CARLOS HENRIQUE NETTO LAHOZ

ELICERE
O PROCESSO DE ELICITAÇÃO DE METAS DE DEPENDABILIDADE
PARA SISTEMAS COMPUTACIONAIS CRÍTICOS
Estudo de Caso Aplicado à Área Espacial

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

São Paulo

2009

CARLOS HENRIQUE NETTO LAHOZ

ELICERE
O PROCESSO DE ELICITAÇÃO DE METAS DE DEPENDABILIDADE
PARA SISTEMAS COMPUTACIONAIS CRÍTICOS
Estudo de Caso Aplicado à Área Espacial

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

Área de Concentração:
Engenharia de Computação e Sistemas
Digitais

Orientador: Professor Doutor Livre-
Docente João Batista Camargo Júnior

São Paulo
2009

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

São Paulo, 31 de agosto de 2009.

Assinatura do autor _____

Assinatura do orientador _____

FICHA CATALOGRÁFICA

Lahoz, Carlos Henrique Netto

ELICERE: o processo de elicitação de metas de dependabilidade para sistemas computacionais crítico: estudo de caso aplicado a área espacial / C.H.N. Lahoz. -- ed.rev. -- São Paulo, 2009. 225 p.

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

1. Engenharia de requisitos 2. Segurança de computadores 3. Confiabilidade de Software 4. Computadores digitais I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II. t.

Ao querido Ary, meu maior exemplo de superação.

AGRADECIMENTOS

A Deus, por iluminar meu caminho.

Ao amigo e orientador João Batista Camargo Júnior, pelo constante apoio à realização deste trabalho com um entusiasmo fora de série.

Aos amigos do Grupo de Análise de Segurança – GAS, em especial aos professores Jorge Rady de Almeida Júnior e Selma Shin Shimizu Melnikoff, pela convivência ao longo de todos esses anos.

Aos colegas do Instituto de Aeronáutica e Espaço – IAE, e principalmente a Martha Adriana Dias Abdala pelas inúmeras vezes em que me ajudou na produção do conhecimento científico.

A todos os meus familiares, pela compreensão e paciência.

A doce Beatriz, que mantém minha vida maravilhosamente complexa, ainda que infinitamente simples.

Estabelecer um sentimento de simplicidade em design
requer tornar a complexidade conscientemente disponível
de alguma forma explícita.

John Maeda - MIT Simplicity Consortium

RESUMO

Os avanços tecnológicos na eletrônica e no software têm sido rapidamente assimilados pelos sistemas computacionais demandando novas abordagens para a engenharia de sistemas e de software prover produtos confiáveis, sob critérios bem estabelecidos de qualidade. Dentro deste contexto, o processo de elicitação de requisitos tem um papel estratégico no desenvolvimento de projetos. Problemas na atividade de elicitação contribuem para produzir requisitos pobres, inadequados ou mesmo inexistentes que podem causar a perda de uma missão, desastres materiais e financeiros, a extinção prematura de um projeto ou promover uma crise organizacional. Esta tese apresenta o processo de elicitação de metas de dependabilidade, chamado ELICERE, aplicado em sistemas computacionais críticos, que se fundamenta nas técnicas de engenharia de requisitos orientada a metas, chamada i^* , e nas técnicas de engenharia de segurança HAZOP e FMEA, que identificam e analisam perigos operacionais de um sistema. Depois de criar os modelos do sistema usando os diagramas i^* , eles são analisados através de palavras-guia baseadas no HAZOP e FMEA, de onde as metas relacionadas à dependabilidade são extraídas. Através desta abordagem interdisciplinar, ELICERE promove a identificação de metas, que atendam aos requisitos de qualidade, relativos à dependabilidade, para sistemas computacionais críticos ainda na fase de concepção de um projeto. A abordagem do estudo de caso é baseada em um estudo qualitativo e descritivo de um caso único, usando o projeto de um foguete lançador hipotético, chamado V-ALFA. A aplicação do ELICERE neste projeto espacial teve a intenção de aperfeiçoar as atividades de engenharia de requisitos do sistema computacional do Veículo Lançador de Satélites Brasileiro, e também como forma de explicar como o processo ELICERE funciona.

Palavras-chave: Dependabilidade. Metas. Elicitação. Sistemas computacionais críticos. Projeto espacial.

ABSTRACT

The technological advances in electronic and software have been rapidly assimilated by computer systems, demanding new approaches for software and system engineering to provide reliable products, under well-known quality criteria. In this context, requirements engineering has a strategic role in project development. Problems in the elicitation activity contribute to producing poor, inadequate or even non-existent requirements that can cause mission losses, material or financial disasters, premature project termination or promote an organizational crisis. This thesis introduces the dependability goals elicitation process, called ELICERE, applied to critical computer systems based on a goal-oriented requirement engineering technique, called i^* , and the safety engineering techniques HAZOP and FMEA, which will be applied for the identification and analysis of operational risks of a system. After creating the system models using i^* diagrams, they are analyzed through guidewords based on HAZOP and FMEA, from which goals related to dependability are extracted. Through this interdisciplinary approach, ELICERE promotes the identification of goals that meet the quality requirements, related to dependability for critical systems, still in the project conception phase. The case study approach is based on a qualitative and descriptive single-case, using a computer system project of a hypothetical launching rocket, called V-ALFA. The ELICERE application in this space project intends to improve the requirement engineering activities in the computer system of the Brazilian Satellite Launch Vehicle, and also a way to explain how the ELICERE process works.

Keywords: Dependability. Goals. Elicitation. Critical computer systems, Space project.

