos em muitas *Transnational Companies* (TNCs) apresenta um caráter cada vez mais descentralizado, com times de engenharia distribuídos entre matriz e filiais, muitas vezes trabalhando conjuntamente em projetos de novos veículos.

Existe uma diferença entre “desenvolvimento global de produtos” (DGP) e “desenvolvimento de produto global”. No primeiro caso, o termo “global” se refere à estrutura organizacional para o desenvolvimento de produto de uma organização transnacional; com unidades distribuídas em diferentes países que interagem de diferentes maneiras, e cujo resultado de um trabalho conjunto pode ser, por exemplo, aplicado em apenas um mercado específico (CHIESA, 1999). No caso de desenvolvimento de produto global, o termo “global” está associado ao produto e não à estrutura organizacional para o seu desenvolvimento, remetendo ao conceito dos anos 90 de desenvolver uma “plataforma mundial” de produto (HUMPHREY *et al.*, 2001; DIAS, 2003).

Vários estudos anteriores mostraram que a indústria automobilística tendia a centralizar suas atividades de pesquisa e desenvolvimento de produtos em poucos países. Em geral, essa centralização ocorria naqueles países onde as matrizes de suas empresas multinacionais estavam instaladas (MILLER, 1994; CALABRESE, 2001). Essa tendência é controversa em estudos subsequentes. Dias (2003) e Consoni (2004) argumentam que não há padrões claramente adotados como consenso entre as montadoras de veículos no que se refere à centralização ou descentralização do desenvolvimento de produto. Tais decisões são baseadas na estratégia da empresa e no seu “grau de controle hierárquico predominante”, entendido aqui como o grau de autonomia que as montadoras atribuem ou não às suas subsidiárias (SEGISMUNDO *et al.*, 2014).

O desafio do processo de desenvolvimento é garantir que a identidade da empresa ou da marca permanecerá, independente do local de desenvolvimento do

produto, suportando a integração, o controle e o gerenciamento globalizados, de forma sistemática.

A metodologia proposta nesta tese, auxiliará as atividades de desenvolvimento global, pois a complexidade será mapeada e dividida, com um método claro de identificação das partes, estruturando a documentação e facilitando a divisão das atividades e responsabilidades globais.

2.3.3 HETEROGENEIDADES DE SOFTWARE

Dentro do setor automotivo, as soluções de *softwares* são bastante diversificadas. Existem aplicações desde sistemas de entretenimento a sistemas de segurança crítica em tempo real. Elas podem ser agrupadas de acordo com a aplicação, associadas aos requerimentos não funcionais:

- Multimídia: telemática, interface homem-máquina, *software* tipicamente em tempo real, o qual também dispõe de sistema “off-board” de tecnologia, denominado evento discreto e processamento de dados. Destacam-se neste grupo as novas tecnologias dos aplicativos dos computadores portáteis e celulares, que possuem interface com o veículo.

- Conforto e carroceria: tipicamente *software* de tempo real brando, baseadas em eventos discretos de programas de controle.

- Sistema de segurança crítica: tempo real crítico, baseadas em eventos discretos, requerimentos críticos de segurança.

- Trem de força e controle do chassi: tempo real crítico, algoritmos de sistemas de controle com processamento de eventos discretos com rigorosos requerimentos.

- SW de Infraestrutura: base branda e crítico de tempo real, com *software* baseado em eventos de atualização e de gestão com diagnose.

Adicionalmente, o veículo passou a integrar algumas tecnologias: os telefones celulares, “tablets”, computadores portáteis, dentre outros, interconectados e utilizando conexão via cabo ou sem fio, tecnologia de rede Bluetooth ou Wifi. Elas possibilitam a instalação de vários aplicativos para compartilhar conteúdos de entretenimento, diagnósticos remotos, atualização de *softwares*, controle da localização do veículo, sistemas de emergência, comunicação entre veículos e formação de comboio.

Seguindo a tradição de encontrar soluções, a indústria automotiva desenvolve em grande parte as suas próprias abordagens, e não é diferente quando se trata do domínio de *software* (BLOOS, 1999).

É grande a quantidade de tecnologia proprietária de *software* nos automóveis. Isto se aplica ao sistema operacional, protocolos de comunicação, ferramentas, arquitetura. Essencialmente, a engenharia de *software* automotivo tem suas necessidades específicas de domínio próprio, mas as experiências de outras áreas podem servir como base para a indústria automotiva, que se beneficiaria de experiências existentes e tecnologias de outros domínios. Em particular, as áreas de telecomunicações e aviônicos, pois a gestão do ciclo de *software* em veículos está em sua fase inicial.

Muitos fornecedores, mesmo alguns OEMs, não possuem nível de maturidade elevada para desenvolvimento de *software*, o que é bastante crítico, considerando que os sistemas são desenvolvidos de forma distribuída e o *software* é altamente complexo, multifuncional, distribuído, com tempo e segurança crítica (BROY, 2006).

A reutilização de soluções de *software* é pouco aplicada dentro da indústria automotiva (HARDUNG *et al.*, 2004, LARRUCEA *et. al*, 2017). Em muitos casos, a reutilização de parte do *software* de uma geração de um carro para outro representa um valor aproximado de 10%, enquanto mais de 90% do *software* é simplesmente reescrito. A principal razão é a utilização de *hardware* específico, o que faz com que seja difícil mudar ou adaptar o código existente. A quantidade de automatização na produção de *software* para o automóvel é bastante baixa. Ferramentas são usadas

apenas de forma isolada. Não há nem um fluxo de projeto definido, nem uma estrutura definida para desenvolvimento de funções distribuídas (BROY, 2006).

A identificação e estruturação da estratégia de teste proposta nesta tese, criará uma base para facilitar a documentação e reutilização das atividades de teste, o que pode ser utilizado com base para novos desenvolvimentos, facilitando a reutilização de conceitos e dados das funcionalidades desenvolvidas.

De acordo com Pretschner, do ponto de vista de *software* e engenharia de sistemas integrados, existe a necessidade de desenvolvimento de competências de várias disciplinas (PRETSCHNER *et al.*, 2007):

- Os sistemas que contém *Software / Hardware* consistem em um grande número de funções e processos que trocam informações sobre links de comunicação separados. Estes sistemas apresentam todos os detalhes e complexidades de um computador distribuído, em que são necessárias habilidades da ciência e engenharia da computação para construir sistemas automotivos com sucesso.

- Estes sistemas são conectados a sensores e atuadores que medem valores físicos e operam processos físicos, em tempo real. Os sistemas de SW / HW precisam responder a essas entradas em um clássico sistema de controle, o conhecimento da teoria de controle é necessário.

- Para algumas funções do veículo, o conhecimento multidisciplinar é necessário, principalmente pela interfaces entre os sistemas de diferentes bases de tecnológicas. Historicamente, tanto os métodos como os modelos de teoria de controle têm sido bastante diferentes dos modelos de processamento de informação. Na teoria de controle, ferramentas tais como Matlab / Simulink (MATHWORK, 2016; WEEKS, 1995) são usadas para modelar equações diferenciais. Em contraste, os engenheiros de processamento de dados estão cada vez mais utilizando modelos de dados e comportamento, expressos em “Unified Modeling Language” (UML), e outros tipos de fluxo de dados discretos ou modelos de máquina de estado em

desenvolvimento de automóvel. A junção de conceitos de várias áreas da engenharia é necessária.

A grande heterogeneidade de domínios e a falta de uma ampla aceitação de padronização justificam um conjunto de modelos e abordagens de desenvolvimento com diferentes ferramentas em uso para a construção do *software* automotivo. Uma vez que as ferramentas são desenvolvidas por diferentes fornecedores, não existe um sistema modelo abrangente e, conseqüentemente, as ferramentas também não são integradas. Existem tentativas para a criação de cadeias de ferramentas pragmáticas, ligando as ferramentas em um formulário em que os modelos de dados de uma ferramenta são exportados e importados pela ferramenta seguinte (PREE *et al.*, 2006; KARSAI *et al.*, 2003; AHLUWALIA *et al.*, 2005).

2.3.4 DISTRIBUIÇÃO DAS ATIVIDADES DE DESENVOLVIMENTO DE “SOFTWARE”

A indústria automotiva tem sido tradicionalmente organizada de uma maneira altamente vertical, em que engenheiros mecânicos trabalharam muito por mais de um século para tornar os vários subsistemas em carros independentes. Isso facilitou o desenvolvimento independente e a produção das peças e deu origem a uma divisão altamente bem-sucedida do trabalho. Na atualidade, cerca de 30% do valor do produto é criado pelos OEMs que tendem a concentrar-se no motor, integração, design e comercialização da marca (PRETSCHNER *et al.*, 2007). Há décadas, o modelo "ideal" de desenvolvimento do automóvel tem sido realizar a produção de suas peças por uma cadeia de fornecedores e a montagem pelo fabricante final do veículo. Assim, uma grande parte das atividades de engenharia e produção é terceirizada, o que também facilita a otimização do custo e distribuição de risco.

Com o *software* se tornando um importante componente de inovação, as responsabilidades do OEM têm evoluído desde a montagem de peças à integração do sistema. Funções que eram tradicionalmente isoladas dentro do veículo passaram a interagir com outros sistemas do veículo (MARO *et al.*, 2018). Dentro desse cenário, a atividade de integração e testes de *software* se torna um grande desafio.

O desenvolvimento de um veículo é uma tarefa bastante complexa e demanda mão de obra especializada para execução de diversas atividades. Para facilitar o direcionamento da distribuição das atividades, as mesmas são divididas em "core", atividades exclusivas das empresas que trazem um diferencial no mercado e exigem confidencialidade, ou as atividades não "core", atividades mais gerais que podemos encontrar através de empresas especializadas (fornecedores), podem ser chamadas de atividades ou componentes de prateleira.

Na indústria automotiva, uma das grandes vantagens é que os fornecedores principais possuem vínculo com as empresas por serem consideradas *benchmark* nos tópicos que trabalham e são fornecedores das principais indústrias automotivas. Com isso, é possível usar sinergias no desenvolvimento e produção, uma vez que eles geralmente produzem sistemas semelhantes para diferentes OEMs, compartilhando também parte do custo, gerando baixo custo por unidade para os OEMs, embora apenas para funções que não são confidenciais.

Um dos principais efeitos negativos dessa distribuição do trabalho com fornecedores e parceiros é que os complexos processos distribuídos precisam ser coordenados e requerem gerenciamento através de um processo e metodologia bastante robustos, que favoreçam a transparência de cada um dos envolvidos na etapa de desenvolvimento, além da sua integração.

2.3.5 REQUERIMENTOS DE SEGURANÇA

O automóvel moderno está definitivamente com sistemas intensivamente controlados por *software*, sendo chamado de computador sobre rodas. O *software* controla hoje um crescente número de funções tradicionais que outrora eram controladas pelo motorista. Grande parte dessas funções está relacionada a sistemas de segurança, motivado pela eletrificação dos veículos e pelos veículos autônomos e conectados (PELLICCIONE *et al.*, 2017).

A normatização da segurança especificada na área automotiva é definida em ISO 26262 (ISO 26262, 2011), derivada da norma industrial IEC 61508 (IEC, 1999),

a parte 6 dessa norma é especificamente direcionada ao desenvolvimento de *software*. Requerimentos de segurança para *software* também são definidos dentro da padronização Autosar (AUTOSAR, 2018). As normas não descrevem as técnicas de aplicação de uma solução, mas trazem disciplina e gerenciamento das adaptações na estrutura do projeto de sistemas com *software* que afetam a segurança (PANARONI *et al.*, 2008).

2.3.5.1 *Safeware* Automotivo

Software relacionado com sistemas de segurança é chamado de *safeware*, para diferenciar de um *software* comum (LEVESON, 1995). Conceitualmente, o *software* não tem influência direta nos riscos, mas ao mesmo tempo pode ser considerado, em muitos casos, responsável pela segurança em nível de sistema. O *Software* tem poder para comandar através de atuadores os elementos do sistema e pode se tornar a maior fonte de risco do sistema global (PANARONI *et al.*, 2008), conforme mostrado na Figura 13.

Figura 13 – Sequência de causas onde o *software* é uma potencial fonte de riscos.

Fonte: (PANARONI *et al.*, 2008).

O *Safeware* requer uma combinação de: um extremo e rigoroso processo de verificação e validação e uma série de medidas de segurança no interior do produto (PANARONI *et al.*, 2008). Existem diversos padrões no mercado que estabelecem regras e orientações para desenvolvimento dos projetos em relação à segurança funcional:

- IEC 61508 Parte 3 (genérica e não específica para automóvel) (IEC 61508, 1999).

- Diretrizes de Integridade Misra (menos conhecida) (MISRA-C, 1998).
- MISRA C: 2004 (muito utilizada, com integração em algumas ferramentas) (MISRA-C, 2004).
- MIL STD 882D, Requisitos de programa de segurança de sistemas (MIL-STD-882D, 2000).
- DO 178B, Considerações de *software* em sistemas de bordo e equipamentos de certificação (DO-178B, 1992).

2.3.5.2 Segurança funcional em eletrônica automotiva

Atualmente, a segurança funcional em eletrônica automotiva é normatizada pela ISO 26262, que integra o processo de desenvolvimento de produto automotivo, desde a fase de especificação, passando pelo projeto, integração, verificação, validação e liberação para produção. Essa norma foi adaptada a partir da norma de segurança funcional IEC 61508 para os sistemas automotivos. Ela define a segurança funcional para equipamentos aplicados em produtos eletroeletrônicos automotivos relacionados a sistemas de segurança. Sua principal meta é abordar os possíveis riscos que esses sistemas possam causar por mau funcionamento.

Embora intitulada “segurança funcional de veículos rodoviários”, a norma está somente relacionada com a segurança funcional de sistemas eletroeletrônicos e não inclui os sistemas mecânicos. A norma é baseada no risco, em que situações operacionais perigosas são qualitativamente avaliadas e medidas de segurança são definidas, para evitar ou controlar falhas sistemáticas, detectar ou controlar falhas aleatórias de *hardware*, ou mitigar seus efeitos.

A ISO 26262 propicia um ciclo de desenvolvimento automotivo seguro (gestão, desenvolvimento, produção, operação, manutenção e desativação) e suporta a adaptação das atividades necessárias durante todas as fases; abrange

aspectos funcionais de segurança de todo o processo de desenvolvimento, incluindo atividades como a especificação de requisitos, projeto, implementação, integração, verificação, validação e configuração. Ela fornece uma abordagem especificamente automotiva baseada na avaliação das classes de risco (Níveis de Integridade de Segurança Automotiva, ASILS). De acordo com a classificação ASILS os requisitos de segurança são especificados para alcançar um risco residual aceitável. Essa especificação é utilizada no processo de teste do sistema, incluindo a validação do software.

As dez partes da ISO 26262 são: Parte 1 - vocabulário; Parte 2 - gerenciamento da segurança funcional; Parte 3 - fase conceitual; Parte 4 - nível de sistema no desenvolvimento de produto; Parte 5 - nível de *hardware* no desenvolvimento de produto; Parte 6 - nível de *software* no desenvolvimento de produto; Parte 7 – operação e produção; Parte 8 – processo de suporte; Parte 9 - nível de integridade de segurança automotiva (ASIL) - análise orientada; Parte 10 – Orientação ISO 26262.

O desafio principal para o tópico de segurança funcional é adequar o processo de desenvolvimento, de modo a garantir que as atividades do projeto incluam de forma sistemática o assunto, trazendo ao projeto transparência e rastreabilidade.

2.3.5.3 ASIL – Nível de integridade da segurança automotiva, processo de avaliação.

O trabalho sobre o nível de segurança dos sistemas se inicia com a análise de perigos e a avaliação dos riscos, resultando em avaliação de ASIL para todos os eventos perigosos identificados no foco da segurança. Cada evento perigoso é classificado de acordo com a gravidade (S) da lesão que pode ser gerada. Classificações de Gravidade (S):

S0 - sem lesão;

S1 - lesão leve a moderada;

S2 - lesão severa com risco de vida (sobrevivência provável)

S3 – lesão fatal à vida (sobrevivência incerta)

A Gestão de Riscos reconhece que a análise da gravidade de uma possível lesão é modificada pela forma provável de como o prejuízo pode acontecer, ou seja, para um dado risco, um evento perigoso é considerado um risco menor quando a probabilidade de acontecer é menor. Dentro do processo de avaliação de análise de perigos e riscos desta norma, a probabilidade de um risco de prejuízo é ainda classificada de acordo com uma combinação de exposição (E), a frequência esperada relativa das condições operacionais em que a lesão pode possivelmente acontecer, e controle (C), a probabilidade relativa de que o controlador pode agir para evitar que a lesão. Classificação da exposição (E):

E0 - incrivelmente improvável

E1 – muito baixa probabilidade (lesão pode acontecer apenas em condições de funcionamento raras)

E2 - baixa probabilidade

E3 – media probabilidade

E4 - alta probabilidade (lesão pode acontecer na maioria das condições operacionais)

Classificação da controlabilidade (C):

C0 - controlável

C1 - simplesmente controlável

C2 - normalmente controlável (a maioria dos motoristas poderia agir para evitar lesões)

C3 - difícil de controlar ou incontrolável

Em termos de classificações, uma "segurança automotiva de nível de integridade D" de eventos perigosos (abreviado "ASIL D") é definida como um evento que tem possibilidade razoável de causar um risco de vida (sobrevivência incerta) ou lesão fatal, com possível lesão física na maioria das condições de funcionamento, e com pouca chance do motorista fazer algo para evitar a lesão, ou seja, ASIL D é a combinação das classificações S3, E4 e C3. Por exemplo, uma

incontrolável (C3) lesão fatal (S3) de perigo hipotético poderia ser classificada como ASIL A se o perigo tem uma probabilidade muito baixa (E1). O nível ASIL mais baixo é o nível QM. QM refere-se à consideração do padrão que abaixo ASIL A, não há nenhuma relevância para segurança e são necessários apenas os processos de gestão da qualidade (ISO 26262, 2011).

A classificação de gravidade, exposição e definições de controle são de caráter informativo, não prescritivo e, efetivamente, deixa espaço para avaliação subjetiva ou diferentes critérios entre os fabricantes de automóveis e fornecedores de componentes (WARD *et al.*, 2013; VAN, 2012). Em resposta a isso, a SAE (*Society for Automotive Safety Engineers*) elaborou a J2980 - Considerações sobre a ISO26262 ASIL “*Hazard Classification*” para fornecer orientação mais explícita para a avaliação da exposição, gravidade e controlabilidade para um determinado perigo (J2980, 2015).

O método ASIL é um desenvolvimento relativamente recente. As discussões sobre ASIL costumam comparar seus níveis com aqueles definidos em outros sistemas de segurança ou de gestão de qualidade bem estabelecidos (conforme Tabela 1). Em particular, o ASIL é comparado com os níveis de redução de risco SIL definidos na norma IEC 61508 e os níveis de projeto utilizados no contexto de DO-178C e DO-254.

Tabela 1 – Comparativo da classificação do nível de segurança.

Comparativo do ASIL					
Norma	Níveis específicos de segurança para cada Norma				
Automotivo (ISO 26262)	QM	ASIL-A	ASIL-B/C	ASIL-D	-
Geral (IEC-61508)	(SIL-0)	SIL-1	SIL-2	SIL-3	SIL-4
Aviação (DO-178/254)	DAL-E	DAL-D	DAL-C	DAL-B	DAL-A
Ferrovia (CENELEC 50126/128/129)	(SIL-0)	SIL-1	SIL-2	SIL-3	SIL-4

2.3.6 GESTÃO DO CONHECIMENTO

Conhecimento tem sido descrito como "um estado ou fato de saber", uma condição de "compreensão adquirida através da experiência ou estudo; a soma ou a gama do que foi percebido, descoberto ou aprendido "(SCHUBERT *et al.*, 1998). Conhecimento é definido como uma crença justificada que aumenta a capacidade de uma entidade para uma ação efetiva (HUBER, 1991; NONAKA *et al.*, 1995). A gestão do conhecimento refere-se a identificar e alavancar o conhecimento coletivo de uma organização para ajudar a organização a ser competitiva (VON KROGH, 1998). A gestão do conhecimento existe para aumentar a capacidade de inovação e resposta (HACKBARTH, 1998).

A memória organizacional inclui o conhecimento residente em vários componentes e formas, incluindo a documentação escrita, informações estruturadas e armazenadas em bancos de dados eletrônicos, conhecimento humano armazenado em sistemas especialistas, procedimentos e processos organizacionais documentados e conhecimento tácito adquirido por indivíduos e redes de indivíduos (TAN, 1999).

Empresas que têm um processo de desenvolvimento de produto, que com foco na gestão, controle e armazenamento do conhecimento, maximizam a utilização de seus recursos, reduzem o tempo gasto em desenvolvimento e validação e minimizam os riscos em novos projetos (GUIMARÃES, 2011).

Para a maioria dos gestores, é difícil perceber claramente os benefícios práticos do mapeamento do conhecimento sobre os processos da empresa. Entretanto, às vezes, é mais fácil perceber o custo de não ter este conhecimento registrado (FILHO, 2000).

Dentro dos processos de desenvolvimento de um sistema eletroeletrônico automotivo, todas as etapas exigem um elevado nível de integração e conhecimento para serem realizadas. O modelo de integração e o conhecimento, na maioria das vezes, estão nos profissionais e nas empresas fornecedoras e não descritos nos processos de desenvolvimento do veículo. Com o aumento da velocidade das

inovações tecnológicas, a integração e o conhecimento se tornaram as variáveis que mais influenciam no resultado final, mesmo para processos bastante conhecidos, nos quais o modelo de integração do conhecimento já está estabelecido. Para isso, é necessário avaliar e definir novos métodos e ferramentas de gerenciamento de projetos, em que a integração e o conhecimento sejam abordados como partes do processo, documentando cada uma das etapas de forma padronizada e organizada para que o conhecimento e a integração sejam registrados, armazenados e controlados.

Os sistemas eletrônicos possuem alguns fatores desafiadores, como nível de complexidade da atividade de desenvolvimento, grande número de participantes envolvidos no desenvolvimento, necessidade de mão de obra especializada, reutilização de atividades de projetos prévios, exigências normativas, dentre outros. A gestão do conhecimento deve ser abordada de forma sistemática e sempre focar nos desafios, trazendo clareza e robustez ao processo de desenvolvimento de sistemas eletroeletrônicos. A estruturação proposta nesta tese, contribuirá com a organização e gestão do conhecimento oriundo do desenvolvimento do produto.

2.3.7 INTEGRAÇÃO DOS SISTEMAS E SUBSISTEMAS

O grande volume de *software* em sistemas eletrônicos dos veículos desafia a indústria automotiva a adaptar o modelo de desenvolvimento de produto, principalmente devido à interdependência dos sistemas, componentes e funções. O processo para desenvolvimento de *software* possui diferenças se comparado ao modelo tradicional de desenvolvimento da indústria de automóvel.

O processo de desenvolvimento tradicional divide o veículo em sistemas independentes, de modo que os carros poderiam ser desenvolvidos e produzidos de uma forma altamente modular (BROY, 2006). Com a introdução de funções baseadas em *software*, esta independência desapareceu, quase todas as funções possuem diversas interações dentro do sistema do veículo, incluído muitos componentes. Isso tornou o tema de integração um desafio para o desenvolvimento de produto.

A eletrônica embarcada revolucionou os sistemas do veículo, não só pelo fato de substituir atuadores mecânicos antes baseados em relés por módulos e semicondutores, mas também por promover a integração dos diversos sistemas veiculares, distribuindo e compartilhando o processamento de dados através de redes seriais (MILLER, 1995). De acordo com Hans-Georg Frischkorn (2005), esta integração depende de dois fatores cruciais: o processo desenvolvimento dos sistemas e a padronização e utilização de arquiteturas abertas, modulares e hierarquicamente estruturadas.

O primeiro diz respeito à cooperação entre fornecedores e montadoras, através do trabalho conjunto e coordenado entre as partes. O segundo refere-se à cooperação de todo o segmento automotivo, englobando a padronização de interfaces e protocolos, além de sistemas e funções básicos, que proporcionarão um aumento na reutilização de componentes já consagrados, diminuição no tempo de desenvolvimento de *software* e *hardware* e permitirão a verificação e validação de suas funções desde as primeiras fases do desenvolvimento do projeto.

Com isso, atualizações e melhorias de *software* serão mais simples e de fácil implementação, garantindo também a compatibilidade entre fornecedores e montadoras. Ainda que pareça contraditório, esta padronização irá possibilitar uma maior diferenciação entre aplicações e funções com participação de *software*. Isso através de calibrações e desenvolvimento de funções distintas, porém baseadas em interfaces de *hardware* e *software* (sistemas operacionais) comuns, da mesma forma que ocorre atualmente no mercado de informática (FRISCHKORN, 2005).

2.3.8 TEMPO DE DESENVOLVIMENTO

Existe uma competição acirrada entre as empresas automotivas em busca de espaço maior no mercado. Tal competição se manifesta na adoção de várias estratégias de diferenciação: novos modelos de veículos, novas tecnologias, melhor atendimento ao cliente, dentre outros.

O tempo é um fator muito importante para o processo de desenvolvimento e pode variar dependendo do modelo aplicado. O modelo clássico de desenvolvimento

de produto não possui uma clareza na integração de sistemas eletrônicos, o que deixa as definições das etapas muito abertas e acaba gerando surpresas durante o processo de desenvolvimento. Desta forma, é necessário modificar os prazos de algumas etapas, normalmente aumentando o prazo final de desenvolvimento, justamente pela ausência de um planejamento antecipando, trazendo riscos ao projeto.

Em desenvolvimento de produtos complexos, existem diversas atividades que precisam ser trabalhadas para chegar ao produto final, isso inclui a participação de várias pessoas e empresas. A falta de informações, direção, metodologias e processos torna o trabalho lento, pois cada participante pode adotar um caminho diferente, duvidoso, que talvez não seja adequado e rápido. Ao invés de trabalhar no desenvolvimento do produto, o grupo inicia o desenvolvimento de um processo. A proposta deste trabalho é trazer uma metodologia que sistematize a estratégia e planejamento dos testes, estabelecendo os passos de cada atividade, definido com precisão o tempo necessário para execução das atividades de testes de *software* e reduzindo duplicidades e lacunas que possam existir, aprimorando, desta forma, o tempo de desenvolvimento.

2.3.9 GERENCIAMENTO DOS TESTES PARA VALIDAÇÃO DO SISTEMA ELÉTRICO-ELETRÔNICO

O projeto de um veículo exige uma validação rigorosa, principalmente pela grande exposição ao risco, necessitando de um período grande em relação ao projeto como um todo. Normalmente o gargalo em relação a tempo dos projetos está na área de testes.

Podem-se classificar os testes em três categorias: teste funcional, de durabilidade e estrutural. O teste funcional valida a funcionalidade do veículo, nele, todas as funções do veículo são de alguma forma checadas. O teste de durabilidade verifica e valida o comportamento do veículo no longo prazo. O teste estrutural valida a estrutura mecânica do veículo, normalmente em condições extremas para acelerar a velocidade do resultado.

Os sistemas eletroeletrônicos passam por algumas etapas preliminares até o teste final no veículo. A primeira etapa inicia-se com os testes individuais dos componentes, depois, testes do sistema ou subsistema e, ainda, um teste de sistema elétrico completo do veículo, conforme mostrado na Figura 14.

Figura 14 – Níveis de testes do sistema eletroeletrônico de um veículo.

Fonte: Elaborada pelo autor.

Existem diversos desafios para gerenciamento dos testes do projeto, o grande número de funções, de sistemas e de componentes, de versões de *hardware* e *software* dificulta essa atividade. A proposta deste trabalho será focada nos testes funcionais de *software* do veículo, para definir quais os níveis de testes que devem ser aplicados para cada função do veículo.

2.4 TESTE DE SOFTWARE EM SISTEMAS EMBARCADOS AUTOMOTIVOS

2.4.1 DIMENSÕES DE SISTEMAS EMBARCADOS AUTOMOTIVOS

Na indústria automotiva, o modelo V é amplamente utilizado no processo de engenharia de desenvolvimento de sistema de *software* (AWÉDIKIAN *et al.*, 2014), principalmente nos sistemas embarcados (PÉREZ *et al.*, 2010). Dedicase a descrever o processo de teste porque o vincula de maneira adequada às fases do desenvolvimento do sistema. Spillner *et al.* (2007), Schäuffele *et al.* (2006) e Gruszczynski (2006) afirmam que o modelo V é referência como modelo de ciclo de vida para o desenvolvimento e teste dos sistemas embarcados no campo automotivo. Os níveis exatos de teste usados diferenciam-se ligeiramente em cada projeto, e ainda entre diferentes organizações (MYERS *et al.*, 2011).

Nesta tese, nas atividades de teste, o modelo V (Figura 15) utilizado foca nos níveis mais altos do lado direito, a partir do nível componente, que inclui o teste de

software já integrado com o *hardware*. Ao todo, serão abordados quatro níveis de teste, começando na parte inferior, com o teste de componente, teste de sistema, teste de veículo e terminando no topo, com o teste de aceitação.

Figura 15 – Modelo V do sistema embarcado automotivo.

Fonte: Elaborada pelo autor.

Devido às diferentes aplicações dos sistemas embarcados, é difícil ter apenas uma única descrição de classificação de teste que represente todas as possibilidades do sistema. No entanto, com base na literatura (NEUKIRCHEN, 2004; ZANDER, 2009; DAI, 2006), Zander *et al.* (2017) elaborou uma seleção e compilação dessas possibilidades, resumidas e apresentadas na Figura 16.

Figura 16 – Dimensões do teste de sistemas embarcados.

Fonte: Adaptada de ZANDER *et al.* (2009).

2.4.1.1 Objetivo do teste

Ao desenvolver um produto, o teste é dividido em duas atividades críticas, validação e verificação (HOOD *et al.*, 2007). Essas atividades garantirão que o que está sendo desenvolvido atenda às expectativas dos clientes. Há uma diferença entre validação e verificação, a validação investiga se o que está sendo desenvolvido atende às necessidades do cliente final do produto; a verificação, se o produto atende ao que foi especificado. Essa diferença fica clara de acordo com os pressupostos de BOEHM (1979), usando essas duas questões:

- Estamos construindo o produto certo? (Validação)
- Estamos construindo certo o produto? (Verificação)

O teste de *software* dentro do escopo da validação e verificação é testado com objetivos diferentes. Esses objetivos podem ser categorizados como: testes estáticos, também chamados de revisão, que não envolvem a execução do *software*; e testes dinâmicos, são aqueles relacionados à execução e se dividem em três tipos: estrutural, funcional, e não funcional (ZANKER *et al.*, 2017).

Teste estático: é geralmente definido como o processo de encontrar os erros, falhas e defeitos. Erros em um programa podem ser revelados sem execução, apenas examinando seu código fonte (INTERNATIONAL SOFTWARE QUALIFICATION BOARD, 2006). Da mesma forma, outros artefatos de desenvolvimento podem ser revisados (por exemplo, requisitos, modelos ou especificação de teste) (ZANDER *et al.*, 2017). No domínio automotivo, o teste estático é executado em um nível inferior ao nível do componente, ou na fase de especificação, que não é o foco desta tese.

O teste dinâmico, em contraste ao teste estático, baseia-se na execução, descritos abaixo, seus tipos são:

Teste estrutural: cobrem a estrutura do SST (*Software Sob Teste*), durante a execução do teste (por exemplo, controle ou fluxo de dados). Para isso, a estrutura interna do sistema do *software* (por exemplo, código ou modelo) precisa ser conhecida. Esses testes também são chamados de testes de caixa branca ou caixa de vidro (MYERS, 2011; INTERNATIONAL SOFTWARE QUALIFICATION BOARD, 2006).

Teste funcional: está relacionado com a avaliação do comportamento funcional de um SST em relação aos requisitos funcionais. Ao contrário dos testes estruturais, não exigem nenhum conhecimento sobre a estrutura interna do sistema do *software*. Estes são, portanto, chamados de testes de caixa preta (BEIZER, 1995). Nesta categoria, testes de segurança funcional também estão incluídos. Atualmente, os testes de segurança funcional são apenas uma pequena parte dos testes de *software* na área automotiva. Com a introdução de normas de segurança, como a IEC 61508 e a ISO26262, o significado dos testes de segurança de *software* aumentará consideravelmente nos próximos anos.

Teste não funcional: semelhante aos testes funcionais, os testes não funcionais são executados em relação à especificação de requisitos do sistema. Em contraste com o teste funcional puro, ele visa a avaliação de requisitos não funcionais, como confiabilidade, carga ou desempenho. Eles são geralmente testes de caixa preta. No entanto, para recuperar certas informações, é necessário acesso interno durante a execução do teste. Por exemplo, durante o teste de robustez, o sistema é testado com dados de entrada inválidos que estão fora dos intervalos permitidos para verificar se o sistema ainda está seguro e funcionando corretamente. Como regra geral, a robustez é assegurada por verificações dedicadas de plausibilidade integradas no *software* automotivo (HOOD *et al.*, 2007).

O *Black Box Testing* é uma técnica de teste sem referência à estrutura interna do componente ou sistema. Nela, não é necessário um bom conhecimento de programação, uma vez que se examina apenas os aspectos fundamentais do sistema, sem entrar em detalhes. (KHAN *et al.*, 2012) *White Box Testing* é uma técnica de teste baseada na estrutura interna do componente ou sistema. Nesse caso, é necessário um bom conhecimento de programação, para entender melhor o código fonte.

As pesquisas disponibilizadas na literatura, relacionadas ao campo automotivo, indicam que as contribuições nessa área são dominadas por métodos desenvolvidos para fins de testes e verificações baseados em modelos de baixo nível (SUNDMARK *et al.* 2011; CONRAD, 2004; AWÉDIKIAN *et al.*, 2014; COLLOFELLO *et al.*, 2008). Esta tese terá como foco os testes dos níveis mais altos, começando do componente, com *software* e *hardware* integrados, até o nível de aceitação.

2.4.1.2 Plataformas de teste

A execução do teste é gerenciada pelas chamadas plataformas de teste. A finalidade da plataforma de teste é estimular o objeto de teste com entradas e observar e analisar as saídas do sistema em teste (ST). No domínio automotivo, a plataforma de teste clássica é um carro com um motorista de teste. O motorista determina as entradas do ST através de um cenário de condução e observa a

reação do carro, suportada por diagnóstico especial e medição de *hardware/software*, que registra os dados do teste e permite que o comportamento seja analisado *off-line*.

Além do fato de o veículo ser uma plataforma muito cara, para realizar todos os testes, também existem testes de baixo nível que dificilmente podem ser executados em um veículo. Dentro da definição da estratégia de teste e planejamento, uma plataforma de teste apropriada deve ser escolhida dependendo do objeto, da finalidade e do ambiente de teste (ZANKER *et al.*, 2017).

Model-in-the-Loop (MiL): o primeiro nível de integração, MiL, é baseado no modelo do próprio sistema. Nesta plataforma, o ST é um modelo funcional ou modelo de implementação, testado em malha aberta, sem qualquer modelo de planta em primeiro lugar, ou em teste de malha, fechada com um modelo de planta, ou seja, sem *hardware* físico (SCHÄUFFELE *et al.*, 2006; KAMGA *et al.*, 2007; LEHMANN *et al.*, 2008).

Software-in-the-Loop (SiL): durante o SiL, o ST é testado por *software* em uma configuração de malha fechada ou de malha aberta. Os componentes de *software* em testes são normalmente implementados em linguagem C, e são programados manualmente, ou gerados automaticamente por códigos baseados em modelos. A finalidade do teste no SiL é principalmente o teste funcional (KAMGA *et al.*, 2007). Se o *software* for construído para uma arquitetura de ponto fixo, a escala necessária já faz parte do *software*.

Processor-in-the-Loop (PiL): no PiL, controladores embarcados são integrados em dispositivos embarcados com *hardware* proprietário (ou seja, ECU). O teste no nível do PiL é semelhante aos testes do SiL, mas o *software* incorporado é executado em uma placa, com o processador de destino ou em um emulador do processador de destino. Os testes no nível do PiL são importantes porque podem revelar falhas causadas pelo compilador de destino ou pela arquitetura do processador. É o último nível de integração que permite a depuração durante os testes de uma maneira barata e gerenciável (LEHMANN *et al.*, 2008). Desse modo, o esforço gasto pelo teste PiL vale a pena em quase todos os casos.

Hardware-in-the-Loop (HiL): ao testar o sistema embarcado no nível HiL, o *software* é executado na ECU final. No entanto, o ambiente em torno da ECU ainda é simulado. A ECU e o ambiente interagem através dos conectores elétricos digitais e analógicos. O objetivo do teste é revelar falhas nos serviços de baixo nível da ECU e nos serviços de I/O (*Input and output*) (SCHÄUFFELE *et al.*, 2006). Além disso, os testes finais dos componentes, normalmente realizados e entregues pelo fornecedor, são executados no nível HiL porque o próprio componente é a ECU integrada (KAMGA *et al.*, 2007). O teste requer um comportamento em tempo real do modelo de ambiente para garantir que a comunicação com a ECU seja a mesma da aplicação real. O HiL pode ser apenas para teste de um ECU (sub-sistema) ou com interação de várias ECU de um sistema ou sistemas, ou ainda um HiL completo, simulando o veículo inteiro (todo o sistema do veículo).

Veículo: finalmente, um dos últimos níveis de integração é obviamente o próprio carro. O teste da ECU é executado no carro real, que pode ser um protótipo ou um carro da linha de produção. No entanto, esses testes, realizados apenas em fases posteriores de desenvolvimento, são caros e não permitem que os parâmetros de configuração variem arbitrariamente (LEHMANN *et al.*, 2008). Neste nível de teste, falhas de *hardware* dos componentes são difíceis de serem diagnosticadas porque os sinais internos não estão mais acessíveis (KAMGA *et al.*, 2007).

Devido à complexidade do sistema automotivo e às múltiplas características de cada função, existem diversas possibilidades para testar as funções dos veículos em diferentes combinações de plataformas de teste. O alvo deve ser escolher o cenário de maior eficiência, onde o equilíbrio entre custo, tempo e qualidade é alcançado.

2.4.1.3 Níveis de teste ou escopo de teste

Conforme descrito no início desta sessão, os níveis de teste são divididos de acordo com o modelo V, essas etapas estão vinculadas às especificações do projeto.

Devido à composição do sistema, os testes em diferentes escopos podem revelar diferentes falhas (WEYUKER, 1988; INTERNATIONAL SOFTWARE TESTING QUALIFICATION BOARD, 2006). Na Figura 15, os níveis de teste são divididos em

componente, sistema, veículo e aceitação, com foco na aplicação automotiva; na Figura 16, os testes são divididos em integração de componentes e sistema, modelo este aplicado de forma geral a sistemas embarcados.