LISTA DE FIGURAS

Figura 2.1 – Contexto conceitual da abordagem do processo ELICERE	30
Figura 2.2 – Processo de ER	32
Figura 2.3 – Os envolvidos no processo de ER	34
Figura 2.4 – Relação de dependência entre dois atores	43
Figura 2.5 – Técnicas de análise de perigos no ciclo de vida do produto	48
Figura 2.6 – Família de lançadores do programa Cruzeiro do Sul	73
Figura 3.1 – Processo ELICERE de definição de metas de dependabilidade	78
Figura 3.2 – Seqüência de atividades do processo ELICERE	81
Figura 3.3 – Metas de missão do CD com o próprio CD, com o SA e com o TE	89
Figura 3.4 – Recursos utilizados para a comunicação do CD com o AS	90
Figura 3.5 – Relação de dependência de tarefas entre os atores SI e CD	91
Figura 3.6 – Relação de dependência de meta-soft entre os atores AS e Op_AS.....	92
Figura 3.7 – Relação de dependência compromissada entre os atores CD, TE e TS	93
Figura 3.8 – Relação de dependência crítica entre os atores SS, TE e CD	94
Figura 3.9 – Tarefas internas do ator CD com relação à meta “apontar veículo”	95
Figura 3.10 – O modelo SR do ator CD com relação à tarefa “fornecer dados telemetria”	96
Figura 3.11 – Metas meio da tarefa “analisar modo falha” e suas tarefas fim	98
Figura 3.12 – Decomposição de tarefa “executar basculamento”	99
Figura 3.13 – Árvore de palavras-guia de “meta não atingida”	113
Figura 3.14 – Árvore de palavras-guia de “meta incorreta”	114
Figura 3.15 – Árvore de palavras-guia de “meta adicional”	114
Figura 3.16 – Árvore de palavras-guia de “meta fora do tempo/ordem”	115
Figura 4.1 – Exemplo de sistema de controle de um veículo espacial	136
Figura 4.2 – Modelo SD do sistema CD relacionado à visão de metas.....	143
Figura 4.3 – Modelo SD do sistema CD relacionado à visão de recursos.....	144
Figura 4.4 – Modelo SR da meta “fornecer dados inerciais”	151
Figura 4.5 – Árvore da palavra-guia “meta não atingida” de “fornecer dados inerciais”	152

Figura 4.6 – Questionário de “meta não atingida” de “fornecer dados inerciais”	153
Figura 4.7 – Questionário de “recurso ausente” de “deltaAlfa, deltaBeta, deltaEpsilon”	154
Figura 4.8 – Questionário de “omissão da tarefa” de “adquirir incrementos de ângulos de atitude”	155
Figura 4.9 – Modelo SR da meta “fornecer telemédidas”	156
Figura 4.10 – Árvore da palavra-guia “meta incorreta” de “fornecer telemédidas” ..	157
Figura 4.11 – Questionário de “meta incorreta” de “fornecer telemédidas”	158
Figura 4.12 – Questionário de “recurso incorreto” de “dados vôo”	159
Figura 4.13. – Questionário de “incorreção na tarefa” de “adquirir e condicionar dados”	160
Figura 4.14 – Modelo SR da meta “preparar para vôo”	161
Figura 4.15 – Árvore da palavra-guia “meta incorreta” de “preparar para vôo”	162
Figura 4.16 – Questionário de “meta incorreta” de “preparar para vôo”	163
Figura 4.17 – Questionário de “recurso incorreto” de “perfil de trajetória”	164
Figura 4.18 – Questionário de “incorreção na tarefa” de “carregar perfil da trajetória”	165
Figura 4.19 – Modelo SR da meta “controlar atitude e rolamento 1E”	166
Figura 4.20 – Árvore da palavra-guia “meta fora do tempo/ordem” de “controlar atitude e rolamento 1E”	168
Figura 4.21 – Questionário de “meta fora do tempo/ordem” de “controlar atitude e rolamento 1E”	169
Figura 4.22 – Questionário de “recurso fora do tempo/ordem” de “TM1Ea, TM1Eb, TM1Ec, TM1Ed”	170
Figura 4.23 – Questionário de “tarefa fora do tempo/ordem” de “selecionar sistema de atuação”	171
Figura 4.24 – Modelo SR da meta “controlar rolamento 2E”	172
Figura 4.25 – Árvore da palavra-guia “meta adicional” de “controlar rolamento 2E” ..	173
Figura 4.26 – Questionário de “meta adicional” de “controlar rolamento 2E”	174
Figura 4.27 – Questionário de “recurso adicional” de “MD1, MD2, MD3, MD4”	175
Figura 4.28 – Questionário de “incorreção na tarefa” de “comandar atuadores”	176
Figura 4.29 – Modelo SR da meta “garantir segurança de vôo”	177

Figura 4.30 – Árvore da palavra-guia “meta fora do tempo/ordem de “garantir segurança de voo”	178
Figura 4.31 – Questionário de “meta fora do tempo/ordem” de “garantir segurança de voo”	179
Figura 4.32 – Questionário de “recurso fora do tempo/ordem” de “telecomando” ...	180
Figura 4.33 – Questionário de “tarefa fora do tempo/ordem” de “interromper voo” .	181
Figura 4.34 – Modelo SR da meta “habilitar modo de operação”	182
Figura 4.35 – Árvore da palavra-guia “meta não atingida” de “habilitar modo de operação”	183
Figura 4.36 – Questionário de “meta não atingida” de “habilitar modo de operação”	184
Figura 4.37 – Questionário de “incorreção na tarefa” de “tratar interrupção da sequência automática”	186

LISTA DE TABELAS

Tabela 2.1 – Interpretação de atributos por palavras-guia	54
Tabela 3.1 – Palavras-guia para meta	104
Tabela 3.2 – Palavras-guia para recurso	106
Tabela 3.3 – Palavras-guia para tarefa	108
Tabela 3.4 – Palavras-guia para meta-soft	110
Tabela 3.5 – Comparativo de risco aceitável.....	125
Tabela 3.6 – Matriz de probabilidade aceitável de ocorrência do risco	127
Tabela 3.7 – Relacionamento entre interesses e requisitos não funcionais.....	134
Tabela 4.1 – Classificação de severidades	147
Tabela 4.2 – Risco aceitável	148
Tabela 4.3 – Probabilidade aceitável de ocorrência do risco	149
Tabela 4.4 – Propriedades mensuráveis relacionadas a requisitos não funcionais	191

SUMÁRIO

1 INTRODUÇÃO	15
1.1 CONTEXTO E JUSTIFICATIVA DO TRABALHO	17
1.2 OBJETIVO.....	21
1.3 ABORDAGEM ADOTADA	22
1.4 ESTADO DA ARTE	24
1.5 CONTRIBUIÇÕES DA TESE	26
1.6 HISTÓRICO DA PESQUISA DA TESE.....	27
2 ENGENHARIA DE REQUISITOS E ENGENHARIA DE SEGURANÇA .	
.....	30
2.1 ENGENHARIA DE REQUISITOS.....	31
2.1.1 Processo de engenharia de requisitos e os envolvidos.....	33
2.1.2 Engenharia de requisitos orientada a metas	36
2.1.3 Metas e requisitos	39
2.1.4 Técnica de modelagem organizacional i*.....	40
2.1.4.1 Modelo de dependência estratégica.....	42
2.1.4.2 Modelo de razão estratégica	45
2.2 ENGENHARIA DE SEGURANÇA	46
2.2.1 Técnica HAZOP	50
2.2.2 Palavras-guia HAZOP.....	52
2.2.3 Técnica FMEA.....	55
2.2.4 Palavras-guia no contexto do HAZOP e FMEA.....	56
2.3 SISTEMAS CRÍTICOS DA ÁREA ESPACIAL.....	59
2.3.1 Sistemas espaciais	61

2.3.2 A engenharia de sistemas e de software espacial	63
2.3.2.1 Engenharia de sistemas espacial	63
2.3.2.2 Engenharia de software espacial	67
2.3.3 Foguetes espaciais no Brasil	70
2.3.4 Software em foguetes espaciais no Brasil	73
2.4 CONSIDERAÇÕES	75
3 ELICERE: O PROCESSO DE ELICITAÇÃO DE METAS DE DEPENDABILIDADE APLICADO A SISTEMAS CRÍTICOS	76
3.1 MODELO DO PROCESSO	78
3.2 SEQÜÊNCIA DAS ATIVIDADES DO PROCESSO	79
3.3 APLICAÇÃO DO I* NO PROCESSO ELICERE	84
3.3.1 Modelo SD para o processo ELICERE	87
3.3.1.1 Tipos de dependências do modelo SD para o processo ELICERE	88
3.3.1.2 Relações de dependência do modelo SD	92
3.3.2 Modelo SR para o processo ELICERE	94
3.3.2.1 Tipos de relacionamento do modelo SR	97
3.4 PALAVRAS-GUIA PARA O PROCESSO ELICERE	100
3.4.1 Adequação das palavras-guia para o processo ELICERE	102
3.4.2 Estrutura semântica das palavras-guia	111
3.4.3 Questionário das palavras-guia	116
3.4.3.1 Item tipo da falha	117
3.4.3.2 Item modo de falha	119
3.4.3.3 Item pior nível de perigo esperado	121
3.4.3.4 Item severidade da falha	122
3.4.3.5 Item meta-soft	123
3.5 CONSIDERAÇÕES	132
4 ESTUDO DE CASO DO PROJETO V-ALFA	135

4.1 O VEÍCULO LANÇADOR DE SATÉLITES V-ALFA.....	136
4.2 A MODELAGEM DO SISTEMA COMPUTACIONAL DO V-ALFA	139
4.2.1 Os modelos SD do veículo V-ALFA	140
4.2.2 Os modelos SR do veículo V-ALFA	145
4.3 A PREPARAÇÃO PARA A APLICAÇÃO DAS PALAVRAS-GUIA	145
4.4 CONSIDERAÇÕES PRELIMINARES SOBRE A DEFINIÇÃO DE METAS-SOFT	147
4.5 A APLICAÇÃO DAS PALAVRAS-GUIA NO MODELO DO V-ALFA	149
4.5.1 A aplicação de palavras-guia na meta “fornecer dados inerciais”	150
4.5.2 A aplicação de palavras-guia na meta “fornecer telemedidas”	156
4.5.3 A aplicação de palavras-guia na meta “preparar para vôo”	161
4.5.4 A aplicação de palavras-guia na meta “controlar atitude e rolamento 1E”	166
4.5.5 A aplicação de palavras-guia na meta “controlar rolamento 2E”	172
4.5.6 Aplicação de palavras-guia na meta “garantir segurança de vôo”	177
4.5.7 Aplicação de palavras-guia na meta “habilitar modo de operação”	182
4.6 CONSIDERAÇÕES	187
5 CONSIDERAÇÕES FINAIS	192
5.1 ABORDAGEM CONCEITUAL DO PROCESSO ELICERE	193
5.2 ABORDAGEM PRÁTICA DO PROCESSO ELICERE	195
5.3 CONTRIBUIÇÕES DO PROCESSO ELICERE	201
5.4 TRABALHOS FUTUROS.....	204
5.5 CONCLUSÕES FINAIS.....	206
REFERÊNCIAS	209
REFERÊNCIAS BIBLIOGRÁFICAS COMPLEMENTARES.....	224

1 INTRODUÇÃO

Foi desenvolvido nesta pesquisa de doutorado um processo de elicitación de metas de dependabilidade, chamado ELICERE, que visa contribuir para a melhoria do grau de qualidade de sistemas computacionais críticos. Dependabilidade abrange as características relacionadas à segurança, confiabilidade e outros fatores de qualidade relacionados à confiança no funcionamento de um produto. Por sistemas computacionais críticos entendem-se os sistemas críticos com relação à segurança (do inglês *safety critical system*), também conhecidos como sistemas de alta integridade (uma tradução livre do inglês *high-integrity systems*). Este tipo de sistema é requerido para aplicações onde falhas podem ter severas consequências que ameacem ou provoquem a perda de vidas humanas, danos ao ambiente, perdas materiais e financeiras e comprometimento da missão para qual o sistema foi designado.