Nível de componente: representa o teste completo individual da ECU, onde o *hardware* é integrado com o *software*, e a ECU é testada como uma caixa preta, olhando para as interfaces, com suas interfaces I/O e os barramentos de comunicação de dados, principalmente a rede CAN (*Controller Area Network*) (conforme Figura 17), um dos principais protocolos utilizados na indústria automotiva, conforme Figura 17. Nesse nível, esta tese considerará que o *software* já foi gradualmente integrado aos componentes de *hardware*, e os testes básicos de integração de interoperabilidade foram realizados com sucesso. O objetivo deste nível é testar o sistema de ECU completo e suas interfaces.

Figura 17 – Partes do nível de teste do componente do sistema

Fonte: Elaborada pelo autor.

2.4.2 DESAFIOS PARA TESTES DE SOFTWARE NA INDÚSTRIA AUTOMOTIVA

Conforme discutido na sessão anterior, o modelo V padrão é amplamente utilizado na indústria automobilística para os processos de engenharia de desenvolvimento de *software* de sistema (AWÉDIKIAN *et al.*, 2014). Assim, pode-se inferir que o *software* do sistema automotivo é testado em diversos níveis de

abstração, desde do nível de código fonte até o mais alto nível de integração no veículo, utilizando diferentes ambientes, com diferentes objetivos de teste e diferentes conjuntos de requisitos (CHUNDURI *et al.*, 2016). Os níveis exatos de teste do modelo V são ligeiramente diferentes de um projeto para outro dentro de cada empresa automotiva (KASOJU *et al.*, 2013).

Devido ao conceito modular do desenvolvimento e teste de *software* dos sistemas embarcados, suas atividades são divididas por equipes e departamentos muitas vezes diferentes, antes de realizar o teste de integração (BROY *et al.*, 2007). Tais equipes são as principais responsáveis pelo desenvolvimento em baixo nível e teste do *software*, independente da integração. Após a etapa de teste do *software*, o teste de integração de sistema é realizado na ECU correspondente. Este teste é normalmente realizado como teste de *Hardware* no *Loop* (HIL), onde o sistema é executado em ambientes simulados (BROY, 2006). Nesta mesma sequência, as funções distribuídas em vários componentes ou sistemas do veículo, sejam elas desenvolvidas e testadas pela própria empresa ou por fornecedores, passam pelos testes de integração no nível de teste do sistema, ou do veículo correspondente.

No nível de teste de integração de veículos, todos os sistemas de ECUs e de sistemas correspondentes são integrados, e o teste das funções gerais do *software*, realizadas por interação entre vários sistemas, é realizado em um veículo real ou em uma simulação de HIL do veículo. Conforme descrito em (KASOJU *et al.*, 2013), o teste de integração raramente é realizado por cada unidade organizacional responsável pela ECU, normalmente uma equipe se concentra no teste de integração das funções de *software* distribuído. A abordagem de teste exata, incluindo as ferramentas de teste usadas e os artefatos de teste gerados, pode variar entre diferentes empresas automotivas.

Essa grande interação entre os componentes exige um alto número de participantes envolvidos, o que torna o processo de teste bastante complicado, tanto no nível de planejamento, como também para o controle e gerenciamento (CHUNDURI *et al.*, 2016).

De acordo com Chunduri *et al.* (2016) e conforme discutido anteriormente, cada nível de teste possui diferentes objetivos, os testes das funções do *software* são visualizados com diferentes níveis de abstração em cada nível de teste. No nível de teste de integração, em que o foco é principalmente avaliar a execução de funções de *software* distribuído em um veículo, os requisitos geralmente dizem respeito a como a função deve ser executada dentro da arquitetura do automóvel. Já no teste de *software* e sistema, uma função é vista como um conjunto de módulos de *software* distribuídos em vários sistemas. Portanto, os requisitos para cada sistema geralmente dizem respeito a como cada módulo de *software* individual deve ser realizado, para garantir que a função seja executada quando o sistema estiver integrado.

Atualmente, um dos desafios enfrentados pela indústria automotiva é o contexto crítico de verificação e validação do *software* do sistema embarcado. Há poucos estudos na literatura que estabeleçam uma abordagem voltada para a indústria automotiva testar seu *software* de sistema embarcado (CHUNDURI *et al.*, 2016; KASOJU *et al.*, 2013). Além disso, as pesquisas em teste, verificação e validação no domínio automotivo estão focadas em testes e verificações baseados em modelos específicos de sistemas de baixo nível; e estudos relativos ao teste de integração de alto nível de funções de *software* distribuído são escassos (CHUNDURI *et al.*, 2016; SUDMARK *et al.*, 2011; BRINGMANN *et al.*, 2008).

Existem muitos desafios para os testes de *software* na indústria automotiva, destacam-se nesta tese os seguintes:

- Complexidade dos testes devido à natureza distribuída das funções do *software* (SUDMARK *et al.*, 2011; HAGHIGHATKHAH, 2017);
- Falta de transferência de conhecimento entre os testes (KASOJU *et al.*, 2013);
- Dificuldade para identificar a contribuição e qualidade nos diferentes níveis de testes (SUDMARK *et al.*, 2011);

- Testes sobrepostos entre os níveis de teste, causando desperdício de tempo e recursos (KASOJU *et al.*, 2013; CHUNDURI *et al.*, 2016);
- Falta de documentação adequada para medição dos esforços para de realização dos testes (KASOJU *et al.*, 2013);
- Elevado número de variantes para um único veículo, seus sistemas e componentes, explodindo o número de combinações a serem testadas (SUNDMARK *et al.*, 2011);
- Falta de requisitos de alto nível claramente testáveis (KASOJU *et al.*, 2013) e a falta de rastreabilidade como um obstáculo ao teste (POST *et. al.*, 2009; LEE *et. al.*, 2011; KASOJU *et. al.*, 2013; BRAUN *et, al.*, 2014);
- Falta de um processo de teste unificado para todos os níveis (PARK *et. al.* 2009; KASOJU *et. al.*, 2013); e
- Deficiências na forma de monitoramento dos resultados de testes (KASOJU *et. al.*, 2013).

A indústria automotiva possui características particulares que influenciam a engenharia de *software*, o tema precisa ser sistematicamente explorado para auxiliar as pesquisas e dar suporte ao aumento da complexidade dos sistemas automotivos (HAGHIGHATKHAH *et. al.*, 2017).

O tema de testes dentro da engenharia de *software* em geral é maduro e razoavelmente estudado, as pesquisas focam em diversas áreas como: ferramentas de teste de *software*, abordagens, práticas e processos de teste (HAGHIGHATKHAH *et. al.*, 2017), entretanto, na área de eletrônica embarcada e no domínio da indústria automotiva, existem limitações nos estudos encontrados. Essa abordagem é focada mais nos testes de baixo nível, ou seja, abaixo do nível de componente. Faltam

estudos que incluam o teste de software nos níveis mais alto, níveis de integração de sistema (CHUNDURI *et al.*, 2016).

São necessários estudos científicos que conduzam a exploração sistemática dos testes de software dentro de complexo sistemas, para fornecer diretrizes de desenvolvimento e dar base para evolução tecnológica da indústria automotiva (CHUNDURI *et al.*, 2016; HAGHIGHATKHAH *et. al.*, 2017). Esta tese amplia os estudos da engenharia de software dentro do domínio da industria automotiva, focando nos testes dos sistemas embarcados, conforme Figura 18.

Figura 18 – Ilustração da área de estudo da tese.

Fonte: Elaborada pelo autor.

2.4.3 TESTE DE SOFTWARE EM APLICAÇÕES AUTOMOTIVAS

As atividades de teste e verificação são amplamente conhecidas por representarem uma parte importante do custo de desenvolvimento do *software*. No caso das aplicações automotivas, em que o *software* é aplicado em sistemas embarcados, existem desafios adicionais em comparação a teste de *software* regular (SARJEANT *et al.*, 2015); a mesma afirmação pode ser feita com relação à segurança funcional desses sistemas. À medida que a complexidade e o tamanho

do *software* incorporado aumentam, o custo e o esforço para testar seguem naturalmente essa tendência. Dentro desse cenário, existe a necessidade de reduzir o esforço da atividade de teste e, ao mesmo tempo, conseguir que uma maior garantia de qualidade seja alcançada pela atividade (FLEMSTRÖM *et al.*, 2015).

As pesquisas encontradas na literatura, relacionadas com verificação e validação de *software* em domínios automotivos, revelaram que as contribuições nessa área são dominadas por métodos desenvolvidos para fins de testes e verificações baseados em modelos de baixo nível (SUNDMARK *et al.*, 2011; CONRAD, 2004; AWÉDIKIAN, 2014), o que dificulta encontrar metodologias em que se discutam a estruturação e o planejamento completo dos testes no veículo.

Haghighatkhal *et al.* (2017), em seu mapeamento sistemático sobre engenharia automotiva de *software*, argumenta que são fundamentais uma metodologia de desenvolvimento abrangente, juntamente com suporte a ferramentas de desenvolvimento e integração baseadas em modelos. Também destaca a falta de contribuição analítica e teórica para avançar na pesquisa e no apoio à engenharia de sistemas automotivos complexos num futuro próximo. De maneira semelhante, Sundmark *et al.* (2011) discute o processo de liberação do sistema elétrico automotivo e a grande utilidade para a indústria de alguns estudos sobre métodos e técnicas para lidar com testes de grandes configurações (SUNDMARK *et al.*, 2011).

Chunduri (2016) propôs uma estratégia de verificação que aprimora o processo para identificar e mitigar lacunas e redundâncias em testes de *software* de sistemas automotivos. Foi aplicado o método em que a função é descrita e estruturada como uma “árvore” (UCT), para melhor identificar todas as ramificações da função e, assim, definir a estratégia de teste geral para cada função; abranger com eficiência todo o teste da função e identificar também redundâncias em cada nível de teste. O método UCT descreve passo a passo as interações entre os sistemas e subsistemas para mapear as funções do sistema embarcado, é escrito em uma linguagem natural de fácil compreensão.

Um modelo de como aplicar UCT é apresentado por Somé (2006). Essa abordagem facilita a visualização e torna fácil a identificação de cada parte do

sistema, ajudando eliminar redundâncias e lacunas nos testes. Outra contribuição relevante na investigação de Chunduri *et al.* (2016) foi a informação de como a Scania organizou as atividades gerais do teste.

A Scania, como muitos outros OEMs, usa a abordagem do modelo V para o desenvolvimento do sistema de EE e divide o nível de teste em várias etapas, conforme mostrado na Figura 19. Entretanto, conforme descrito por Chunduri *et al.* (2016) não existe um método que auxilia o planejamento dos testes em todos os níveis. Ao contrário, deixa ao responsável de cada nível definir o que precisa ser validado, sem uma avaliação geral de todas as etapas para identificar as lacunas e redundâncias.

A visualização dos cenários de função para definir a estratégia de teste só é possível quando requisitos de funcionalidade estão disponíveis. Há uma lacuna a ser trabalhada no início do projeto, quando a estratégia de teste precisa ser planejada e nem todas as especificações estão disponíveis. Considerando que representa uma grande parte do desenvolvimento, o custo das atividades de teste precisa ser estimado para avaliar o quadro geral do projeto.

Figura 19 – Modelo V aplicado na Scania para desenvolvimento dos sistemas embarcados distribuídos

Fonte: (CHUNDURI *et al.*, 2016)

Existem investigações importantes de Perez *et al.* (2010 e 2009) que introduzem e discutem a estratégia de teste em multinível, principalmente na reutilização do roteiro de teste. Os níveis de teste são relatados a partir de diferentes pontos de vista, como teste de integração, de multinível, *bottom-up* e *top down*. A discussão vai além do escopo dos detalhes do nível de teste e propõe métodos para definir uma estratégia de teste para reduzir o esforço, compartilhando artefatos de teste comuns em todos os níveis. Uma das abordagens principais de Perez *et al.* (2010) demonstra que a essência do teste funcional está presente em todos os níveis, o que possibilita a utilização da sequência dos testes no nível mais alto, também para os níveis inferiores, com a inclusão de adaptadores para entradas e saídas e parâmetros que adequem o roteiro para outros níveis, conforme mostrado na Figura 20.

Figura 20 – Estruturação do teste de software em multinível

Fonte: (PÉREZ *et al.*, 2010)

Na abordagem de planejamento dos testes de sistemas eletrônicos embarcados na indústria automotiva, é necessário incluir a ISO 26262 (2011). Um padrão para segurança funcional de sistemas elétricos e/ou eletrônicos em automóveis de produção. Os testes de *software*, para sistemas de função onde o ASIL é A, B, C ou D, precisam seguir requisitos de testes definidos, incluindo o nível de teste e plataforma em que cada função deve ser testada. As investigações de Pérez *et al.* (2010) e Chunduri *et al.* (2016) não incluem essa variável na definição da estratégia de teste proposta. Essa variável será incluída na estratégia proposta por esta tese.

A proposta apresentada utilizará parte da contribuição de Pérez *et al.* (2010); sua metodologia mostra que a especificação do nível do sistema é uma parte comum validada em todos os níveis de teste, auxiliando a rastreabilidade das atividades de testes, com base no processo de desenvolvimento do modelo em V. Ainda, adiciona-se a contribuição de Chunduri *et al.* (2016), que detalhou em baixo

nível uma estratégia para obter uma visão da contribuição de função de cada componente.

Em resumo, a combinação da visão sistêmica de Pérez *et al.* (2010) com a rastreabilidade funcional de Chunduri *et al.* (2016), adaptadas à visão de modelo em V, são a base para definir nossa contribuição nesta tese. Adiciona-se a ela a avaliação dos pontos importantes da indústria automotiva, incluindo questões relevantes de segurança da ISO 26262 (2011). Um método sistemático será projetado, com a finalidade de planejar e estruturar os testes de software em sistemas embarcados automotivo, conforme Figura 21 abaixo.

Figura 21 – Metodologia proposta para Tese

Fonte: Elaborada pelo autor.

2.4.4 RASTREABILIDADE DOS TESTES DE SOFTWARE

A rastreabilidade dos testes é um assunto bastante abordado na literatura. É um meio de alinhar requisitos e teste durante o desenvolvimento de *software* e sistema (BARMI *et al.*, 2011; OOI *et al.* 2014). O estabelecimento de um link através da rastreabilidade entre essas duas disciplinas melhora o resultado geral do processo de desenvolvimento de *software* e sistema (BARMI *et al.*, 2011).

Uma rastreabilidade focada nos requisitos de testes ajuda a determinar quais pontos serão cobertos por quais testes. A rastreabilidade também ajuda a identificar o grau de cobertura de teste dos requisitos e permite a adoção de uma métrica

adequada e precisa para a medição da cobertura de teste (BARMÍ *et al.*, 2011). Ter meios precisos de medir a cobertura de teste é crucial para a indústria (OOI *et al.*, 2014) e é um dos pontos tratados na metodologia dessa tese.

Embora os pesquisadores e os profissionais da indústria identifiquem a necessidade de interligar os requisitos e os testes, na maioria das vezes, pouco esforço é feito para concretizar essa necessidade, deixando uma lacuna entre as duas áreas (BARMÍ *et al.*, 2011; REMPEL *et al.*, 2013). Consequentemente, a rastreabilidade é um dos desafios no alinhamento de requisitos e processos de teste (SABALIAUSKAITE *et al.*, 2010).

Mesmo dentro do contexto da indústria automotiva, onde existe o alinhamento de requisitos e testes através de rastreabilidade, o tema ainda é um desafio no sentido operacional (KASOJU *et al.*, 2013). Além disso, para sistemas grandes e complexos, como no caso de sistemas automotivos, normalmente existem requisitos em múltiplos níveis de abstração (CHUNDURI *et al.*, 2016). Por isso, é necessária uma solução que estruture a rastreabilidade em todos os níveis de testes.

A rastreabilidade efetiva raramente acontece por acaso ou por meio de esforços individuais de cada participante do processo. Para estabelecer uma rastreabilidade eficaz, é importante ter o mapeamento da função como um todo e identificar cada uma das etapas e processos de teste (REMPEL *et al.*, 2013). A incorporação da rastreabilidade foi estudada em várias abordagens de especificação e teste de requisitos. Descobriu-se que a maioria desses estudos se concentrava em estabelecer a rastreabilidade na especificação e nos testes de requisitos baseados em modelos formais (BARMÍ *et al.*, 2011). Outros estudos abordaram o estabelecimento de rastreabilidade por meio da especificação de requisitos semi-formais e testes baseados em casos de uso e cenários (NASLAVSKY *et al.*, 2005; THANGARAJAH *et al.*, 2011).

Neste trabalho, o foco não é trazer uma nova metodologia de rastreamento, mas incluir, dentro da estratégia de planejamento dos testes, um mapeamento geral do sistema, identificando e estruturando cada etapa do complexo sistema eletrônico automotivo a fim de facilitar a organização e a implementação da rastreabilidade.

3 DESENVOLVIMENTO DA METODOLOGIA DE DEFINIÇÃO DA ESTRATÉGIA DE TESTES DE *SOFTWARE* NA INDÚSTRIA AUTOMOTIVA

Conforme descrito na seção 2.3, os sistemas eletrônicos embarcados automotivos possuem um alto nível de complexidade. Assim, o primeiro desafio deste trabalho é organizar essa complexidade: estruturar as interfaces entre funções, componentes e sistemas, para facilitar a definição das atividades e responsabilidades dentro do complexo mundo automotivo globalizado.

A base para organizar os sistemas embarcados utilizará como referência dois modelos da literatura: o "design baseado em função" (FIPS, 1993), que traz uma abordagem de desenvolvimento baseada na compreensão completa da função; e a estruturação baseada em MBD (LENNON, 2007; NICOLESCU, 2010), que é amplamente utilizado para o desenvolvimento de *software*, principalmente para facilitar a divisão de atividades complexas em pacotes de trabalho. Esta combinação trará uma visão estruturada e funcional de cada função do veículo, e ao mesmo tempo, irá modularizar as atividades, facilitando a divisão de responsabilidade e a organização das atividades.

Com a aplicação desses métodos, o *software* será dividido em módulos, facilitando o uso da metodologia AUTOSAR (2018), que padroniza toda a estrutura de *software*. Isso trará transparência e robustez ao desenvolvimento, além de, com a modularidade, facilitar a reutilização do *software* para outras aplicações. Além de gerar organização para todas as atividades de desenvolvimento, auxiliando no cumprimento das exigências do padrão ISO 26262 (ISO 26262, 2011) e trazendo rastreabilidade em todos os pontos do desenvolvimento do sistema.

O sistema elétrico geral do veículo pode ser definido como um conjunto de sistemas, onde cada sistema fica responsável por agrupar um conjunto de funções relacionadas ao seu tema, que, por sua vez, possui um número de componentes responsáveis por executar uma função ou parte dela, conforme apresentado na Figura 22.

Figura 22 – Correlação de sistema, componente e função.

Fonte: Elaborada pelo autor.

De acordo com o modelo V, na Figura 15, todos os níveis de teste são diretamente ligados a uma etapa de especificação do projeto. A partir da especificação, é definida uma estratégia para realização dos testes que, uma vez definidos, possibilitam analisar e identificar suas redundâncias e lacunas (PÉREZ *et al.*, 2010; CHUNDURI *et al.*, 2016).

A estratégia de teste precisa ser assertiva e objetiva desde o início do projeto, considerando que o plano de teste afeta diretamente o custo do projeto, se os valores ultrapassarem o limite do projeto, isso pode motivar seu cancelamento.

Neste trabalho, as primeiras quatro seções definem a base da aplicação da metodologia para organizar e estruturar a complexidade do sistema automotivo, são elas: 1) Especificação de função; 2) Especificação do sistema; 3) Mapeamento dos sistemas, componentes e funções; e 4) Definição dos fatores característicos da aplicação que influenciam os testes de *software* no domínio da indústria automotiva, Figura 23. Na seção 3.5, será proposto um fluxograma que consolida a estratégia de definição dos níveis de teste para cada função. E na seção 3.6 as atividades de teste são estruturadas para possibilitar o rastreamento, gerenciamento e controle das atividades de testes.