Através da adoção deste processo pretende-se identificar, antecipadamente em projetos, as possíveis fragilidades de segurança contra falhas acidentais, intencionais ou até mesmo outros fatores ligados à falta de confiança no funcionamento de um sistema crítico. Inicialmente, o modelo do processo utiliza uma adaptação da técnica de modelagem *i** (do inglês, *istar*) para sistemas críticos, ainda nos estágios iniciais de concepção de um sistema. Utiliza também os recursos de técnicas de análise de perigos baseados nas palavras-guia dos métodos HAZOP (do inglês, *HAZard and OPerability analysis*) e FMEA (do inglês, *Failure Modes and Effects Analysis*), que são adaptadas para a identificação e análise de perigos operacionais do sistema em questão.

Vale ressaltar que a palavra elicitación é um anglicismo derivado da palavra *elicitation*, derivado do verbo *elicit*, que significa obter informações ou reações. Este termo ainda não foi oficializado formalmente pela língua portuguesa, mas é usado freqüentemente para descrever a atividade de obtenção de requisitos (GALETTI; SPINOLA, 2005). As palavras eliciar, capturar e elucidar também são bastante utilizadas como tradução de *elicitation*, em português. Devido à sua ampla utilização

pela comunidade da área de engenharia de requisitos no Brasil, o termo elicitar será adotado diretamente no texto. O nome ELICERE, adotado para o processo de elicitação proposto na tese, foi escolhido por ser o verbo original em latim que significa fazer sair, extrair (FARIA, 1962).

A palavra dependabilidade também não existe formalmente na língua portuguesa. É outro anglicismo derivado da palavra *dependable*, que significa confiável. É adotado por pesquisadores e até por regulamentações Brasileiras (ABNT-NBR-14857-2, 2002), com um sentido bastante amplo, abrangendo diversas características relacionadas à segurança e à confiabilidade de um produto. Este termo vem também sendo adotado como substituto para tolerância a falhas pela comunidade acadêmica. Weber (2002) enfatiza que, tanto o evento da Sociedade Brasileira de Computação, o Simpósio Brasileiro de Tolerância a Falhas (SCTF), quanto o evento internacional *Fault Tolerant Computing Symposium* (FTCS), adotaram a denominação *dependable*. O SCTF em 2000 foi rebatizado para *Dependable Systems and Networks* (DNS), e o FTCS em 2003 passou a ser chamado de *Latin American Dependable Computing* (LADC).

O conceito de dependabilidade, neste trabalho, é usado como a habilidade de entregar um serviço de forma confiável. Mais especificamente, é o grau de confiança que se deposita em um sistema computacional de modo a dar uma garantia justificável ao serviço a ser entregue. Segundo Laprie apud Barbacci et al. (1995), dependabilidade é composto pelas características não funcionais do sistema que justificam atribuir confiança no seu funcionamento.

Como motivação e base para aplicação do processo ELICERE foi utilizado como estudo de caso o sistema computacional de um veículo hipotético lançador de satélites intitulado V-ALFA. Este veículo hipotético baseia-se no primeiro de uma família de veículos lançadores de satélite do Programa Cruzeiro do Sul. Esta família de foguetes foi apresentada como parte do Programa Espacial Brasileiro em outubro de 2005, e tem a ambição de capacitar o Brasil na tecnologia de lançadores de maior porte (atingindo órbitas de um mil quilômetros e com uma carga útil de duas toneladas) e movidos a propulsão líquida. O primeiro lançador desta família, o ALFA seria uma evolução direta do Veículo Lançador de Satélites (VLS-1). Originalmente,

o VLS-1 é um foguete composto por quatro estágios, todos movidos a combustível sólido. O projeto do veículo ALFA foi concebido de forma a substituir os últimos dois estágios sólidos do VLS-1 por um estágio de combustível líquido.

Este capítulo está organizado de forma a, após uma breve introdução, apresentar o contexto e a justificativa da pesquisa da tese. São feitas considerações sobre problemas em sistemas críticos, principalmente os da área espacial, que motivaram a criação do processo proposto. Em seguida são apresentados os objetivos deste trabalho de tese e a abordagem adotada. Logo após, são listadas as contribuições relevantes da tese no âmbito de um trabalho de doutorado. Finalmente, é feito um breve histórico do desenvolvimento do trabalho da tese em termos de produção científica decorrentes da pesquisa.

A tese foi estruturada de modo a identificar no Capítulo 1 os fatores motivacionais e de contexto onde a pesquisa está inserida. No Capítulo 2 são descritos alguns conceitos relevantes sobre a técnica i^* , sobre o uso de palavras-guia do HAZOP e sobre o contexto das aplicações computacionais críticas da área espacial, usado como ilustração da aplicação do processo. No Capítulo 3 é apresentado o processo ELICERE, que integra as técnicas i^* , HAZOP e FMEA para a definição das metas de dependabilidade. O modelo do processo de elicitação é definido através de uma sequência de atividades e produtos ilustrados neste capítulo. O capítulo 4 ilustra a maneira como o processo deve ser aplicado, através de exemplos extraídos de um estudo de caso. Finalmente, no Capítulo 5, é feito um resumo das principais considerações da pesquisa, as contribuições da tese à área de pesquisa de dependabilidade e os seus possíveis desdobramentos em futuros trabalhos.