Figura 23 – Sequência de causas em que o *software* é uma fonte potencial de riscos.

Fonte: Elaborada pelo autor.

3.1 ESPECIFICAÇÃO DA FUNÇÃO

O veículo tem várias funções que são descritas pela parte interessada do veículo, de forma facilmente compreensível, numa linguagem para não especialistas, principalmente na fase inicial do projeto. A parte interessada, neste caso, pode ser um passageiro, motorista, operador, oficina de manutenção ou produção, dentre outros, chamados também de *stakeholders*.

Na fase inicial do projeto, a função é descrita independentemente do *hardware* ou *software*. Sua descrição não leva em conta como a função será executada, nem suas interdependências dentro do veículo. A Tabela 2 mostra o modelo proposto nesta metodologia para especificar as funções. Todas as funções do veículo receberão uma identificação Fn, um nome e uma descrição básica. Como exemplo, as funções utilizadas por Marrero Pérez *et al.* (2010) e Chunduri *et al.* (2016) são descritas na tabela.

Tabela 2 – Especificação da função veicular.

Identificação da função (Fn)	Nome da Função	Descrição Básica
F3	Luz Externa Automática	Ligar automaticamente as luzes externas do veículo quando a iluminação externa estiver escura.
F5	Display do nível de combustível	Indicar o nível de combustível para o motorista.

3.2 ESPECIFICAÇÃO DO SISTEMA

O veículo é composto de vários sistemas, e cada sistema é composto de elementos que têm a mesma classe de funções. A forma como o sistema é definido é diferente de uma empresa para outra, normalmente é um nome correlacionado com um pacote de funções. Nesta tese, a proposta é utilizar um modelo básico para descrição do sistema, conforme Tabela 3, no qual será incluído o mínimo de informação. Cada sistema receberá uma identificação, um nome, uma pequena descrição da relação do sistema com a função específica e, nas duas últimas colunas, o nome das funções que o sistema representa.

Tabela 3 – Modelo básico para especificação do sistema.

Sistema Identificação Sn	Nome do sistema	Descrição do sistema em relação a função	Funções relacionadas aos sistemas Nome	Descrição
S2	Luzes Externas	Identifica a quantidade de luz externa e Aciona a Luz externa	F003	Luz Automática Externa
S4	Cockpit de motorista	Controla a ativação da função de Luz externa automática	F003	Luz Automática Externa
S4	Cockpit de motorista	Indica o nível do combustível para o motorista	F005	Exibição de Nível de Combustível

3.3 MAPEAMENTO DOS SISTEMAS, COMPONENTES E FUNÇÕES

O mapeamento, neste caso, utilizará o conceito de arquitetura do sistema eletrônico (RUSHTON *et al.*, 2000; REICHART *et al.*, 2004), que define como os componentes e sistemas estão organizados, incluindo o mapeamento da função dentro da arquitetura do sistema, apresentando cada componente dentro de seu sistema e mostrando a correlação de cada função entre os componentes. Em suma, o mapeamento focará na interação entre quatro informações importantes do sistema elétrico do veículo: função, sistema, componente e contribuição de função. A informação de nível mais baixo nesta metodologia é a especificação da contribuição da função, que representa a parte da função da qual o componente é responsável.

Nesta etapa, a função é mapeada dentro de cada componente e sistema. O método proposto nesta tese é ilustrado com o exemplo da Figura 24, em que os sistemas são apresentados com seus componentes e as funções são mapeadas, representada por uma linha ligando cada componente que participa da execução da função.

No exemplo da figura, a mesma função, reportada no artigo de Pérez *et al.* (2010) e Chunduri *et al.* (2016), foi utilizada contendo 3 sistemas: o “cockpit” do motorista, trem de força e luzes externas. O sistema “cockpit” do motorista representa o conjunto de funções que estão em torno da posição do motorista; no exemplo da Figura 24, os componentes: painel de instrumento, controlador da ignição e o interruptor do farol. O sistema trem de força representa as funções relacionadas à força motriz do veículo. E o sistema luzes externas representa todas as funções relacionadas às luzes que estão do lado de fora do veículo. A linha verde conecta os componentes que participam da função luz automática externa e a linha azul, a função do display do nível de combustível.

Figura 24 – Mapeamento da função.

Fonte: Elaborada pelo autor.

Este é um passo da especificação do sistema, em que cada componente tem a lista de suas contribuições de funções definidas. Operacionalmente, a Tabela 4 é a base definida nesta tese para a representação do mapeamento do sistema.

Tabela 4 – Mapeamento da função dentro do sistema do veículo.

Sistema Sn	Funções Fn	Componente (ECU) Cn	Contribuição de função
S2	F3	C5	Identificação do interruptor de ignição
		C9	Identificação do interruptor de luz
		C8	Identificação de luminosidade externa Ligue / desligue a luz
S4	F5	C9	Exibição de Nível de Combustível
		C4	Verifique as falhas
		C6	Medição do nível de combustível no tanque

Adicionalmente à Tabela 4, uma matriz é proposta na Tabela 5, para mapear todo o sistema do veículo, combinando todas as informações definidas anteriormente. Um dos pontos adicionais nesta tabela é a identificação para contribuição da função $F_n S_n C_n$, que representa um código exclusivo para cada contribuição de função e se torna parte do “DNA” da função.

Tabela 5 – Matriz dos sistemas, componentes e funções de um veículo.

Sistemas		S1			S 2			S3			Numero de componetes que participam da função	Numero de sistemas que participam da função
Componentes		C1	C2	C3	C4	C5	C6	C7	C8	C8		
Funções	F1	F1S1C1		F1S1C3						F2S3C8	2	1
	F2										1	1
	F3							F3S3C7			1	1
	F4		F4S1C2			F4S2C5					2	2
	F5					F5S2C5					1	1
	F6	F6S1C1			F6S2C4	F6S2C5	F6S2C6				3	3
	F7							F7S3C7			1	1
	F8		F8S1C2								1	1
	F9								F9S3C9		1	1
...											...	...
Numero de funções por componte		2	2	1	1	3	1	2	1	0	...	...

Nesta tese, o nível mais baixo de especificação de função é a contribuição de função e a identificação é $F_n S_n C_n$. Utilizando este código é possível identificar o componente e o sistema responsável por essa contribuição. Caso o componente tenha mais de uma contribuição para a função, o F_n receberá um ponto e o número da contribuição pode ser adicionado após o ponto, por exemplo, a função F1 tem duas contribuições do componente C1, neste caso a contribuição seria: F1.1S1C1 e F1.2S1C1.

Este método irá mapear a função no sistema global, no qual cada contribuição tem um número exclusivo; este número pode ser facilmente rastreado, gerando uma maneira sistemática e completa para visualizar as funções do veículo. Essa abordagem de mapeamento suportará qualquer atividade de gerenciamento, não apenas para testes, mas para o ciclo de vida de desenvolvimento completo, a produção e também o pós-venda. Por exemplo, se um veículo no campo tiver alguma falha de função, isso pode ser rastreado com facilidade para obter o histórico de especificação, testes e relatórios.

Esta abordagem também facilita o desenvolvimento de ferramentas para gerenciar e organizar o desenvolvimento completo, pois a identificação possui o “DNA” da função.

3.4 FATORES/ PARÂMETROS DO DOMÍNIO DA INDÚSTRIA AUTOMOTIVA

Como descrito no estado da arte desta tese, a indústria automotiva possui características e desafios específicos dentro do processo de desenvolvimento do produto, que precisam ser considerados na estratégia de teste. Dentre os quais, pode-se destacar as características da aplicação, segurança e padrão de qualidade com alguns requisitos para o teste, relevância do ponto de vista do cliente, nível de complexidade do sistema e complexidade da execução do teste.

Este trabalho está focado no domínio da indústria automotiva, mas essa metodologia também pode ser aplicada a outras áreas. Isto porque, conforme descrito na Figura 25, genericamente, quase todas as aplicações de sistemas embarcados possuem os fatores que estão aqui sendo propostos e avaliados, padrões de níveis de segurança, diferentes níveis de complexidade, influências do meio externo ou do operador na funcionalidade do sistema e exigem, em alguns casos, também a participação dos clientes.

Figura 25 – Principais fatores que precisam ser avaliados em sistemas embarcados para definição da estratégia de testes.

Fonte: Elaborada pelo autor.

3.4.1 SEGURANÇA FUNCIONAL

Conforme descrito na seção 2.3.6, a segurança funcional é um tópico relevante para a aplicação automotiva devido ao aumento do *software* embarcado em funcionalidades críticas do veículo, com sistemas complexos e processos desafiadores de desenvolvimento de produtos.

O elemento principal para definir a estratégia de teste com relação ao fator de segurança funcional, está na análise e avaliação dos riscos no início do ciclo de vida do projeto, onde os níveis de integridade de segurança automotiva (ASILs) são identificados para cada sistema. Com base nessa classificação, que identifica os sistemas com potencial riscos de acidentes, são definidos os requisitos para evitar o risco residual e métodos de testes são recomendados para verificar e garantir a aprovação do sistema.

O *software* não tem influência direta sobre os riscos, mas, em muitos casos, pode ser considerado responsável pela segurança do nível do sistema. O *software* tem a capacidade de passar falhas através de elementos do sistema. A compreensão de todo o ecossistema onde a função do *software* será aplicada é importante e necessária. Essas informações irão influenciar e contribuir com a definição da estratégia de teste de *software* e são um dos fatores usados para definir o plano de teste.

A identificação do ASIL de cada função começa com uma análise de riscos, todos os eventos perigosos são identificados no foco de segurança funcional. Existem na literatura vários métodos para a análise de risco utilizando o conceito ASIL (ISO, 2011, WARD *et al.*, 2012; VAN, 2012). Cada evento perigoso é classificado de acordo com a gravidade (S) das lesões que podem ser geradas, a probabilidade de um risco de lesão é ainda classificada de acordo com uma combinação de exposição (E), a frequência relativa esperada de condições de operação em que a lesão pode ocorrer, e controle (C), a probabilidade relativa que o controlador pode agir para evitar a lesão.

A classificação de risco usando o método ASIL, recomendado pela norma ISO 26262, será usada para avaliar cada função do veículo. Conforme descrito na tabela 5, existem 5 resultados possíveis: QM, A, B, C e D. Caso o nível for A, B, C ou D significa que a função sistema ou componente tem relevância à segurança funcional. O nível QM significa que o risco associado a um evento perigoso não é relevante à segurança e não requer medidas de segurança, de acordo com a ISO 26262 (ISO, 2011).

Assim que a especificação da função, sistema e o mapeamento estiverem definidos, a avaliação do nível de risco seguindo a metodologia ASIL deve ser aplicada individualmente para cada função. Essa classificação será um dos parâmetros para a definição da estratégia de teste.

Existem recomendações na ISO 26262 (ISO, 2011), que definem, de acordo com a Tabela 6, quais testes devem ser realizados a partir da definição do nível ASIL. Em resumo, quando a classificação é QM, não há relevância à segurança funcional e não existe recomendação, no caso do ASIL A, B, C ou D, há recomendações específicas para serem seguidas no planejamento dos testes.

Tabela 6 – Requisitos de teste da ISO 26262 para funções e sistemas relevantes à segurança funcional.

Métodos		ASIL			
		A	B	C	D
1a	<i>Hardware-in-the-loop</i>	++	++	++	++
1b	ECU integrada no ambiente de rede (*)	++	++	++	++
1c	Teste no Veículo	+	+	++	++

“++” indica que o método é altamente recomendado para o ASIL identificado;

“+” indica que o método é recomendado para o ASIL identificado;

(*) O ambiente de rede pode ser um banco de provas parcialmente ou totalmente integrado com o sistema do veículo.

Seguindo a recomendação da ISO 26262, será obrigatório testar o *software* nas condições reais do veículo quando o ASIL for A, B, C ou D. Porém no caso do

ASIL, A e B, esse teste pode ser em outros níveis, como: nível de componente; nível de sistema; e veículo HIL, desde que haja evidência de que nestes níveis de teste a representatividade das condições reais do veículo é garantida, eliminando o teste real do veículo. Entretanto, isso deve ser muito bem documentado, uma vez que, em caso de acidente no campo, a empresa deve estar preparada para explicar os detalhes do teste.

Para evitar erros de interpretação em um tópico tão relevante, se a função for ASIL, A, B, C ou D, o teste real do veículo será considerado como obrigatório. Em resumo, para a metodologia desta tese, o importante é saber se a função é relevante ou não à segurança funcional, se o nível da relevância for A, B, C ou D não fará diferença na definição inicial do planejamento da estratégia de teste.

3.4.2 RELEVÂNCIA DO CLIENTE

O cliente precisa fazer parte da estratégia do teste de *software*, considerando que nas últimas décadas o *software* passou a fazer parte da funcionalidade do sistema embarcado e da inovação em uma ampla variedade de domínios - de telecomunicações a dispositivos médicos -, incluindo os sistemas automotivos (LIGGESMEYER *et al.*, 2009).

Muitos produtos contendo *software* embarcados foram desenvolvidos para uso em vários campos, trazendo valor para a sociedade (FEITOSA *et al.*, 2010) e aumentando a relevância da participação do *stakeholder* no desenvolvimento do produto, incluindo a validação do *software*. Essas mudanças levaram a uma situação na qual a engenharia de *software* se tornou uma disciplina vital em sistemas embarcados e um importante participante em inovação e concorrência no mercado (LIGGESMEYER *et al.*, 2009), com impacto direto na usabilidade das funções do veículo. Dentro dos testes de *software*, esse fator precisa ser considerado.

A maioria dos métodos presentes na literatura não consideram, na definição da estratégia de teste, a validação no cliente (BINDER, 1999; PÉREZ *et al.*, 2010, CHUNDURI *et al.*, 2016). Essa validação é um dos níveis mais alto do modelo em V, nesta tese chamado de teste de aceitação. Para o planejamentos e definição da

estratégia de testes, a necessidade do teste de aceitação será avaliada com uma pergunta que precisa ser previamente respondida: a função exige a aprovação dos clientes antes de ser liberada para produção?

3.4.3 COMPLEXIDADE DO SISTEMA

A literatura tem muitas definições para o termo complexidade, Zayed *et al.* (2010) apresenta uma série de abordagens para caracterizar a complexidade, Neil Johnson (2009) afirma que “mesmo entre os cientistas, não há uma definição única de complexidade, mas tradicionalmente tem sido transmitida como o estudo de fenômenos que emergem de uma coleção de objetos interativos”. O sistema embarcado na indústria automotiva é um ótimo exemplo para ser explorado quando o tema é complexidade, principalmente devido à interação entre sistema, componente e funções.

Conforme apresentado na Figura 26, vários são os objetos que precisam ser desenvolvidos e testados, começando do nível mais baixo para o mais alto: modelos, unidades de *software*, *software* integrado, componentes com *hardware* e *software*, sistemas e o próprio veículo. Os testes podem ser realizados em diferentes níveis, sendo preciso criar uma metodologia sistemática que facilite o entendimento e definição do teste

Figura 26 – Estrutura do sistema eletrônico de veículos.

Fonte: Elaborada pelo autor.

Nesta tese, o foco será nos níveis mais altos de teste do *software*, por isso os objetos: modelos; unidade de *software*; e *software* integrado, não farão parte do contexto, considerando que na maioria dos casos esses testes de baixo nível possuem uma complexidade menor e são normalmente administrados e gerenciados pelos próprios fornecedores, que trazem para o OEM apenas o relatório final.

A análise de complexidade dessa tese terá como base a quantidade de interfaces que cada função do veículo possui, ou seja, quantos componentes e sistemas participam da funcionalidade da função e quantas funções cada componente ou sistema possui.

Para estruturar a definição dos testes no nível de componente, do sistema e do veículo, algumas perguntas serão formuladas:

a) Quantos componentes interagem com a função?

Essa resposta está inserida no contexto do mapeamento da função, definido pelo modelo proposto na Tabela 5 da seção 3.3. Caso a função tenha apenas um componente, o teste no nível de sistema não é necessário, quando há mais de um componente, uma segunda pergunta precisa ser respondida.

b) Quantos sistemas interagem com a função?

Da mesma forma que na questão anterior, essa resposta também está contida no mapeamento da função, definido pelo modelo proposto na Tabela 5 da seção 3.3. Existem duas respostas possíveis:

- Um único sistema: nesse caso todos os componentes fazem parte do mesmo sistema e podem ser completamente testados no nível de sistema, isso sem considerar a avaliação das sessões 3.4.4 e 3.4.5.

- Mais de um sistema: neste caso o teste de veículo ou do HiL do veículo é necessário.

3.4.4 INFLUÊNCIA DO MOTORISTA NA FUNCIONALIDADE DO VEÍCULO

Há uma grande diversidade do tipo de função no veículo, desde uma função muito simples e passiva até funções ativas, com impactos na segurança funcional. Algumas dessas funções têm o seu comportamento influenciado pelo operador, que na maioria dos casos é o motorista do veículo. Caso a interferência da ação do motorista não possa ser simulada em laboratório, o teste no veículo real é necessário.

Para facilitar a aplicação desse fator na metodologia proposta, duas questões serão incluídas para avaliação de cada função:

a) A função é funcionalmente influenciada pelo motorista?

Caso a resposta seja não, esse fator não influenciará o planejamento dos testes, caso seja sim, a pergunta seguinte precisa ser respondida.

b) A influência pode ser simulada em laboratório?

Caso a resposta seja sim, esse fator não influenciará no planejamento dos testes, caso seja não, o teste em um veículo real com o motorista deve ser considerado.

3.4.5 INFLUÊNCIA DO AMBIENTE EXTERNO NA FUNCIONALIDADE DO VEÍCULO

Esta seção tem a mesma abordagem da anterior, mas é relacionada à influências externas ao veículo. As novas tecnologias possuem cada vez mais interações com o meio externo; estas geram inúmeras variáveis dentro do sistema do veículo e sempre são passíveis de simulação em laboratórios. Pode-se citar os elementos do trânsito, diferentes tipos de ações e reações dos usuários, assim como situações de condução de natureza mais dinâmica, principalmente na direção da solução de veículos autônomos (BUTENUTH *et al.* 2017).

Outra forte influência são as características das funções relacionadas à conectividade ou eletromobilidade. Neste caso, o veículo é incluído em um ecossistema da cidade, como por exemplo, na estrutura de IoT (*Internet of Things*), infraestrutura de carregamento das baterias e esse tipo aplicação precisa de validações nos ambientes reais com o veículo para aprovação final do produto, e isto deve ser considerado no planejamento dos testes.

Grande parte dos novos veículos possui capacidades poderosas de rede, comunicação e processamento de dados, podem se comunicar com outros veículos ou trocar informações com os ambientes externos em vários protocolos, incluindo *wifi*, *bluetooth*, dentre outros (MCDONALD *et al.*, 2001). Como resultado, muitos serviços inovadores de telemática, como a segurança remota para desabilitar o mecanismo e o diagnóstico remoto, foram desenvolvidos para aumentar a segurança, a conveniência e a diversão dos motoristas (ZHAO, 2002). Um dos próximos passos nesta evolução é a proliferação de soluções de veículos autônomos (GERLA *et al.*, 2014).

Dentro deste cenário, as interfaces das funções do veículo, em muitos casos, vão além dos sistemas embarcados e precisam do veículo no ambiente real para validar as funções de *software*, pois a simulação em laboratório nem sempre é possível.

Na metodologia proposta nesta tese, algumas perguntas com relação a este fator precisam ser respondidas para definição da estratégia de testes:

a) A função é influenciada pelo ambiente externo?