1.1 CONTEXTO E JUSTIFICATIVA DO TRABALHO

A crescente dependência da sociedade atual dos chamados sistemas de informação, seja no entendimento, criação e manutenção de sistemas físicos, como biológicos e sociais, gera uma necessidade de aprimoramento do homem nas habilidades de entender e projetar sistemas de engenharia, principalmente os

classificados como sistemas complexos (MAGEE; DE WECK, 2004). Bar-Yam (2003) define sistema complexo como aquele que, além de considerar um número razoável de variáveis distintas envolvidas, integra vários sistemas, possui restrições de alto desempenho, demanda funções múltiplas, alta taxa de resposta e possui protocolos específicos dentro de seu contexto de atuação. Johnson (2006) enfatiza que sistemas complexos podem ter múltiplas interações que ocorrem entre diferentes componentes, e acrescenta ainda que esse tipo de sistema seja identificado por apresentar um fluxo de interações temporais complexo, em resposta às rápidas mudanças do sistema e de seu ambiente. Leveson (2002) explica no seu livro disponível na internet, que estão sendo projetados sistemas com interações potenciais entre os componentes que podem não estar sendo completamente planejadas, compreendidas, antecipadas ou resguardadas. A operação de alguns sistemas é tão complexa que desafia a compreensão de qualquer pessoa e até mesmo de peritos, que, muitas vezes não têm informações completas sobre seu comportamento potencial. O software facilitou, de forma mais integrada, a implementação de sistemas que contêm um grande número de componentes interagindo dinamicamente e onde o forte acoplamento permite disfunções entre uma e outra parte do sistema. O problema, segundo a autora, é que estão sendo desenvolvidos sistemas que vão além da habilidade intelectual para administrar: o aumento da complexidade interativa e do acoplamento torna difícil para os projetistas a consideração de todos os estados potenciais do sistema ou para operadores lidarem com todas as situações normais, anormais e as perturbações de forma segura e eficiente.

No caso específico da área espacial, a busca por uma automação maior dentro dos sistemas tem aumentado consideravelmente a complexidade da missão e dos objetivos requeridos para alcançá-la. Ingham et al. (2004) afirma que o uso de robôs, por exemplo em naves espaciais, apesar da complexidade advinda de seu uso, atualmente se tornou uma necessidade econômica para a continuação do progresso no espaço. Os autores relatam que, nos projetos de sistemas espaciais, existe uma lacuna entre os requisitos especificados pelos engenheiros de sistema e sua tradução em termos de requisitos especificados pelos engenheiros de software. Lutz e Mikulski (2004) acrescentam ainda que em muitos sistemas embarcados ou que possuem algum tipo de interatividade com o homem, especialmente no domínio de

sistemas críticos, como de uma nave espacial, informações a respeito dos requisitos são descobertos durante todo desenvolvimento do sistema, podendo este processo se estender muito além do previsto.

Um outro estudo, realizado por Hecht e Buettner (2005), no período de 1998 a 2000, informa que aproximadamente metade das anomalias observadas em veículos espaciais estava relacionada a software. Os autores afirmam que, em um sistema em desenvolvimento, é na geração de requisitos de software onde ocorre a maior fonte de erros. No estudo realizado por estes especialistas, das falhas oriundas de requisitos metade tem como causas a pobreza de descrição, a ambigüidade, incorreção ou falta de clareza. A outra metade é devida à omissão completa de requisitos.

É possível afirmar também que práticas inadequadas de especificações de requisitos ou conhecimento insuficiente do sistema, levando a um entendimento pobre sobre seu contexto e seus requisitos operacionais e de qualidade, estão entre as maiores fontes de insucesso de projetos críticos. Na conclusão da investigação do acidente do foguete Ariane 501 (LYONS, 1996), da Agência Espacial Européia (ESA) (no inglês *European Space Agency*) foram identificadas práticas pobres de especificação de requisitos. O relatório da investigação dos acidentes das naves Mars Climate Orbiter – MCO (NASA-MCO, 1999) e Mars Polar Lander – MPL (NASA-JPL, 2000), recomendaram treinamento mais adequado das equipes de desenvolvimento devido a problemas na especificação dos requisitos do sistema. Hjortnaes (2003) observa, na sua análise de mais de dezoito projetos da ESA, uma falta de maturidade e estabilidade da *baseline* de requisitos de software quando o processo de desenvolvimento se inicia. Ele cita ainda que, em muitos dos relatórios analisados, o desenvolvimento dos requisitos não foi conduzido de forma adequada ou não existia um entendimento correto dos mesmos.

Já com relação às questões de dependabilidade de sistemas, negar que o software seja um item de segurança-crítica, que só provê informações e não lança energia diretamente no sistema, está se tornando cada vez menos aceitável. A dependência cada vez maior do software em sistemas de informações está introduzindo novos caminhos para o risco provocado pela perda de informações ou pela informação

incorreta, podendo levar a uma perda física, científica, ou financeira inaceitável. Os sistemas de software têm atingido um nível de complexidade e acoplamento entre os seus componentes que ultrapassa a habilidade humana de controle sobre suas interações. O comportamento de um sistema computacional pode não ser totalmente planejado, compreendido, antecipado, ou totalmente confiável. Mecanismos de confiabilidade, como redundância, nesta situação, podem tornar o problema pior, adicionando complexidade que pode resultar em acidente até então não previsto. Durante a integração e testes das naves Voyager e Galileo, foram encontrados 387 erros (LUTZ, 1993). Destes, os potencialmente relacionados à segurança surgiram mais comumente de discrepâncias entre as especificações de requisitos documentados e os requisitos necessários para o correto funcionamento do sistema e do mau entendimento sobre a interface do software com o resto do sistema. No relatório do acidente do MPL, recomenda-se o uso de FMECA (da sigla em inglês, *Failure Modes and Effects and Criticality Analysis*) e FTA (da sigla em inglês, *Fault Tree Analysis*), com as apropriadas redundâncias para eliminarem falhas, apesar da discussão atual sobre a efetividade do uso destas técnicas convencionais para a identificação de falhas de software.

No caso específico do acidente do terceiro protótipo do Veículo Lançador de Satélites (VLS-1 V03), da Agência Espacial Brasileira (AEB), tanto o relatório oficial da investigação do acidente (SERVIÇO PÚBLICO FEDERAL, 2004), quanto o estudo independente proposto por Almeida e Johnson (2005) reforça a pouca atenção ao desenvolvimento de uma cultura de segurança apropriada ao projeto. Faltaram revisões, auditorias externas e validação por especialistas de avaliação de risco durante o desenvolvimento do projeto. Almeida e Johnson argumentam ainda que em algumas áreas do programa espacial brasileiro não há o reconhecimento da importância do gerenciamento de segurança de sistemas, como é recomendado para as indústrias da área de espaço da América do Norte e Europa.