Caso a resposta seja não, esse fator não influenciará o planejamento dos testes, caso seja sim, a pergunta seguinte precisa ser respondida.

b) A influência pode ser simulada no laboratório?

Caso a resposta seja sim, esse fator não influenciará no planejamento dos testes, caso seja não, o teste em um veículo nas condições reais da aplicação deve ser considerado.

3.4.6 RESUMO DAS PRINCIPAIS CARACTERÍSTICAS DO DOMÍNIO AUTOMOTIVO

Para facilitar a aplicação da metodologia, na Tabela 7, os fatores que influenciarão o planejamento dos testes e que devem ser avaliados por todas as funções são descritos de forma mais compacta. A última coluna da tabela mostra as opções que cada função pode ter:

- Fator 1 - Segurança funcional: a função pode ser QM, irrelevante à segurança funcional; ou nível ASIL A, B, C ou D que são relevantes à segurança funcional.
- Fator 2 - Relevância do Cliente: sim, para funções relevantes, que precisam da validação do cliente; não, para funções não relevantes, que não precisam da validação do cliente.
- Fator 3 - Nesse fator, dois aspectos são unificados, embora abordados separadamente nas seções 3.4.4 e 3.4.5: influência do motorista e do ambiente externo, considerando que os resultados são os mesmos para ambos, ou seja, se é possível ou não simular em laboratório.
- Fator 4 – Complexidade do sistema: nenhuma, quando interage com apenas um componente; pouca, quando a interação é em mais de um componente, mas no mesmo sistema; e grande, quando a interação é entre mais de um sistema.

Tabela 7 – Fatores de classificação de funções.

Símbolo	Fatores	Nível
<i>F1</i>	Nível do risco para Segurança Funcional: análise de risco de um possível perigo, analisando a Gravidade, Exposição e Controlabilidade do cenário operacional do veículo	QM – sem relevância. A, B, C ou D – com relevância.
<i>F2</i>	Relevância do cliente / A função precisa de validação do cliente antes da produção?	Não Sim
<i>F3</i>	Influência do Motorista / Ambiente Externo - Influência na funcionalidade / A função pode ser facilmente simulada?	Sim Não
<i>F4</i>	Nível de Complexidade da função	Nenhuma (1 ECU) Pouca (mais de 1 ECU, mesmo sistema) Grande (mais de 1 sistema)

3.5 METODOLOGIA PARA DEFINIÇÃO DA ESTRATÉGIA DE TESTES

As seções anteriores deste capítulo apresentaram: os fatores importantes, dentro do domínio da indústria automotiva, que influenciam a definição do plano de teste, um modelo de especificação das funções e sistemas, assim como uma forma de mapear a função dentro sistema. A estruturação da metodologia proposta pode ser resumida na Figura 27.

Figura 27 – Parâmetros para definição do plano e estratégia dos testes de multinível para aplicação no domínio da indústria automotiva.

Fonte: Elaborada pelo autor.

Esta seção apresenta o método para definição da estratégia de teste de multinível, utilizando como base as informações das seções anteriores.

O primeiro fluxograma proposto na metodologia desta tese, conforme descrito na Figura 28, avaliará o fator 1, ou seja, qual a relevância da função com relação à segurança funcional.

Figura 28 – Fluxo de trabalho para verificar a relevância da segurança funcional.

Fonte: Elaborada pelo autor.

Esse é um dos mais importantes fatores em aplicações automotivas, devido à potenciais consequências jurídicas e financeiras, caso um acidente aconteça devido a alguma falha no sistema eletrônico. A ISO 26262 (ISO, 2011) tem requisitos específicos para funções que são relevantes à segurança funcional e o plano de teste precisa seguir essas recomendações. A metodologia proposta nesta tese dividirá a estratégia e plano dos testes de multinível em duas partes: estratégia de teste para funções relevantes à segurança funcional e estratégia de teste para funções não relevantes à segurança funcional.

3.5.1 ESTRATÉGIA DE TESTE PARA FUNÇÕES RELEVANTES À SEGURANÇA FUNCIONAL

As funções que são relevantes para a segurança funcional, devem utilizar o fluxograma da Figura 29. A primeira etapa do fluxograma avalia o fator 4 (complexidade do sistema), caso a função possua interação com apenas um componente, o nível de teste do sistema pode ser eliminado. O teste de veículo, contudo, deve ser aplicado, de acordo com a ISO2626 (ISO, 2011), conforme resumido na Tabela 6.

Caso a função tenha interação com mais de um componente do mesmo sistema, o nível de sistema deve ser aplicado. No caso em que a função interagir com sistemas diferentes, a funcionalidade será completa apenas no nível do teste de veículo. Podem existir funções em que os componentes possuam interações no

nível de sistema, neste caso, o teste de nível de sistema é relevante e deve ser considerado, caso contrário o nível de sistema pode ser eliminado.

O último parâmetro a ser avaliado no fluxograma da Figura 29 é o fator 2, relevância do cliente na funcionalidade da função. Caso seja relevante o teste de aceitação deve ser considerado.

Figura 29 – Fluxograma para definição da estratégia de teste para funções relevantes à segurança funcional.

Fonte: Elaborada pelo autor.

3.5.2 ESTRATÉGIA DE TESTE PARA FUNÇÕES NÃO RELEVANTES À SEGURANÇA FUNCIONAL

As funções que não são relevantes para segurança funcional devem utilizar o fluxograma da Figura 30, a primeira etapa do fluxograma avalia o fator 4, complexidade do sistema.

O primeiro passo é avaliar a complexidade da função, seguindo a mesma abordagem adotada quando a função é relevante à segurança funcional, a única diferença, neste caso, é que o nível de teste do veículo não é pré-definido pela ISO 26262 (ISO, 2011). Outro ponto avaliado é se o motorista ou o ambiente externo influenciam na funcionalidade da função. Se for possível simular esse comportamento em laboratório, o teste de veículo deve ser considerado no escopo do teste. Da mesma forma que a seção anterior, a relevância do cliente na funcionalidade da função deve ser avaliada para saber se o teste de aceitação será necessário.

Figura 30 – Fluxograma para definição da estratégia de teste para funções não relevantes à segurança funcional.

Fonte: Elaborada pelo autor.

3.5.3 CONSOLIDAÇÃO DA ESTRATÉGIA DE TESTES

A combinação dos parâmetros apresentados na seção 3.4 e dos os fluxogramas definidos na seção 3.5 resultam em oito possíveis cenários de testes para cada função, conforme descrito na Tabela 8.

Tabela 8 – Tabela de decisão dos níveis de Teste

Identificação da Função	Parâmetros				Níveis de teste necessário			
	Relevante à segurança funcional	Complexidade	Influência do Motorista/ Influência do ambiente externo	Validação do cliente necessária	Nível Componente	Nível Sistema	Nível Veículo	Nível Aceitação
Fn	Não	Nenhuma	Não	Não	X			
Fn	Não	Nenhuma	Não	Sim	X			X
Fn	Sim?	Grande?	Sim?	Não	X		X	
Fn	Sim?	Grande?	Sim?	Sim	X		X	X
Fn	Não	Baixa	Não	Não	X	X		
Fn	Não	Baixa	Não	Sim	X	X		X
Fn	Sim?	Grande?	Sim?	Não	X	X	X	
Fn	Sim?	Grande?	Sim?	Sim	X	X	X	X

X Teste necessário

Os parâmetros que contêm o ponto de interrogação estão relacionados com a questão da necessidade do teste de nível do veículo. Apenas um parâmetro definido por linha, com a resposta da tabela, é suficiente para que o teste de veículo seja necessário. Caso os demais parâmetros da mesma linha com ponto de exclamação sejam diferentes do valor apresentado, o teste no nível de veículo deve ser mantido. Em resumo, a sequência das atividades para ter a definição do nível de teste é:

- Especificação básica da função;
- Especificação do sistema
- Definição de arquitetura do sistema e mapeamento da função dentro do sistema do veículo;
- Preenchimento dos quatro parâmetros da aplicação (seção 3.4);
- Utilização o fluxograma proposto na seção 3.5 para definir os níveis de teste;
- Preenchimento do modelo (Tabela 8) com os resultados.

Quando a Tabela 8 é preenchida com todas as funções do parâmetro e do cenário de nível de teste definido, a quantidade de teste para cada nível pode ser facilmente visualizada; isso pode se dar por componente, por sistema, por teste de veículo, conforme Figura 31.

Figura 31 – Quantidade de testes por nível.

Fonte: Elaborada pelo autor.

A metodologia proposta traz uma forma sistemática de organizar e controlar as atividades e define quais são os níveis de testes necessários para cada função. Isso padroniza e traz robustez às atividades de teste de *software* do sistema embarcado.

3.6 RASTREABILIDADE DAS ATIVIDADES DE TESTE

A rastreabilidade será dividida em duas partes: 1) identificação das atividades de testes; e 2) alinhamento do *test case* ou caso de teste.

3.6.1 IDENTIFICAÇÃO DAS ATIVIDADES DE TESTE

Para estabelecer uma rastreabilidade eficaz, é importante ter o mapeamento da função como um todo, identificando cada uma das etapas e processos de teste (REMPEL *et al.*, 2013). Na seção 3.3, no mapeamento das funções do sistema embarcado, foi gerada uma forma de identificar a contribuição de função com o código FnSnCn. Tal contribuição corresponde a uma parte da função Fn, que é de responsabilidade do componente Cn; também é testada no nível de componente, o

nível mais baixo dos testes abordado nessa tese, assim, a responsabilidade é direcionada para o responsável pelo componente.

Já no caso de uma abordagem em que o sistema todo é analisado, principalmente em sistemas complexos - como no caso de sistemas automotivos, em que normalmente existem requisitos em múltiplos níveis de abstração (CHUNDURI *et al.*, 2016) -, o nível de componente somente representa uma parte das atividades de teste, os outros níveis precisam ter seu escopo e reponsabilidade definidos.

Uma vez que cada função tiver a definição dos níveis de testes necessários para aprovar sua funcionalidade, é necessário definir um método para organizar as responsabilidades e a rastreabilidade entre os níveis de teste, eliminando lacunas e evitando redundâncias de teste. Cada nível de teste possui uma sequência de passos, pré-definida para realização dos testes funcionais e definida individualmente para cada nível; não há interligação entre os níveis, nem também rastreabilidade dos seus resultados (PÉREZ *et al.*, 2010).

A Figura 32 demonstra as várias interações que uma função pode ter nos componentes e sistemas. Neste caso, dois componentes C1 e C2 dentro do sistema S1 interligam suas contribuições, o componente C3 do sistema S2 possui uma outra parte que contribui com a função, e a combinação S1 e S2 representa a função F1 completa.

Analizando o plano de teste para esse exemplo de função, principalmente o fator 4 (complexidade), as seguintes fases para o teste da função serão apresentadas:

- Primeira fase: Teste de componente. Teste individual para cada componente validar suas contribuições de função no nível componente. O componente C1 testa a contribuição da função F1S1C1, o componente C2 testa a contribuição da função F1S1C2 e o componente C3, a contribuição da função F1S1C3.

- Segunda fase: Teste de sistema. Teste da combinação das contribuições de função F1S1C1 e F1S1C2, na qual os componentes fazem parte do mesmo sistema S1. Existem dois possíveis cenários no nível de sistema: contribuições de funções que não tenham interação no nível de sistema, ou seja, nada muda no cenário de teste do nível componente e nível sistema; ou o cenário em que elas interajam, por exemplo, um sinal simulado no nível componente do C1 e no nível sistema é recebido diretamente via “CAN Bus” do C2.
- Terceira fase: Teste de veículo, envolvendo apenas S1 e S2, que possuem a combinação de todas as contribuições de função que representam a função F1.
- Quarta Fase: Teste de aceitação da função pelo cliente.

Figura 32 – Exemplo das interações entre sistemas e componentes de uma função.

Fonte: Elaborada pelo autor.

O mapeamento definido na seção 3.3 criou um método para identificar cada sistema, componente e contribuição de função, com um código único. Da mesma forma, as atividades de teste precisam de uma identificação única para cada fase de teste da função, possibilitando o rastreamento das atividades testes por nível e combinando as características da função e de cada etapa do teste. Considera-se que a documentação e o histórico da cada atividade é muito importante, principalmente para estruturar a complexidade das atividades de testes de *software* dos sistemas embarcados automotivos, descritos no estado da arte deste trabalho.

Os níveis de teste serão identificados por duas letras: teste de componente (TC); teste de sistema (TS), teste de veículo (TV) e teste de aceitação (TA), a primeira letra se refere ao teste e a segunda ao tipo de teste.

A identificação de cada função com os níveis de teste precisa estar de acordo com as características dos testes, conforme detalhado na Tabela 9 e na Figura 33. No nível de componente, cada atividade apenas adicionará a letra TC ao código da contribuição de função (TCFnSnCn). No teste de sistema, a principal identificação está relacionada aos componentes que fazem parte do teste da função, a identificação dessa etapa pode ter mais de um componente no código (TSFnSnCnC...). No nível de veículo, o destaque está em quais sistemas representam a função, a identificação dessa etapa pode ter mais de um sistema na composição do código (TVFnSnS...).

Tabela 9 – Estrutura para identificação da função no nível de teste.

Formula para Identificação da função no nível de teste				
Teste Componente (TC)→	Função (Fn)	Sistema (Sn)	Componente (Cn)	
Teste Sistema (TS)	→	Função (Fn)	Sistema (Sn)	Componente (Cn) Componente (Cn)
Teste Veículo (TV)	→	Função (Fn)	Sistema (Sn)	Sistema (Sn)
Teste de Aceitação (TA)→	Função (Fn)			

Figura 33 – Ilustração da identificação da função em cada nível.

Nível de teste	Código para identificar a função em cada fase do Teste			
Aceitação (TA)	Fn			
Veículo (TV)	Fn	Sn	Sn	
Sistema (TS)	Fn	Sn	Cn	Cn
Componente (TC)	Fn	Sn	Cn	

Fonte: Elaborada pelo autor.

A Tabela 10 apresenta um modelo do código e da interligação entre o código de cada atividade para todos os níveis. No nível de componente, quando existe mais de uma contribuição de função por componente, adiciona-se um ponto depois do código da função, com um número de 1 a n de funções secundárias existentes no componente, conforme exemplo na primeira coluna.

Tabela 10 – Modelo do código de cada atividade de teste.

Atividades no nível de Componente		Atividades no nível de Sistema	Atividades no nível de Veículo	Atividades de nível de Aceitação
Secundária	Principal			
TCFN.1S1C1	TCFNS1C1	TSFNS1C1C2	TVFN	TAFN
TCFN.2S1C1				
TCFN.NS1C1				
TCFN.1S1C2	TCFNS1C2			
TCFN.2S1C2				
TCFN.NS1C2				
TCFN.1S2C3	TCFNS2C3	TSFNS2C3C4		
TCFN.2S2C3				
TCFN.NS2C3				
TCFN.1S2C4	TCFNS2C4			
TCFN.2S2C4				
TCFN.NS2C4				
TCFN.1SNC5	TCFNSNC5	TSFNSNC5CN		
TCFN.2SNC5				
TCFN.NSNC5				
TCFN.1SNCN	TCFNSNCN			
TCFN.2SNCN				
TCFN.NSNCN				

3.6.2 ALINHAMENTO DO TEST CASE ENTRE OS NÍVEIS

O *test case* ou caso de teste é uma sequência de passos do teste específico, que examina todos os aspectos, incluindo entradas e saídas, de um sistema ou componente, fornece uma descrição detalhada das etapas que devem ser tomadas, bem como os resultados que devem ser alcançados e outros elementos que devem ser identificados. Normalmente, cada nível de teste de uma função possui um caso de teste específico, que não tem alinhamento com o conteúdo dos outros níveis.

Na indústria automotiva, a maioria dos componentes são desenvolvidos por fornecedores (LARRUCEA *et. al*, 2017), com exceção de alguns casos que são desenvolvidos pelo próprio fabricante do veículo. No caso dos componentes desenvolvidos pelo fornecedor, seria ideal para o OEM receber o componente totalmente testado, como uma “caixa preta”, onde apenas a integração seria necessária. Mas atualmente esses componentes fazem parte da combinação de

funções do veículo e é difícil isolar o componente, a integração é obrigatória e necessária. Para melhorar a eficiência e o aproveitamento de todos níveis das etapas dos testes, é necessário criar um modelo em que o teste de um nível seja aproveitado em um nível superior, evitando duplicidade ou até mesmo lacunas nos testes.

No nível de componente, o número de participantes envolvidos é menor e mais fácil de controlar, por isso, manter o número máximo dos testes neste nível reduzirá a complexidade para o próximo nível de teste. As falhas básicas do código fonte e de integração com o *hardware* do componente podem ser identificadas com antecedência já no nível de componente, deixando para os outros níveis, de sistema e veículo, apenas os testes de integração e validação da interface entre os componentes.

Outro ponto importante é que os custos para corrigir erros aumentam de forma drástica quando são descobertos tardiamente no processo de desenvolvimento. Manter a maioria dos testes no nível mais baixo, ou seja, no caso desta tese no nível de componente é bastante benéfico. A Figura 34 mostra uma ordem cronológica dos testes no domínio da indústria automotiva.

Figura 34 – Tipos de testes para os componentes e sistemas de um veículo em ordem cronológica.

Fonte: Elaborada pelo autor.

O resultado dos testes no nível de componente precisa de um link com os outros níveis para obter o máximo aproveitamento dos resultados. Neste trabalho a proposta é utilizar o modelo de teste de multinível definido por Marrero Perez e Kaiser (2009), no qual uma sequência de teste no mais alto nível é definida como um núcleo comum entre todos os níveis. Nos níveis inferiores, essa sequência é reutilizada com a inclusão de adaptadores que ajustam as entradas e saídas da estrutura do teste e, em alguns casos, os seus parâmetros.

A proposta de Marrero Perez e Kaiser (2009) diz que o *test case* de uma função consiste em um núcleo comum, chamando *test case core* (TCC), que é definido como sendo o roteiro de teste no mais alto nível de teste da função. Para os testes do nível mais baixo, serão adicionados, quando necessário, alguns adaptadores: adaptador de teste da interface de saída (OTA), adaptador de teste da interface de entrada (ITA) e adaptador de teste de parâmetro (PTA), conforme mostrado na

Figura 35. O TCC é compartilhado com todos os níveis de teste, e o roteiro de testes de alto nível é reutilizado, apenas os adaptadores utilizados em cada nível de teste são específicos.

Uma função quase sempre possui um componente principal (MC) que contribui com sua funcionalidade (Dziobek *et al.*, 2008), há também opcionalmente um ou mais componentes que contribuem marginalmente, chamados componentes auxiliares (AC). A proposta de Marrero Perez e Kaiser (2009) de reutilização do TCC é aplicada apenas aos MC.

Figura 35 – Estrutura de um caso de teste multinível.

Fonte: (PEREZ et al., 2010)

O TCC descreve o roteiro de teste de determinada função do sistema em um nível de abstração. Para esta abordagem sistemática, o TCC será a base da rotina dos testes definida pelo OEM, os níveis mais baixos reutilizarão essa rotina de teste acrescentando quando necessário os adaptadores OTA, ITA e PTA para adequar as interfaces e parâmetros.

Tanto o ITA quanto o OTA precisam basicamente modelar o comportamento de componentes que não estão disponíveis em um determinado nível de teste, mas cuja presença é esperada pelo TCC. O comportamento de tais componentes ausentes não precisa necessariamente ser completamente modelado - um modelo de comportamento parcial cobrindo os sinais de interface necessários é suficiente (comportamento relacionado ao caso de teste). O PTA precisa mapear quaisquer parâmetros de teste que se apliquem ao objeto de teste.