Investir esforços na melhoria das atividades ligadas ao processo de eliciação, no sentido de descoberta e elucidação de requisitos, sejam eles requisitos de missão ou requisitos de qualidade, devem ser estimulados em organizações da área espacial, de forma a garantir o cumprimento de cada missão com segurança e com sucesso.

1.2 OBJETIVO

Esta tese de doutorado tem como objetivo criar um processo de engenharia para auxiliar na definição de metas de qualidade para sistemas computacionais críticos, principalmente metas de segurança, confiabilidade entre outros fatores ligados à dependabilidade, que auxiliarão na definição de requisitos não funcionais do sistema a ser construído. Requisitos não funcionais são requisitos associados a atributos de qualidade do sistema, que auxiliam a garantir que as metas do sistema relacionadas aos requisitos funcionais possam ser cumpridas.

Uma estratégia de elicitação de metas, que auxiliem na criação de requisitos não funcionais específicos de dependabilidade, é uma das formas de melhorar a qualidade dos sistemas computacionais e garantir um nível maior de confiança em sistemas críticos. Diferentes equipes técnicas, projetos que demandam grande quantidade de recursos, complexidade sistêmica alta, fatores gerenciais e culturais muito heterogêneos, são algumas das questões fundamentais que influenciam as atividades de um projeto, especialmente na identificação das metas de dependabilidade de um sistema.

Para alcançar esse objetivo, a tese propõe:

- A adoção de boas práticas de engenharia de sistemas e de software através da definição de um processo para a elicitação de metas de dependabilidade a ser executado na fase inicial do projeto de um sistema crítico;
- Utilização de técnica de modelagem orientada a metas e técnicas de análise de perigos aplicadas ao sistema a ser construído;
- Melhoria do entendimento do sistema com relação à segurança, disponibilidade, confiabilidade, entre outros fatores, tanto por parte de engenheiros de sistema como de software, através de modelos simples e conceitos intuitivos;
- Apresentação de um conjunto possível de fatores de dependabilidade que possam ser associados às metas a serem definidas através do processo ELICERE.

1.3 ABORDAGEM ADOTADA

A abordagem empregada consiste na criação de um processo de elicitación de metas de um sistema crítico, utilizando a técnica de modelagem de sistema organizacionais baseada em metas, chamada i* (pronuncia-se istar ou iestrela). Aliada ao i* são aplicadas palavras-guia baseadas principalmente na técnica de estudo de perigos HAZOP, que identifica e analisa riscos operacionais de um sistema, e FMEA, que analisa os modos de falha do sistema e seus efeitos. Essas técnicas são aplicadas a exemplos do domínio dos sistemas computacionais críticos de sistemas espaciais. Esta abordagem interdisciplinar permite promover a identificação, ainda na fase de concepção de projeto, de metas ou objetivos que atendam aos requisitos de qualidade, mais especificamente as questões relativas à dependabilidade de projetos computacionais críticos.

O estudo de caso apresentado nesta tese foi direcionado para o desenvolvimento de sistemas computacionais da área espacial do Brasil, principalmente pela necessidade e carência no país de soluções adequadas para a definição de metas de dependabilidade. O desafio de elucidar metas, que se transformarão em requisitos, envolve diversos tipos de sistema e múltiplos pontos de vista no desenvolvimento de um projeto de missão crítica. Sistemas como os embarcados em veículos, os de apoio em solo ao lançamento, de apoio à decisão, de testes e simulação, entre outros, compõem o universo de aplicações que podem se beneficiar com o processo proposto.

Tipo e objetivos de uma missão, restrições de tecnologia, carga útil de lançamento, tempo e segurança de missão fazem parte das variáveis que compõem a definição de sistemas de veículos espaciais que direta, ou indiretamente afetam as metas de dependabilidade a serem definidas.

Todos estes tópicos devem estar vinculados a uma política estratégica institucional e nacional para a independência no setor espacial no que tange à observação da terra, aos serviços de telecomunicação e de gerenciamento de tráfego aéreo, entre outros.

O estudo de caso foi baseado em um caso piloto (YIN, 1984) do projeto de um sistema computacional de um veículo espacial hipotético, abordado de forma qualitativa e descritiva (ZAINAL, 2007). O veículo está baseado no foguete ALFA do programa Cruzeiro do Sul (MORAES et al., 2006) e teve como principal objetivo conduzir a pesquisa no sentido de avaliar quais atividades, informações e produtos deveriam compor o processo ELICERE, que demonstrasse a sua viabilidade de execução.

Conforme o guia prático de condução de estudo de caso de Atkins e Sampson (2002), foi adotado a seguinte abordagem:

- identificação da viabilidade do processo no domínio da aplicação, seu modo de operação e comunicação com os envolvidos;
- observação da clareza das atividades propostas;
- descrição de exemplos didáticos que abranjam uma gama suficiente de metas, tarefas e recursos do sistema a ser construído;
- produção de uma versão do processo de forma a facilitar uma futura melhoria.

A idéia é mostrar, através de exemplos simples, a viabilidade do uso do processo, e sugerir metas de dependabilidade que possam ser aplicadas a sistemas computacionais embarcados de novos projetos espaciais do Brasil, como também do projeto do sistema de controle e seqüenciamento de eventos do Veículo Lançador de Satélites. Os exemplos apresentados formam um conjunto de metas estratégicas de missão relacionadas a controle de vôo, comunicação em terra, segurança de vôo e sistema de telemetria, entre outros.

Deve-se ressaltar que o processo proposto deve ser aplicado nas fases iniciais de projeto de um sistema computacional críticos, na fase de elicitação dos requisitos onde, de maneira geral, ainda estão sendo definidas as metas que o sistema proposto deve atender. Geralmente, neste momento, as metas ou objetivos estão muito focados nas funcionalidades do sistema, isto é, nos requisitos de missão, ou no que o sistema deve fazer, e não o que o sistema não pode fazer ou como o sistema deve se comportar para se ter confiança no seu funcionamento (DE LEMOS, 1997).

1.4 ESTADO DA ARTE

No estado da arte atual da engenharia de requisitos, principalmente no que diz respeito as técnicas orientadas a metas conhecidas como GORE (da sigla em inglês *Goal-Oriented Requirements Engineering*), destacam-se a metodologia KAOS (no inglês, *Knowledge Acquisition in autOmated Specification*), a estrutura NFR (do inglês, *Non Functional Requirements Framework*) e variações da abordagem i*, como a do projeto RESCUE (da sigla em inglês, *Requirements Engineering with Scenarios in User-Centred Environments*).

A metodologia KAOS (DARDENNE; VAN LAMSWEERDE; FICKAS, 1993) possui um conjunto rico de técnicas de análise, combinando diferentes níveis de abstrações, como a semi-formal, para modelar e estruturar metas; qualitativa, para seleção entre as possíveis alternativas do sistema a ser construído; e formal, quando é necessário criar um modelo mais apurado do sistema (LAPOUCHNIAN, 2005). Metas, em KAOS, podem ser relacionadas a serviços (metas funcionais), bem como a qualidade do serviço (metas não-funcionais). A metodologia exige do engenheiro de requisitos um conhecimento aprimorado sobre notação e prova formal de modelos, pois utiliza máquinas de estados, modelos dinâmicos, cenários e diagrama de classes, entre outras formas de representação do sistema (VAN LAMSWEERDE, 2003).

O NFR (MYLOPOULOS; CHUNG; NIXON, 1992) (CHUNG et al., 1999) provê uma representação formal para a análise de requisitos não funcionais, em termos de metas inter-relacionadas que representem os requisitos e também as decisões de projeto, isto é, as metas de negócio. A idéia é refinar o modelo do sistema através de operadores lógicos (AND/OR) e expor as influências positivas e negativas (softgoal “+” e “-“) de seus modelos. Nesta abordagem é necessário que o engenheiro de requisitos possua conhecimento de notação formal de modelagem de sistemas, utilizando a Linguagem de Requisitos Orientada a Metas (GRL) (do inglês, *Goal-oriented Requirement Language*) e de Gráficos de Interdependência de Metas-soft (SIG) (do inglês, *Softgoal Interdependency Graph*) para representar e armazenar os passos e raciocínios do projeto. Além do gráfico, a estrutura oferece

também um repositório de dados, que funciona como uma base de conhecimento, chamada de catálogos (de tipos, métodos e interdependências) que são usados para expressar, previamente, o conhecimento sobre os requisitos não funcionais. Seu uso está voltado basicamente para aplicações da área de Tecnologia da Informação (TI), e a semântica da linguagem SIG e GRL exigem um razoável grau de conhecimento de expressões lógicas assertivas.

O framework i* está sendo atualmente utilizada tanto na disciplina de engenharia de requisitos como na re-engenharia de negócios, na análise de impacto organizacional, e na modelagem de processos de software (LAPOUCHNIAN, 2005). A estrutura tem o foco nos relacionamentos estratégicos entre agentes intencionais de uma organização e provê a caracterização necessária para descrever processos e direcionar mudanças (YU, 1995).

Algumas das pesquisas mais significativas nesta área, que envolveram a técnica de modelagem i*, foram utilizadas na criação da abordagem chamada RESCUE. Esta abordagem utiliza conceitos de engenharia concorrente para identificar requisitos operacionais do gerenciamento do tráfego aéreo britânico (MAIDEN; JONES, 2004) (MAIDEN; KAMDAR; BUSH, 2006). Foi ainda utilizada no sistema CORA-2 (do inglês, *Conflict Resolution Assistance*), para resolver potenciais conflitos entre aeronaves pelos controladores de voo, e no projeto DMAN (do inglês, *DEparture MANager*), para auxiliar os controladores de voo no escalonamento e gerenciamento de partidas de aeronaves da maioria dos aeroportos da Europa. Em muitos destes trabalhos a ferramenta REDEPEND (do inglês, *Requirements Engineering DEPENDencies*) foi utilizada para criar casos de uso e cenários, através dos recursos de modelagem do i*. Devido à abordagem RESCUE ser mais utilizada para a especificação de requisitos operacionais, uma análise de potenciais perigos ou da fragilidade do sistema quanto à segurança está fora do seu escopo (MAIDEN; JONES, 2004).

A engenharia de software orientada a agentes (AOSE) (da sigla em inglês, *Agent-Oriented Software Engineering*) é uma das mais promissoras técnicas da engenharia de requisitos em estudo atualmente. Ela é baseada na premissa de que as atuais aplicações de software estão fortemente baseadas em arquiteturas dinâmicas e

abertas, onde seus componentes podem realizar tarefas em um número variado de ambientes e domínios, e esses componentes devem possuir atributos que permitam o aprendizado, a comunicação e seu auto gerenciamento, características essenciais para o conceito de agentes (MYLOPOULOS, 2008). O projeto Tropos (CASTRO; MYLOPOULOS; KOLP, 2002) apresenta uma metodologia de desenvolvimento orientada a agentes, que utiliza o i* para representar as razões dos requisitos e as escolhas das possíveis configurações do sistema. Possui diversas fases de um projeto, que vai desde a definição de requisitos iniciais, requisitos finais, projeto arquitetural, detalhado e implementação. Uma outra abordagem para o i* também foi aplicada para integrá-lo com modelos UML (ALENCAR et. al., 2006) (CASTRO et al., 2001) e mais recentemente estão em estudo uma simplificação do i* orientado a aspectos (ALENCAR et al., 2008), o i* orientado a serviço (ESTRADA et al., 2008), ou a criação e exploração de um novo diagrama para representar as relações de um ator, intitulado diagrama SA (da sigla em inglês, *Strategic Actor*) (LEITE; CAPPELLI, 2008) (LEITE et al., 2007).