Nesta tese, o uso do TCC será aplicado apenas entre os níveis de teste do componente, sistema e veículo. No caso do nível do teste de aceitação, esse conceito não será utilizado, pois esse teste não possui uma sequência padronizada que possa ser facilmente reutilizada para outros níveis; quando necessário, o teste de aceitação receberá uma sequência de testes exclusiva. A reutilização do TCC será aplicada apenas para o nível de sistema e componente, considerando que o nível máximo neste caso será o do veículo.

Conforme discutido nas seções anteriores, a maior parte do sistema embarcado no domínio automotivo tem seus componentes desenvolvidos pelos fornecedores, mas a responsabilidade final é do OEM. O maior desafio está na integração dos componentes dos fornecedores ao sistema eletrônico geral do veículo, em que muitas contribuições de funções interagem para gerar outras funcionalidades no veículo.

Adaptando a abordagem de Marredo Perez e Kaiser (2009), o teste funcional, o foco deste trabalho, será controlado e alinhado em todos os níveis, como definido na

Figura 35. A criação da sequência de teste no nível mais alto se torna uma referência para todos os outros níveis. Isso suportará o objetivo de manter todas as

sequências dos componentes principais de níveis mais baixos alinhadas e com a mesma direção do que foi definido no nível mais alto pelo OEM.

Conforme descrito na Figura 36, a reutilização da sequência de teste só será aplicada nos casos em que a alta complexidade da função (Fator 4) exigir dois ou três níveis de teste.

Figura 36 – Abordagem do caso de teste para reutilizar e adaptar para outro nível de teste.

Fonte: Elaborada pelo autor.

Considerando a rastreabilidade proposta nesta tese, o modelo proposto por Perez *et al.* (2009) será aplicado para trazer uma forma processual e sistemática de alinhar os testes principais das funções do veículo.

Será adicionado um código na atividade de teste, que reutilizará o *test case* de outro nível. O componente ou sistema que reutilizar o TCC receberá, no fim do seu código de identificação, as letras RE, para registrar que a atividade está alinhada com o TCC daquela função. Neste caso, o código da atividade de teste de *software* do componente ficará TCFNSNCNRE, e, no caso do sistema, TSFNS2CNCNRE. No nível do componente essa identificação será utilizada apenas no código principal.

4 APLICAÇÃO DA METODOLOGIA DE ESTRUTURAÇÃO E PLANEJAMENTO DOS TESTES

Para validar a metodologia de planejamento e estruturação dos testes de multinível, será estudado a aplicação do método em uma função real do veículo, executando passo a passo a metodologia proposta nesta tese. A função selecionada para ser utilizada é a do “Controle Automático de Luz”.

A sequência da aplicação proposta seguirá a seguinte ordem:

- Especificação básica da função;
- Especificação do sistema;
- Definição de arquitetura do sistema e mapeamento da função dentro sistema;
- Definição dos parâmetros da função (seção 3.4);
- Definição dos níveis de testes necessários para a função; e
- Definição da sequência de testes com a rastreabilidade proposta.

4.1 ESPECIFICAÇÃO DA FUNÇÃO

A especificação da função também é chamada de requisito de usuário ou requisitos de partes interessadas. A descrição deve estar em um nível de abstração alto, em que não é necessário conhecimento técnico para entender o que é a função. Na prática, esse tipo de descrição é feito por equipes de vendas e de *marketing*, que normalmente têm contato direto com o cliente.

O modelo possui três informações: identificação de função, nome da função e descrição da função, conforme Tabela 11.

Tabela 11 – Descrição das funções

Identificação da função (Fn)	Nome da Função	Descrição Básica
F003	Controle de luz automático	Ligue as luzes do veículo automaticamente quando a iluminação externa estiver escura.

Para evitar que exista uma comutação repetida, a função de controle automático da luz possui uma histerese de 10%. Quando a iluminação externa estiver abaixo do limite de 60%, a escuridão será detectada, o farol será ligado e permanecerá até que a iluminação tenha aumentado acima de 70% (10% de histerese), então o farol será desligado e permanecerá assim até que a iluminação diminua para o valor de 60%, isso será incluído na sequência dos testes.

4.2 ESPECIFICAÇÃO DO SISTEMA

Nesta etapa, os sistemas que participam da função são identificados. Na função ALC são necessários dois sistemas: 1) S1 que representa o sistema de interface do motorista com as funções em seu entorno, neste caso, a ignição do veículo que liga grande parte do sistema eletrônico, e a leitura do botão de acionamento da função luz automática; e 2) S2 que identifica a luminosidade externa e também controla o acionamento das luzes externas.

Seguindo o modelo proposto na seção 3.2, as informações são organizadas e descritas na Tabela 12.

Tabela 12 – Descrição dos sistemas que participam da função ALC

Sistema Identificação Sn	Nome do Sistema	Descrição do sistema em relação a função	Funções relacionadas aos sistemas	Descrição
S1	<i>Cockpit</i> do Motorista	Controla a ativação da função de Luz externa automática	F003	Luz Automática Externa
S2	Luzes Externas	Identificação da quantidade de luz externa e acionamento da Luz externa	F003	Luz Automática Externa

4.3 DEFINIÇÃO DE ARQUITETURA DO SISTEMA E MAPEAMENTO DA FUNÇÃO DENTRO SISTEMA

Há dois pontos principais a serem definidos e que fazem parte da especificação do sistema: a arquitetura da função dentro do sistema eletrônico do veículo e a contribuição de cada componente na funcionalidade da função. A Figura 37 apresenta a arquitetura da função e a Tabela 13 apresenta o detalhe da contribuição de cada sistema e componente. Nesta etapa, a informação ainda está em um nível abstrato, porém, algumas informações básicas dos sistemas e componentes do veículo são necessárias, isto normalmente já é atividade da equipe técnica do departamento de desenvolvimento do produto.

O sistema S1 possui 2 componentes que participam da função, C5 componente de controle da ignição e C9 componente de controle do botão de controle da luz. O sistema S2 possui um componente que participa da função, o componente C8 responsável pelo controle da luz externa.

Figura 37 – Arquitetura da Função

Fonte: Elaborada pelo autor.

A definição da estrutura dos sistemas do veículo é organizada de maneiras muito diferentes entre as empresas OEM, seguindo as premissas e organização de seus sistemas e componentes. É uma definição muito proprietária, em muitos casos, o modo como é organizado está mais relacionado com o negócio na empresa do que com uma definição técnica.

Na Tabela 13 cada parte do sistema é identificada com um código, seguindo a própria lógica de interação da função, conforme modelo definido na seção 5.3, e recebe uma descrição de sua participação, chamada de contribuição da função.

Tabela 13 – Contribuição da Função

Funções	Identificação do Sistema Sn	Componente envolvido (ECU)	Contribuição da Função	
F3	S1	C5	F3S1C5	Sinal do interruptor de ignição
		C9	F3S1C9	Sinal do interruptor de luz
	S2	C8	F3.1S2C8	Identificação de luminosidade externa
			F3.2S2C8	Ligar/desligar a luz

A codificação definida para cada contribuição de função é exclusiva e pode ser facilmente rastreada e identificada. Com o código é possível descrever exatamente de qual componente ou sistema a contribuição da função faz parte; o código leva o “DNA” da contribuição da função. O mapeamento da função dentro de cada sistema pode ser visualizado na Tabela 14.

Tabela 14 – Mapa da Função ALC no sistema do veículo.

Identificação do Sistema		S1	S1	S2
Identificação do Componente		C5	C9	C8
Identificação da Função	F3	F3S1C5	X	
		F3A1C10	X	
		F3.1S2C10		X
		F3.2S2C9		X

4.4 PARÂMETROS DA APLICAÇÃO

A Tabela 8 é referência para esta parte do estudo de caso. Uma avaliação da aplicação deve ser feita para cada um dos parâmetros, a função definida para o estudo de caso tem seu resultado apresentado na Tabela 15.

Tabela 15 – Definição dos parâmetros da função

Identificação da função	Parâmetros				
	Relevância de segurança	Complexidade de interação	Contribuição do nível do Sistema com a função	A função pode ser facilmente simulada?	A função precisa de validação do cliente antes da Produção?
F03	Sim	Grande	Não	Não	Sim

O método ASIL é utilizado para avaliar o nível de risco relacionado à segurança funcional. De acordo com a ISO 26262, o risco é avaliado com base nos perigos que um sistema pode apresentar, cada situação é avaliada em termos da gravidade de possíveis lesões, quanto tempo o veículo fica exposto a essa possibilidade de acidente, bem como a probabilidade relativa de que um motorista possa agir para evitar a lesão (WARD *et al.*, 2012).

Neste trabalho, a avaliação não será discutida em detalhes, considerando que este é um tópico disponível na literatura (ISO 26262, 2011; DEBOUK *et al.*, 2010; HASEGAWA *et al.*, 2017). A função ALC é considerada ASIL A (severidade - S3,

exposição – E2 e controlabilidade – C2), pois caso ocorra uma falha no sistema automático de luz do veículo isso pode causar um acidente grave, ou seja, a função é relevante à segurança funcional.

Com relação à complexidade, o nível de interação da função é grande, pois mais de um sistema participa da função com um componente no S2 e dois componentes no S3.

A função não possui contribuição no nível de sistema, pois apesar dos componentes C5 e C9 estarem no mesmo sistema S1, são contribuições individuais que não geram um resultado diferente com a combinação das suas contribuições de função no nível do sistema S1.

No caso da simulação da função, foi considerado que não é muito viável executar o procedimento completo em laboratório, principalmente em função da interferência dos aspectos externos na detecção do nível de iluminação.

O cliente não tem muita interação com a função ALC, apenas no acionamento do botão que ativa a função, portanto não é necessário o teste de aceitação. No caso de o botão de ativação ter uma interface bastante diferente com o motorista, o teste de aceitação pode ser considerado.

4.5 DEFINIÇÃO DOS NÍVEIS DE TESTE

Com todos os parâmetros definidos na Tabela 15 , considerando que a função é relevante à segurança funcional, o fluxo de trabalho da Figura 29 pode ser executado e resultará no planejamento dos testes, no qual a necessidade de cada nível de testes é avaliada para a função. O fluxo resultante para a função ALC é destacado em vermelho na Figura 38.

O primeiro fator avaliado é a complexidade da função, que neste caso é alto, pois a função é executada por mais de um sistema (S1 e S2). Na sequência, a contribuição do nível de sistema é avaliada; principalmente no caso da função ALC, não existe contribuição no nível de sistema, ou seja, a combinação entre sistemas

não gera um resultado diferente no nível do veículo. Neste caso elimina-se o nível de sistema e aplica-se direto o teste de veículo que é mandatório devido a função ser relevante à segurança funcional. A última parte do fluxograma avalia a necessidade de teste por parte do cliente. Para a função ALC considera-se que não existe uma complexidade ou interferência do cliente que exija essa validação, por isso o teste de aceitação não será planejado.

Figura 38 – Fluxograma de definição dos níveis de teste da função ALC.

Fonte: Elaborada pelo autor.

O resultado da avaliação da função ALC definiu que o nível do componente e o nível do veículo são necessários para a aprovação da função. O resumo é descrito na Tabela 16.

Tabela 16 – Níveis de teste requeridos para a função ALC.

Função	Testes Necessários			
	Nível Componente	Nível Sistema	Nível Veículo	Nível Aceitação
F03	X		X	

4.6 RASTREABILIDADE

A rastreabilidade dos testes será dividida em duas partes: 1) identificação das atividades de testes; e 2) alinhamento do *test case* ou caso de teste.

4.6.1 IDENTIFICAÇÃO DAS ATIVIDADES DE TESTE DA FUNÇÃO ALC

Na metodologia proposta nesta tese, cada atividade de teste precisa receber uma identificação exclusiva. Conforme o resultado da avaliação feita na seção 4.5, a função ALC precisa de 2 níveis de testes, os níveis de componente e de veículo.

Para as atividades do teste de componente, existem 3 componentes que participam da função. Com isso, 3 atividades principais serão necessárias; de acordo com o método da seção 3.6.1, cada atividade no nível do teste de componente recebe um código com as letras iniciais TC, que será somado ao código da contribuição de função e irá gerar o código TCFnSnCn. Utilizando as informações definidas na tabela 12, as atividades de testes do componente da função ALC serão:

- No componente C5 - TCF3S2C5
- No componente C9 - TCF3S1C9
- No caso do componente C9 existe mais de uma contribuição para função, F3.1S2C8 e F3.2S2C8, neste caso será definida uma atividade principal que representará a participação total do componente TCF3S2C8.

No nível de veículo, existem dois sistemas que participam da função S1 e S2, a identificação e a definição da seção 3.5; inicia-se com as iniciais TV somado ao número da função e os sistemas envolvidos, o que gera o código TVFNSnSn.... Utilizando as informações definidas na tabela 12, as atividades de teste do componente da função ALC será:

- TVF3S1S2

A Tabela 17 apresenta os códigos de todas as atividades da função ALC (F3), que precisa de 4 testes principais, 3 no nível de componentes e 1 no nível do veículo.

Tabela 17 – Níveis de teste requeridos para a função ALC antes da reutilização do TCC.

Atividades no nível de Componente		Atividades no nível de Sistema	Atividades no nível de Veículo	Atividades de nível de aceitação
Secundária	Principal			
-	TCF3S1C5	nn	TVF3S1S2	nn
-	TCF3S1C9			
TCF3.1S2C8	TCF3S2C8	nn		
TCF3.2S2C8				

- não existe
nn teste não é necessário

4.6.2 ALINHAMENTO DO *TEST CASE* OU CASO DE TESTE EM TODOS OS NÍVEIS DE TESTE.

Conforme proposto na seção 3.6.2, o *test case* do nível mais alto será utilizado como base para todos os outros níveis. No caso da função ALC, a sequência dos testes do nível mais alto é a do teste do veículo. Na Tabela 18, um *test case* da função ALC proposto por Perez et al. (2010) é utilizado para ser o TCC no nível do veículo deste estudo de caso.

Tabela 18 – *Test Case* para a função ALC no nível de veículo.

Sequência	Ações	Condição de Aprovação
1	Ignição ligada, colocar o interruptor do farol na posição automático e estar com o ambiente do veículo iluminação clara (acima de 70%).	Farol = desligado
2	Iluminação escura (abaixo de 60%).	Farol = ligado
3	Iluminação em (~65%).	Farol = ligado
4	Iluminação do ambiente aumentada em 10% (histerese).	Farol = desligado
5	Iluminação do ambiente diminuída em 10% (histerese).	Farol = desligado
6	Iluminação do ambiente diminuída em 10% (histerese).	Farol = ligado
7	Aumente a iluminação do ambiente até 60%	Farol = ligado
8	Aumente a iluminação do ambiente sem passar 70%.	Farol = ligado
9	Continue aumentando a iluminação do ambiente.	Farol = desligado

No nível do veículo, o teste é realizado normalmente utilizando todos os sensores e atuadores reais, apenas o ambiente externo, no caso a luz, pode ser simulado em um banco de prova ou no próprio ambiente real na aplicação. O teste de nível de componentes pode ser dividido em três partes, o mais importante é o componente core C8 (MC) e outros auxiliares (AC) C9 e C5. Seguindo o conceito de Perez *et al.* (2009), o componente C8 reutilizará o TCC, adicionando um ITA e um OTA.

O ITA e o OTA adaptam a interface do TCC à interface de teste, configurando o *test case* para o objeto de teste, neste caso para o componente C8. No caso da função ALC, a ilustração da reutilização do *test case* no nível de componente com a descrição do ITA e OTA é apresentada na Figura 39.

Figura 39 – Reutilização do *test case* no teste da função ALC.

Fonte: Elaborada pelo autor.

Um OTA é necessário para simular o componente auxiliar C5, para estimular o sinal de ignição por barramento CAN e também estimular o sinal do componente C9, sinal de interruptor luminoso por CAN. O adaptador deve seguir a especificação do componente auxiliar para estimular os valores corretos, por exemplo, o tempo de envio das mensagens CAN no barramento CAN. Após a definição dos ITA e OTA, o teste do componente pode ser executado, seguindo o *test case* proposto na Tabela 18, sem modificação na sua sequência.

Além disso, um teste específico deve ser feito em C5 e C9 para validar o teste completo de nível de componente dessa função. O teste do veículo deve ser executado somente quando todos os componentes forem aprovados no nível do componente, para evitar identificação de falhas básicas de componentes nos níveis superiores de teste. Neste estudo de caso, o objetivo foi mostrar a metodologia da rastreabilidade e não definir e discutir o conteúdo dos *test case*.

A Tabela 19 apresenta o código final de todas as atividades de teste do ALC, considerando a reutilização do *test case* no componente C8, ou seja, apenas no

teste do componente C8 foi adicionado as letras RE, registrando que esse *test case* está alinhado com o TCC.

Tabela 19 – Níveis de teste requeridos para a função ALC considerando a reutilização do TCC.

Atividades no nível de Componente		Atividades no nível de Sistema	Atividades no nível de Veículo	Atividades de nível de
Secundária	Principal			
-	TCF3S1C5	nn	TVF3S1S2	nn
-	TCF3S1C9			
TCF3.1S2C8	TCF3S2C8RE	nn		
TCF3.2S2C8				

- não existe

nn teste não é necessário

Aplicando a proposta de Pérez *et al.* (2009) na estruturação da rastreabilidade dessa tese, a identificação da reutilização do *test case* fica explícita no código TCF3S2C8RE, definindo um alinhamento com os testes superiores e estabelecendo um modo processual e sistemático de ligação dos testes entre os níveis, conforme definido na seção 3.6.2.

4.7 ANÁLISE DA APLICAÇÃO DA METODOLOGIA DE ESTRUTURAÇÃO E PLANEJAMENTO DOS TESTES

A primeira observação e conclusão com a aplicação da metodologia proposta para a função ALC, demonstra que o desenvolvimento da especificação para iniciar as atividades de planejamento e estruturação dos testes é bastante simples e sem necessidade de um aprofundamento técnico. Todos os passos foram realizados sem especificação detalhada no nível de componente.

O método utilizado para identificar cada parte da função estruturou de uma forma sistemática o mapeamento das funções dentro do sistema embarcado do veículo, facilitando a visualização de todas as partes do sistema embarcado.

Os fatores analisados no domínio automotivo para definir os níveis de teste da função adicionaram um grau de eficiência qualificada na definição dos níveis de

teste de cada função, incluindo uma forma processual, estruturada e fundamentada para a definição da estratégia dos testes. Isso contribuiu para a retirada do teste de nível de sistema, que seria redundante, dada a análise realizada.

A rastreabilidade desenvolvida nesta tese e aplicada à função ALC, estruturou e mapeou a complexidade das interligações entre os componentes e sistemas, criando controle, gerenciamento na execução do teste e na documentação dos resultados. Conforme o método aqui proposto, no primeiro passo, a função ALC foi dividida em 3 contribuições principais da função, cada parte recebeu um código único: F3S1C5; F3S1C9; e F3S1C8; a composição do código determina exatamente a qual função, componente e sistema a contribuição de função pertence. No segundo passo, a utilização do método de Perez *et al.* (2009) para reutilização do *test case*, junto com o método de codificação das atividades propostas nesta tese, sistematizou o alinhamento das atividades de teste.

5 CONCLUSÃO

Este trabalho apresentou uma contribuição para a estruturação e planejamento dos testes de *software* embarcado automotivo, definiu fatores importantes para os testes nesse domínio e projetou uma abordagem sistemática para definir um plano de teste funcional do sistema, para o desenvolvimento de um veículo completo e o alinhamento das atividades de teste. A metodologia foi criada usando a característica do domínio automotivo, mas pode ser estendida para outras aplicações, apenas seguindo a mesma sequência e ideia e detalhando cada requisito de aplicação e ambiente. Na metodologia proposta, dois pontos principais devem ser destacados: 1) estruturação e planejamento dos testes; e 2) rastreabilidade dos testes.

5.1 ESTRUTURAÇÃO E PLANEJAMENTO DOS TESTE DE SOFTWARE

A proposta apresentada nesta tese desenvolveu um método de desdobramento e identificação de todas as funções elétricas e eletrônicas do veículo. A partir de um sistema embarcado com alto nível de complexidade, composto por muitas funções, componentes e sistemas que interagem entre si, tornou-se possível

o mapeamento e a estruturação de todas as funções do veículo, através da: divisão da função em contribuições de função; criação uma identificação para cada parte da função; e designação correta da responsabilidade para cada componente, de forma a mostrar a correlação entre componentes e sistemas. A estruturação proposta organizou o sistema embarcado automotivo, facilitando a visualização e o entendimento do um sistema, ainda que seja um sistema complexo, com muitas interações entre os componentes e sistemas.

Junto com a estruturação dos sistemas embarcados, o trabalho identificou os fatores que influenciam as decisões de quais são os níveis de testes necessários para a validação de cada função do veículo, formalizando a definição da estratégia de todos os níveis de testes. Desse modo, a decisão do planejamento de testes não se individualiza por nível e pelo conhecimento e experiência do especialista que executa o teste daquele nível, o que traz eficiência do desenvolvimento de produto.

Quando os testes são avaliados em conjunto, elimina-se a existência de lacunas e redundâncias entres os níveis testes e torna possível identificar o plano exato de teste de cada função, melhorando a eficiência do processo de desenvolvimento do produto, com a aplicação customizada para cada função do veículo, de forma a eliminar desperdícios com testes desnecessários.

Destaca-se, dentre os fatores definidos para planejamento dos testes nesta tese, a segurança funcional, normatizada na indústria automotiva pela ISO 26262, que insere uma forma estruturada de definição do nível de risco de acidente para cada função, e a aplicação dos níveis de teste apropriados para mitigar ou identificar as possíveis falhas de cada função executada por software.

A abordagem proposta neste trabalho trouxe um método para estruturar e identificar cada parte do sistema, transformando a complexidade em um sistema totalmente identificado e organizado, de forma a criar uma base de planejamento da estratégia de testes e definir, sistematicamente, os níveis necessários para cada função do veículo. Essa estruturação facilitará a documentação das atividades de desenvolvimento, necessárias conforme a regulamentação da ISO 26262, para

demonstrar robustez e abrangência de todos os testes aplicados no desenvolvimento do produto em caso de acidentes.

O método de identificação dos níveis de testes necessários para cada função incluiu uma forma processual, estruturada e fundamentada para a definição da estratégia dos testes, facilitando a identificação de duplicidade e lacunas no planejamento dos testes. Esse método pode ser aplicado também para outras áreas fora do domínio da indústria automotiva.

Dentre os níveis de testes da metodologia desta tese, foi incluído de forma sistemática o teste de aceitação, que é pouco discutido na literatura, criando de forma processual a discussão sobre a participação do cliente nos testes de *software* para funções estrategicamente necessárias. Com a expansão das funções que possuem *software* dentro do veículo, haverá cada dia mais a interferência de funções que tenham interação direta com os usuários, por isso, essa validação precisa ser avaliada.

Todo o planejamento e estruturação propostos nesta tese podem ser definidos no início do projeto com o mínimo de informação, sem a necessidade de especificação detalhada dos componentes, antecipando o planejamento e a quantificação dos custos dos testes. Isso torna o processo decisório de aprovação dos projetos mais robusto e assertivo, pois o tempo e os custos necessários para testar cada projeto serão mais precisos, o que auxilia na tomada de decisões em que o custo e o tempo necessários para os testes são fatores decisórios.

5.2 RASTREABILIDADE

A rastreabilidade é um tópico complexo quando se discute desenvolvimento de *software*. Este trabalho procurou trazer a discussão para um nível mais prático, organizando a visão do sistema, passo a passo, e fornecendo uma identificação exclusiva para cada sistema, componente, função e contribuição da função de forma sistemática. Todos os objetos de alto nível do sistema embarcado receberam um código estruturado, o DNA de cada parte do sistema, tornando o complexo sistema embarcado rastreável.

Duas contribuições principais foram apresentadas: a criação de um método para identificação de todas funções, contribuições de função, sistemas e componentes; e a identificação das atividades de testes de componente, sistema, veículo e aceitação. A proposta, nos dois casos, definiu um modelo de código único e exclusivo, que possui na sua estrutura a configuração e estruturação da parte identificada dentro do sistema complexo embarcado. Além da contribuição para a fase de desenvolvimento do produto, o método de codificação cria uma base para identificar a documentação e o registro das atividades de testes do *software*, que podem ser facilmente rastreadas. Para futuros projetos ou para solucionar problemas de falhas em campo, apenas com o nome da função é possível resgatar todas os documentos e testes relacionados a ela.

5.3 TRABALHOS FUTUROS

O domínio automotivo já é bastante complexo, mas os desafios continuarão aumentando, com foco em inovações, como os carros autônomos, haverá cada vez mais veículos eletrificados e integração de soluções de conectividade (IoT). Dentro deste cenário, ações externas, como as de “*cracker / hacker*”, dentro do sistema do veículo podem ser um desafio para os futuros sistemas, o que pode influenciar no planejamento dos testes de *software*. Será necessário, então, estudar e incluir esse fator de vulnerabilidade dentro da metodologia para a definição do planejamento dos níveis de teste necessários à aprovação do sistema.

A metodologia proposta cria uma base estruturada de identificação e mapeamento do sistema embarcado com uma codificação exclusiva para cada parte do sistema automotivo, isso pode ser utilizado como base para rastreamento e modelo de futuras ferramentas de desenvolvimento e documentação de sistemas embarcados.

Neste trabalho foram estudados os sistemas embarcados no domínio da indústria automotiva, mas o método poderia ser considerado para outras aplicações, em outros domínios da indústria, considerando a existência de aplicações que também possuem um grau elevado de complexidade, com desafios similares. Extrapolando de forma geral a metodologia para o domínio da indústria aeronáutica,

conforme Figura 40, identificamos que de forma geral a adaptação do método para indústria aeronáutica é possível. No caso do fator relacionado à segurança funcional, ao invés de utilizar a norma ISO 26262, específica para indústria automotiva, pode-se aplicar a norma DO-178 B/C, específica para indústria aeronáutica.

Para o fator de complexidade, os métodos na maioria dos casos são iguais, considerando que o modelo em V é utilizado nos dois domínios, automotivo e aeronáutico. Já os fatores de relevância dos clientes e influências externas e do motorista, ou do piloto, no caso da indústria aeronáutica, são aspectos que precisam de uma avaliação de cada domínio, pois os operadores e o ambiente de utilização do produto são diferentes. Os blocos de especificações e mapeamentos podem ser utilizados na maioria dos casos no mesmo formato.

Figura 40 – Adaptação do método proposto na tese para o domínio da indústria aeronáutica.

Fonte: Elaborada pelo autor.

REFERÊNCIAS

ABELEIN, U.; LOCHNER, H.; HAHN, D.; STRAUB, S. **Complexity, quality and robustness** - The challenges of tomorrow's automotive electronics. Dresden, Germany: Design, Automation & Test in Europe Conference & Exhibition, pp. 870-871, 2012.

AFZAL, W.; ALONE, S.; GLOCKSIEN, K.; TORKAR, R. **Software test process improvement approaches: A systematic literature review and an industrial case study**. Journal of Systems and Software, v. 111, pp. 1-33, 2016.

AHMADIAN, M.; NAZARI, Z. J.; NAKHAEE, N.; KOSTIC, Z. **Model based design and SDR**. London, UK: 2.ed. IEE/EURASIP Conference on DSP Enabled Radio, pp. 19–99, 2005.

AHLUWALIA, J.; KRÜGER, I.; MEISINGER, M.; PHILLIPS, W. **Model-based run-time monitoring of end-to-end deadlines**. Jersey City, NJ, USA: Proceedings of the 5th ACM International Conference on Embedded Software, pp. 100-109, 2005.

AUTOSAR, Automotive Open System Architecture. **AUTOSAR consortium**. Disponível em: < www.autosar.org >. Acesso em: 01 Jan 2018.

AWEDIKIAN, R.; YANNOU, B. **A practical model-based statistical approach for generating functional test cases: application in the automotive industry**. **Software Testing, Verification and Reliability**, v. 24, n. 2, pp. 85-123, John Wiley & Sons, 2014.

BAHILL, A. T.; GISSING, B. **The Systems Engineering Process, Re-evaluating systems engineering concepts using systems thinking**. IEEE Transaction on Systems, Man and Cybernetics, Part C: Applications and Reviews, v. 28, 4. ed. pp. 516-527, 1998.

BARMI, Z. A.; EBRAHIMI, A. H; FELDT, R. **Alignment of requirements specification and testing: A systematic mapping study**. Berlin, Germany:

Proceedings of the 4th IEEE International Conference on Software Testing, Verification and Validation Workshops ICSTW, pp. 476-485, 2011.

BARTELS, A.; RUCHATZ, T.; BROSIG, S. **Intelligence in the Automobile of the Future**. In: Smart Mobile In-Vehicle Systems. New York: Springer, 2014. pp. 35-46

BEIZER, B. **Black-Box Testing: Techniques for Functional Testing of Software and Systems**. Hoboken: John Wiley & Sons, Inc., 1995.

BERTOZZI, M.; LUCA B.; BROGGI, A.; BUZZONI, M.; CARDARELLI, E.; CATTANI, S.; CERRI, P. VIAC. **An out of ordinary experiment**. Baden-Baden, Germany: Intelligent Vehicles Symposium (IV), IEEE, 2011, pp. 175-180.

BINDER, R. V. **Testing Object-oriented Systems, Object Technology Series**. Reading: Addison-Wesley, 1999.

BLOOS, M. **Echtzeitanalyse der Kommunikation**. In: KfZ-Bordnetzen auf Basis des CAN Protokolls. Dissertation, TU München, 1999.

BRAUN, P.; BROY, M.; HOUDEK, F.; KIRCHMAYR, M.; MÜLLER, M.; PENZENSTADLER, B. **Guiding requirements engineering for software-intensive embedded systems in the automotive industry**. Computer Science-Research and Development, v. 29, 1. ed. pp. 21-43, 2014.

BRINGMANN, E.; KRÄMER, A. **Model-based testing of automotive systems. Software Testing, Verification, and Validation**. 1st International Conference on. IEEE. pp. 485-493, 2008.

BOEHM, B. W. **Verifying and validating software requirements and design specifications**. IEEE software, v. 1, 1. ed. pp. 75, 1984.

BROGGI, A.; MEDICI, P.; CARDARELLI, E.; CERRI, P.; GIACOMAZZO, A.; FINARDI, N. **Development of the control system for the vislab intercontinental**

autonomous challenge. In: Intelligent Transportation Systems (ITSC), 13th International IEEE Conference, 2010, pp. 635-640.

BROY, M.; SLOTOCH, O. **From requirements to validated embedded systems.** In: International Workshop on Embedded Software. Berlin, Heidelberg: Springer, 2001 pp. 51-65.

BROY, M. **Challenges in automotive software engineering.** Shanghai, China: Proceedings of the 28th International Conference on Software Engineering. ACM, 2006, pp. 33–42.

BROY, M.; KRUGER, I. H.; PRETSCHNER, A.; SALZMANN, C. **Engineering automotive software.** Proceedings of IEEE, v. 95, 2. ed. pp. 356–373, 2007.

BUTENUTH, M., KALLWEIT, R., & PRESCHER, P. **Vehicle-in-the-Loop Real-world Vehicle Tests Combined with Virtual Scenarios.** ATZ worldwide, v. 119, 9. ed. pp. 52-55, Springer Fachmedien Wiesbaden, 2017.

CALABRESE, G. **R&D globalization in the car industry.** International Journal of Automotive Technology and Management, v. 1, 1. ed. pp. 145-159, 2001.

CHIESA, V. **Technology development control styles in multinational corporations: a case study.** Journal of Engineering and Technology Management, v. 16, 1. ed. pp. 191-206, Elsevier, 1999.

_____. **Global R&D Project Management and Organization: a Taxonomy.** Journal of Product Innovation Management, v. 17, 1. ed. pp. 341-359, Castellanza, Italy, 2000.

CHUNDURI, A.; FELDT, R.; ADENMARK, M. **An Effective Verification Strategy for Testing Distributed Automotive Embedded Software Functions: A Case Study.** In: International Conference on Product-Focused Software Process Improvement. Cham: Springer, 2016, pp. 233-248.

CLARK, K. B.; FUJIMOTO, T. **Product Development Performance: Strategy, Organization and Management in the World Auto Industry.** Boston: Harvard Business School Press, 1991.

CLARK, K. B.; WHEELWRIGHT, S. C. **Managing New Product and Process Development: Text and Cases.** New York: Free Press, 1993.

CLAUDIO, F.; LIMA, D. A. S.; DAMICO, F. Z.; MOTTA, F.; COELHO, L. T.; FRANCO, R. **Steps for requirements writing.** Instituto de Aeronáutica e Espaço (Institute of Aeronautics and Space). NAVCON Navegação e Controle, 2013

CLAUSING, D. **Total quality development: A step-by-step guide to world-class engineering,** 1994.

COLLOFELLO, J. S.; YANG, Z.; TVEDT, J. D.; MERRILL, D.; RUS, I. **Modeling software testing processes.** In: Computers and Communications. Conference Proceedings of the IEEE Fifteenth Annual International Phoenix Conference, Scottsdale, AZ, USA, 1996, pp. 289-293.

CONRAD, M. **A systematic approach to testing automotive control software.** SAE Technical Paper, 2004.

CONRAD, M. **Verification and Validation According to ISO 26262: A Workflow to Facilitate the Development of High-Integrity Software.** Natick: The MathWorks, Inc, 2012.

CONSONI, F. **Da tropicalização ao projeto de veículos: um estudo das competências em desenvolvimento de produtos nas montadoras de automóveis no Brasil.** Instituto de Geociências, Universidade Estadual de Campinas, Campinas, 2004.

COOPER, G. R. **Winning at new products.** Addison-Wesley Pub. Co, 1986.

_____. **Perspective: The Stage-Gate Idea-to-Launch Process – Update, What’s New and NexGen Systems.** The Journal of Product Innovation Management, v. 25, 3. ed. pp. 213-232, Wiley Online Library, Hamilton, ON, Canada, 2008.

_____. **Doing it right.** Ivey Business Journal, v. 64, 6. ed. pp. 54-60, GenSight, 2000.

COULON, D. **Whatever the Future of the Automotive Industry, Electronics is the Key.** Industry Conditions. Disponível em: <<http://www.ttiinc.com/object/me-coulon-20140909.html>>. Acesso em: 2014.

DAI, Z. R. **An Approach to Model-Driven Testing with UML 2.0, U2TP and TTCN-3.** PhD thesis, Technical University Berlin, Fraunhofer IRB Verlag, 2006.

DAJSUREN, Y.; VAN DEN BRAND, M. G. **Automotive System and Software Engineering:** State of the Art and Future Trends, Atlantis Press, 1. ed. 2017.

DEBOUK, R.; JOYCE, J. **ISO 26262 Hazard and Risk Assessment Methodology.** Minneapolis, Minnesota, 2010.

DIAS, A. V. C. **Produto mundial, engenharia brasileira:** integração de subsidiárias no desenvolvimento de produtos globais na indústria automobilística. Tese (Doutorado) - Universidade de São Paulo, 2003.

DIETER, G. E. **Engineering Design.** 2. ed. New York: McGraw Hill, 1990.

DRUFRESNE, T.; MARTIN, J. **Process Modeling for E-Business.** INFS 770 Methods for Information Systems Engineering: Knowledge Management and E-Business, Spring, Fairfax, Virginia, 2003.

DO-178B, Software Considerations in Airborne Systems and Equipments Certification, RTCA Inc. document, Dec/1992.

EWEN, D. **A Software Safety Certification Plug-in for Automated Code Generators Feasibility Study and Preliminary Design**. USRA/RIACS- NASA Ames Research Center, Moffett Field, CA 94035, Nov/2006.

FEITOSA, D.; FELIZARDO, K. R.; DE OLIVEIRA, L. B. R.; WOLF, D.; NAKAGAWA, E. Y. **Software engineering in the embedded software and mobile robot software development: a systematic mapping**. In: SEKE, 2010, pp. 738–741.

FENNEL, H.; BUNZEL, S.; HEINECKE, H.; BIELEFELD, J.; FÜRST, S. **Achievements and exploitation of the AUTOSAR development partnership**. SAE Technical Paper, Detroit, USA, 2006.

FILHO, J. T. **Gerenciando conhecimento**. Rio de Janeiro: SENAC, p. 77, 2000..

FIPS. **Integration definition for function modelling (IDEF0)**. Federal information processing standards publication 183, National Institute of standards and technology, USA, 1993

FLEMSTRÖM, D.; SUNDMARK, D.; AFZAL, W. **Vertical test reuse for embedded systems: A systematic mapping study**. In: Software Engineering and Advanced Applications (SEAA), 41st Euromicro Conference, IEEE, 2015, pp. 317-324. Funchal, Portugal, 2015

FRANCO, F.; MAURO M.; STEVAN S.; LUGLI A. B.; TORRES W. **Model based Functional safety for the embedded software of automobile power window system**. In: 11th IEEE/IAS International Conference on Industry Applications (INDUSCON), pp. 1–8. Juiz de Fora, Brazil, 2014.

FRISCHKORN, H. **Thinking in architecture**. dSpace News. Jan/2005.

FÜRST, S., BECHTER, M. **AUTOSAR for connected and autonomous vehicles: The AUTOSAR adaptive platform**. In: Dependable Systems and Networks Workshop, 46th Annual IEEE / IFIP International Conference, Toulouse, France, 2016, pp. 215-217.

GERLA, M.; LEE, E. K.; PAU, G.; LEE, U. **Internet of vehicles**: From intelligent grid to autonomous cars and vehicular clouds. Seoul, South Korea. In: Internet of Things (WF-IoT), World Forum on, IEEE, 2014, pp. 241-246.

GHAHAM, B. B. **Detail Process Charting: Speaking the Language of Process**. New Jersey, USA. John Wiley & Sons, 2004.

GRIMM, K. **Software technology in an automotive company**: major challenges. In: Proceedings of the 25th international conference on Software Engineering. IEEE Computer Society, 2003, pp. 498-503.

GRUSZCZYNSKI, B. **An overview of the current state of software engineering in embedded automotive electronics**. East Lansing, MI, USA. In: Electro/information Technology, International Conference, IEEE, 2006, pp. 377-381.

GUIMARÃES, A. A. **Knowledge Management in R&D Departments of Automotive Companies**. SAE BRAZIL TECHNICAL, 2011.

HACKBARTH, G. **The impact of organizational memory on IT systems**. South Carolina, USA. AMCIS Proceedings, 1998.

HAGHIGHATKHAH, A.; BANIJAMALI, A.; PAKANEN, O.; MARKKU, K. **Automotive software engineering**: A systematic mapping study. Journal of Systems and Software, v. 28, pp. 25-55, Elsevier, 2017.

HARDUNG, B.; KÖLZOW, T.; KRÜGER, A. **Reuse of Software** In: Distributed Embedded Automotive Systems. In Pisa, Italy. Proc. EMSOFT'04, 2004, pp. 203-210.

HARROLD, M. **Testing**: a roadmap. In: Proceedings of the Conference on the Future of Software Engineering. Limerick, Ireland, ACM, 2000, pp. 61-72.

HASEGAWA, M., & KANEKO, T. **Detailed Study of Hazard Analysis and Risk Assessment of ISO 26262 for Motorcycles**. Japan, SAE Technical Paper, 2017.

HOOD, C.; WIEDEMANN, S.; FICHTINGER, S.; PAUTZ, U. **Requirements management: The interface between requirements development and all other systems engineering processes**. Springer Science & Business Media, 2007.

HUBER, G. P. **Organizational learning**: The contributing processes and the literatures. *Organization science* 2.1, pp. 88-115, 1991.

HUDI, R. **Electronics are the core of most innovations**. Disponível em: <http://www.audi.com/com/brand/en/vorsprung_durch_technik/content/2015/08/Ricky-Hudi.html>. Acesso em: 2016.

HUMPHREY, J.; LECLER, Y.; SALERNO, M. S. **Global Strategies and Local Realities**: the Autoindustry in Emerging Markets. London: Macmillan Press, 2001.

HARVEY, M. G.; GRIFFITH, D. A. **The role of globalization, time acceleration, and virtual global teams in fostering successful global product launches**. *Journal of Product Innovation Management*, v. 24, n. 1, pp. 486-501, 2007.

INTERNATIONAL COUNCIL OF SYSTEMS ENGINEERING. Disponível em: <<http://www.incose.org>>. Acesso em: 2017.

ISO. **ISO 26262, Road vehicles-Functional safety**. International Standard ISO/FDIS, 2011.

IEC 61508. **Functional Safety of Electrical/Electronic/ Programmable Electronic Safety-Related Systems**. Electrotechnical Commission, 1999.

INTERNATIONAL SOFTWARE TESTING QUALIFICATION BOARD. **Standard glossary of terms used in Software Testing**. Version 1.2, Editor: van Veenendaal E., The Netherlands. IT Power Consultants, MEval, commercial.

J2980, Considerations for ISO 26262 ASIL Hazard Classifications.

JABLONSKI, T.; SCHUMANN, H.; BUSSE, C.; HAUSSMANN, H.; HALLMANN, U.; DREYER, D.; SCHÖTTLER, F. **Die neue elektromechanische Lenkung APA-BS**. ATZelextronik, v. 3, pp. 30–35, 2008.

JOHNSON, N. **Simply complexity**: A clear guide to complexity theory. Oneworld Publications, 2009.

JONES, L. M.; PITTS, B. M. **Successfully implementing the Stage-Gate NPD PROCESS**. Stage Gate Institute, 2010.

KAHN, R. S.; KOBAYASHI, S.; BEUTHE, M.; GASCA, J.; GREENE, D.; LEE, D.; MUROMACHI, Y.; NEWTON, P.; PLOTKIN, S.; SPERLING, S. **Transport and its infrastructure**. Eds. Climate Change, 2007.

KAMGA, J.; HERRMANN, J.; JOSHI, P. D-MINT. **Automotive case study** – Daimler, Deliverable 1.1, Deployment of model-based technologies to industrial testing. ITEA2 Project, Germany, 2007.

KARSAI, G.; SZTIPANOVITS, J.; LEDECZI, A.; BAPTY, T. **Model-integrated development of embedded software**. Proceedings of the IEEE, 91(1), pp. 145-164, 2003.

KASOJU, A.; PETERSEN, K.; MÄNTYLÄ, M. V. **Analyzing an automotive testing process with evidence-based software engineering**. Information and Software Technology, v. 55, 7. ed. pp. 1237-1259, 2013.

KATUMBA B.; KNAUSS E. **Agile development in automotive software development**: Challenges and opportunities. In: Product-Focused Software Process Improvement, Springer, 2014, pp. 33–47.

KHAN, M. E.; KHAN, F. **A Comparative Study of White Box, Black Box and Grey Box: Testing Techniques.** (IJACSA) International Journal of Advanced Computer Science and Applications, v. 3, n. 6, 2012.

KIM, B.; KASHIBA, Y.; DAI, S.; SHIRAISHI, S. **Testing autonomous vehicle software in the virtual prototyping environment.** IEEE Embedded Systems Letters, v. 9, 1, ed. pp. 5-8, 2017.

LARRUCEA, X.; WALKER, A.; COLOMO-PALACIOS, R. **Supporting the Management of Reusable Automotive Software.** IEEE Software, v. 34, 3. ed. pp. 40-47, 2017.

LAUGIER, C.; PHILIPPE, M.; URBANO, N. **Editorial for special issue on Perception and Navigation for Autonomous Vehicles.** IEEE Robotics and Automation Magazine, 2014.

LEE, S. H.; PARK, T. W.; CHUNG, K. H.; CHOI, K. H.; KIM, K. K.; MOON, K. H. **Requirement-based testing of an automotive ECU considering the behavior of the vehicle.** International Journal of Automotive Technology, v. 12, 1.ed. pp. 75-82, 2011.

LEHMANN, E.; KRÄMER, A. **Model-Based Testing of Automotive Systems.** In: Proceedings of IEEE ICST 08, Lillehammer, 2008.

LENNON, T. MASS, N. **Model-based design for mechatronic systems.** ELECTRONICS WORLD-SUTTON THEN CHEAM, 2008.

LEVESON, NANCY G. **Safeware: System Safety and Computers.** ACM, New York, 1995.

Liker, J. **The Toyota Way**, McGraw-Hill, 2004.

LIGGESMEYER, P.; TRAPP, M. **Trends in embedded software engineering. Software,** IEEE, v. 26, 3. ed. pp. 19–25, 2009.

LUTSEY N; SPERLING D. **Regulatory adaptation accommodating electric vehicles in a petroleum world**. Energy Policy, vol. 45, pp. 308-16, 2012.

MARO, S.; STEGHÖFER, J. P.; STARON, M. **Software Traceability in the Automotive Domain: Challenges and Solutions**. Journal of Systems and Software. v. 141, p 85-11, 2018.

MATHWORK, The MathWorks. Matlab/ Simulink, Inc. Disponível em: <www.mathworks.com>. 2016.

MAUBORGNE, P.; DENIAUD, S.; LEVRAT, E.; BONJOUR, E.; MICAËLLI, J. P.; LOISE, D. **Operational and system hazard analysis in a safe systems requirement engineering process—Application to automotive industry**. Safety science, v. 87, pp. 256-268, 2016.

MCDONALD, J.; MURRAY, L.; LINDSAY, P.; STROOPER, P. **Module testing embedded software-an industrial pilot project**. In: Engineering of Complex Computer Systems, Proceedings. Seventh IEEE International Conference on, pp. 233-238, 2001.

MIL-STD-882D: System Safety Program Requirements. US Department of Defence. Defence Standard, Feb/2000.

MILLER, R. **Global R&D networks and large-scale innovations**: the case of the automobile industry. Research Policy, v. 23, n. 1. pp. 27-46, 1994.

MISRA-C, Motor Industry Software Reliability Association. **Guidelines for the Use of the C Language in Vehicle Based Software**, 1998.

MISRA-C, Motor Industry Software Reliability Association. **Guidelines for the use of the C language in critical systems**, 2004.

MUDAMBI, R.; MUDAMBI, S. M.; NAVARRA, P. **Global Innovation in MNCs: The effects of subsidiary self-determination and teamwork.** Journal of Product and Innovation Management, v. 24, n. 1, pp. 442-455, 2007.

MYERS, G. J.; SANDLER, C.; BADGETT, T. **The art of software testing.** John Wiley & Sons, 2011.

NASLAVSKY, L.; ALSPAUGH, T. A.; RICHARDSON, D. J.; ZIV, H. **Using scenarios to support traceability.** In: Proceedings of the 3rd International Workshop on Traceability in emerging forms of Software Engineering, ACM, 2005, pp. 25–30.

NEUKIRCHEN, H. W. **Languages, tools and patterns for the specification of distributed real-time tests.** 2004. Tese (Doutorado) - Georg August Universidade, Gottingen, 2004.

NICOLESCU, G.; MOSTERMAN, P. J. **Model-Based Design for Embedded Systems.** Computational Analysis, Synthesis, and Design of Dynamic Systems. Boca Raton: CRC Press, 2009.

NONAKA, I.; TAKEUCHI, H.; **The Knowledge-Creating Company,** Oxford: Oxford University Press, 1995.

NÖRENBERG, R.; REISSING, R.; WEBER, J. **ISO 26262 Conformant Verification Plan.** Conference Informatik: Service Science - Neue Perspektiven für die Informatik, v.2, pp. 515-520, 2010.

OOI, S. M.; LIM, R.; LIM, C. C. **An integrated system for end-to-end traceability and requirements test coverage.** In: Proceedings of the 5th IEEE International Conference on Software Engineering and Service Science (ICSESS), pp. 45-48, 2014.

OPPAT, K. **Disseminative Capabilities**: a case study of collaborative product development in the automotive industry. Schriften zum europäischen Management, Gabler Edition Wissenschaft, 2008.

PANARONI, P.; SARTORI, G.; FABBRINI, F.; FUSANI, M.; LAMI, G.. **Safety in automotive software**: an overview of current practices. In: Computer Software and Applications. 32nd Annual IEEE International, pp. 1053-1058, 2008.

PARK, G.; KU, D.; LEE, S.; WON, W. J.; JUNG, W. **Test methods of the autosar application software components**. ICCAS-SICE, IEEE, pp. 2601-2606, 2009.

PAHL, G.; BEITZ, W. **Engineering design**: a systematic approach. Springer Science Business Media, 2013.

PREE, W.; TEMPL, J. **Modeling with the timing definition language (TDL)**. In: Automotive Software Workshop, Springer Berlin Heidelberg, 2006, pp. 133-144.

PELLICCIONE, P.; KNAUSS, E.; HELDAL, R.; ÅGREN, S. M.; MALLOZZI, P.; ALMINGER, A.; BORGENTUN, D. **Automotive architecture framework**: the experience of volvo cars. Journal of Systems Architecture, v. 77, pp. 83-100. 2017.

PENDLETON, S. D.; ANDERSEN, H.; DU, X.; SHEN, X.; MEGHJANI, M.; ENG, Y. H.; ANG, M. H. **Perception, planning, control, and coordination for autonomous vehicles**. Machines. Machines, v. 5, 1. Ed. n. 6, 2017.

PELZ, G.; OEHLER, P.; FOURGEAU, E.; GRIMM, C. **Automotive system design and autosar**. In: Advances in Design and Specification Languages for SoCs, pp. 293-305, Boston: Springer, 2005.

PÉREZ, A. M.; KAISER, S. **Integrating test levels for embedded systems**. Testing: Academic and Industrial Conference-Practice and Research Techniques, TAIC PART'09, IEEE, pp. 184-193, 2009.

_____. **Bottom-up reuse for multi-level testing.** Journal of Systems and Software, v. 83, 12. ed. pp. 2392-2415, 2010.

POST, H.; SINZ, C.; MERZ, F.; GORGES, T.; KROPF, T. **Linking functional requirements and software verification.** Requirements Engineering Conference, RE'09, 17th IEEE International, pp. 295-302, 2009.

PRASAD, B. **Concurrent engineering fundamentals.** Englewood Cliffs, New Jersey: Prentice Hall, 1996.

PRETSCHNER, A.; BROY, M.; KRUGER, I. H.; STAUNER, T. **Software engineering for automotive systems: a roadmap.** 2007 Future of Software Engineering, IEEE Computer Society, pp. 55-71, 2007.

PRETSCHNER, A.; PRENNINGER, W.; WAGNER, S.; KÜHNEL, C.; BAUMGARTNER, M.; SOSTAWA, B.; ZÖLCH, R.; STAUNER, T. **One evaluation of model-based testing and its automation.** Proceedings of the 27th International Conference on Software Engineering, ACM, pp. 392–401, 2005.

PUGH, S. **Total Design.** Reading: Addison-Wesley Publishers, 1991.

PUGH, S.; CLAUSING, D. **Creating Innovative Products Using Total design: The Living Legacy of Stuart Pugh,** Boston: Publisher Addison-Wesley Longman Publishing Co., Inc., 1996.

QUINTELLA, H. L. M. M.; ROCHA, H. M. **Nível de maturidade e comparação dos PDPs de produtos automotivos.** Revista Produção, v. 17, 1. ed. pp. 199-217, 2007.

RAN, Z.; YAN, H.; ZHANG, H.; LI, Y. **Approximate optimal AUTOSAR software components deploying approach for automotive E/E system.** International Journal of Automotive Technology, v. 18, 6, ed. pp. 1109-1119, 2017.

RANA, R.; STARON, M.; BERGER, C.; HANSSON, J.; NILSSON, M.; TÖRNER, F. et al. **Selecting software reliability growth models and improving their**

predictive accuracy using historical projects data. Journal of Systems and Software, v. 98, pp. 59-78, 2014.

REEDY, J.; LUNZMAN, S. **Model Based Design Accelerates the Development of Mechanical Locomotive Controls.** SAE Technical Paper, 2010.

REICHART, G.; HANEBERG, M. **Key drivers for a future system architecture in vehicles.** Detroit: Convergence Transportation Electronics Association, 2004.

REMPEL, P.; MADER, P.; KUSCHKE, T. **An empirical study on project specific traceability strategies.** Proceedings of the 21st IEEE International Requirements Engineering (RE) Conference, pp. 195–204, 2013.

ROS, J; NAGELHOUT, D.; & MONTFOORT, J. **New environmental policy for system innovation:** Casus alternatives for fossil motor fuels. Applied Energy, v. 86, 2. ed. pp. 243-250, 2009.

ROZENFELD, H.; FORCELLINI, F. A; AMARAL, D. C. **Gestão de desenvolvimento de produtos:** uma referência para a melhoria do processo. Editora Saraiva, 2006.

RUSHTON, G.; MERCHANT, V. **Vehicle electrical/electronic system design considerations.** Warrendale: Society of Automotive Engineers, Inc., 2000.

SABALIAUSKAITE, G.; LOCONSOLE, A.; ENGSTRÖM, E.; UNTERKALMSTEINER, M.; REGNELL, B.; RUNESON, P. **Challenges in aligning requirements engineering and verification in a large-scale industrial context.** In: International Working Conference on Requirements Engineering: Foundation for Software Quality, Springer Berlin Heidelberg 2010, pp. 128-142.

SARGEANT, J.; LOCKYER, J.; MANN, K.; HOLMBOE, E.; SILVER, I.; ARMSON, et al. **Facilitated reflective performance feedback:** developing an evidence-and theory-based model that builds relationship, explores reactions and content, and coaches for performance change, Academic Medicine, v. 90, 12. ed. pp. 1698-1706, 2015.

SCHÄUFFELE, J.; ZURAWKA, T. **Automotive Software Engineering–Grundlagen, Prozesse, Methoden und Werkzeuge effizient einsetzen**. 3. Aufl, Friedr, Wiesbaden: Vieweg & Sohn, 2006.

SCHLAGER, J. **Systems engineering**: key to modern development. IRE Transactions v. 3, 3, ed. pp. 64–66, JUL/1956.

SCHUBERT, P.; DAVID-MICHAEL, L.; BEAT, S. **A global knowledge medium as a virtual community**: the Net Academy concept. AMCIS Proceedings: 207, 1998.

SEGISMUNDO, A.; MIGUEL, P. A. C. **Análise da descentralização no desenvolvimento de novos produtos**: uma investigação em uma empresa do setor de veículos comerciais. v. 24, 1. ed. p. 118-131, 2014.

SOMÉ, S. S. **Supporting use case based requirements engineering**. Information and Software Technology, v. 48, 1. ed. pp. 43-58, 2006.

SPILLNER, A.; LINZ, T.; ROSSNER, T.; WINTER, M. **Software testing practice**: Test management: A study guide for the certified tester exam ISTQB advanced level. Rocky Nook, Inc., 2007.

STARON, M. **Automotive Software Architectures**: An Introduction. 1. ed. Springer, 2017.

STRICHMAN, O., TZOREF-BRILL, R., “Hardware and Software: Verification and Testing”. In 13th International Haifa Verification Conference, Haifa, Israel, v. 10629, 1st ed, Springer, 2017.

SUBRAMANIAM, M. **Integrating cross-border knowledge for transnational new product development**. Journal of Product Innovation Management, v. 23, 1. ed., p. 541-555, 2006.

SUNDMARK, D.; PETERSEN, K.; LARSSON, S. **An exploratory case study of testing in an automotive electrical system release process**. Industrial Embedded Systems (SIES), 6th IEEE International Symposium, pp. 166-175, 2011.

TAN, A. H. **Text mining**: The state of the art and the challenges. In: Proceedings of the PAKDD 1999 Workshop on Knowledge Discovery from Advanced Databases, v. 8, 1999, pp. 65-70.

THANGARAJAH, J.; JAYATILLEKE, G.; PADGHAM, L. **Scenarios for system requirements traceability and testing**, In Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), v. 1, pp. 285–292, 2011.

THOMAS, C. **Transportation options in a carbon-constrained world**: hybrids, plug-in hybrids, biofuels, fuel cell electric vehicles, and battery electric vehicles. Int J Hydrogen Energy, v. 34, pp. 9279–9296, 2009.

TOWNSEND, D. J.; CATANTONE, J. R.; **Evolution and Transformation of Innovation in the Global**. Journal of Product Innovation Management, v. 31, pp. 4-7, 2013.

VAN, E. H. **QI**: Assessment of the ISO 26262 Standard, Road Vehicles – Functional Safety, SAE Government/Industry Meeting, p. 9, 2012.

VON KROGH, G. **Care in knowledge creation**. California management review, v. 40, pp. 133-153, 1998.

WARD, D. D.; CROZIER, S. E. **The uses and abuses of ASIL decomposition in ISO 26262**. In: System Safety, incorporating the Cyber Security Conference, 7th IET International Conference on, OUT/2012, pp. 1-6.

WEBER, M.; WEISBROD, J. **Requirements engineering in automotive development** - Experiences and challenges. Proceedings of the IEEE Joint International Conference on Requirements Engineering, pp. 331–340, 2002.

WEBER, J. **Automotive Development Processes: Processes for Successful Customer Oriented Vehicle Development**. Springer-Verlag Berlin Heidelberg, 2009.

WEILKIENS, T. **Systems Engineering with SysML/UML: Modeling, Analysis, Design**, Elsevier, 2011.

WEEKS, J. J. **Automotive Engine Modeling for Real-Time Control Using Matlab/simulink**. *SAE transactions*, pp. 295-309, 1995

WEYUKER, E. **The Evaluation of Program-Based Software Test Data Adequacy Criteria**. In: Communications of the ACM, v. 31-6, pp. 668–675, ACM, New York, 1988.

WINNER, H.; PROKOP, G.; MAURER, M. **Automotive Systems Engineering II**. Springer; 1. ed. 2018.

ZANDER, J.; SCHIEFERDECKER, I.; MOSTERMAN, P. J. **A taxonomy of model-based testing for embedded systems from multiple industry domains**. Model-based testing for embedded systems, pp. 3-22, 2011.

ZANDER, J.; SCHIEFERDECKER, I.; MOSTERMAN, P. J. **Model-based testing for embedded systems**. CRC Press, 2017.

ZANDER-NOWICKA, J. **Model-Based Testing of Embedded Systems in the Automotive Domain**. Tese (PhD, Technical University Berlin, Germany, 2009.

ZAYED, J. M.; NOUVEL, N.; RAUWALD, U.; SCHERMAN, O. A. **Chemical complexity supramolecular self-assembly of synthetic and biological building blocks in water**. Chemical Society Reviews, v. 39, 8. ed. pp. 2806-2816, 2010.

ZHAO, Y. **Telematics**: safe and fun driving. IEEE Intelligent systems, v. 17, 1. ed., pp. 10-14, 2002.