1.5 CONTRIBUIÇÕES DA TESE

O caráter inovador deste trabalho está em relacionar uma técnica de extração de metas de sistemas organizacionais, criada nos anos 80 para sistemas de Tecnologia da Informação (TI), à área de engenharia de segurança, adaptando palavras-guia de técnicas mais tradicionais e consolidadas. Palavras-guia são palavras chaves utilizadas por um grupo de estudo para qualificar os desvios de um projeto analisado. A proposta do trabalho é formalizar um processo abrangendo as atividades de obtenção do modelo do sistema, de investigação de seu comportamento e identificação de metas visando a definição dos requisitos não funcionais, conseqüentemente melhorando a qualidade dos projetos no que tange à sua segurança.

É importante frisar que um dos focos do trabalho foi o de melhorar efetivamente a participação da equipe da engenharia de software nos trabalhos executados pela equipe de engenharia de sistemas. A técnica de modelagem de requisitos de

software é um recurso estratégico no auxílio à definição de sistemas computacionais, tornando-se uma importante contribuição da engenharia de software para a engenharia de sistemas. O processo e seus resultados visam o melhor entendimento do sistema a ser construído e a melhoria da comunicação entre estas duas equipes para a identificação das metas de dependabilidade do sistema. É necessário que no início do projeto as diferentes equipes tenham um entendimento comum acerca dos perigos potenciais do sistema a ser construído para a criação de metas de qualidade.

A motivação para a escolha do caso de uso adotado, voltado para o projeto do sistema computacional de um veículo espacial brasileiro, é devida à carência de técnicas capazes de ajudar na identificação de metas que reflitam as reais necessidades dos usuários e clientes deste segmento do sistema. Uma visão integrada entre as engenharias, de software e de sistema, de como abordar a dependabilidade, de forma a garantir o cumprimento e a segurança da missão como um todo, é um desafio que deve ser suplantado. Tanto a disciplina de engenharia de requisitos (no inglês, *Requirements Engineering*) quanto da disciplina da engenharia de segurança (no inglês, *Safety Engineering*) precisam ser revistas, dentro do contexto das novas tecnologias computacionais existentes e da demanda dos futuros projetos.

Não existe um processo bem definido para a elicitação de requisitos, principalmente dos relacionados à segurança de software e de sistema. É preciso adotar uma abordagem sistemática e cultural da segurança, que seja implantada de modo a não depender da figura do herói do projeto, personalidade capaz de resolver diversos tipos de problemas, como ambigüidades e situações de risco, mas não de forma disciplinada e com método.

1.6 HISTÓRICO DA PESQUISA DA TESE

Um das motivações iniciais para a escolha do tema desta tese foi devido a participação do autor como um dos apresentadores do mini-tutorial “Uma

Abordagem Prática de Garantia da Qualidade em Sistemas Críticos” (MOURA; LAHOZ; ABDALA, 2003) no Simpósio Latino Americano de Dependabilidade Computacional (no inglês, *Latin American Symposium on Dependable Computing*). A proposta do tutorial era de estimular a discussão de abordagens aplicáveis ao desenvolvimento de software para sistemas críticos e compartilhar a experiência adquirida com o desenvolvimento do software de bordo do foguete VLS-1.

O passo seguinte foi aprofundar a discussão de quais eram os padrões, modelos e processos adequados para as aplicações críticas. O artigo apresentado no *55th International Astronautical Congress* (IAC) propunha, sob o ponto de vista da qualidade e confiabilidade, uma abordagem para a definição e modelagem dos processos da qualidade para aplicação de software em projetos do setor espacial brasileiro, expressos em uma linguagem de processo adequada para o uso em ambientes de engenharia de software (LAHOZ; MOURA, 2004). Neste mesmo ano foi apresentado no Workshop de Segurança, evento promovido pelo Grupo de Análise de Segurança do Departamento de Sistemas Digitais da POLI/USP, os padrões de qualidade e avaliação de software atualmente adotado pela ESA. Foi ilustrado como exemplo um processo específico para projetos espaciais envolvendo dependabilidade e segurança (LAHOZ; BURGARELI; MOURA, 2004). Em 2005 foi também apresentado no *23rd International System Safety Conference* artigo sobre práticas de modelagem de processo, onde, para o desenho do processo, foi utilizada uma linguagem de modelagem baseada na UML (na sigla em inglês, *Unified Modeling Language*) (LAHOZ; BURGARELI; CAMARGO JÚNIOR, 2005). Processo de software também foi tema de dois tutoriais apresentados no Simpósio Internacional de Melhoria de Processo (SIMPROS), onde foram apresentados trabalhos baseados nos padrões ISO/IEC-12207, 15504 e ECSS (LAHOZ, 2005) (LAHOZ, 2007).

Na primeira conferência promovida pelo *Institution of Engineering and Technology* (IEE/IET) da Inglaterra, sobre Segurança de Sistemas, Lahoz et al. (2006) introduziu os principais obstáculos que a atividade de elicitação de requisitos deveria ultrapassar, sob a ótica de pessoas, processos e tecnologia. Em 2006 ainda, no *IX Workshop on Requirements Engineering* no Rio de Janeiro, Lahoz e Camargo Júnior (2006) apresentaram os resultados preliminares desta pesquisa, enfocando os

principais problemas que a atividade de elicitación de requisitos enfrenta atualmente em projetos espaciais no Brasil. Neste evento foi possível conhecer o autor da técnica de modelagem i*, que motivou o uso deste tipo de recurso no processo ELICERE.

Em 2007, quando o processo ELICERE já estava consolidado como tema central da tese, seu primeiro desenho foi apresentado na *25th International System Safety Conference*, mostrando o uso do i* e das palavras-guia HAZOP na descoberta de requisitos de dependabilidade (LAHOZ; CAMARGO JÚNIOR, 2007). Neste evento foi debatido com diversos especialistas da área de desenvolvimento de processos de segurança, tópicos de interesse do tema metas e requisitos de dependabilidade.

Em 2008 este trabalho de tese, após ter sua qualificação aceita pela Pós-Graduação da USP, foi selecionado para ser apresentado no *Doctoral Symposium* do *16th IEEE International Requirements Engineering Conference* em apresentação oral (LAHOZ; 2008). Nesta oportunidade diversos doutores e professores como Nazim H. Madhavji (*University of Western Ontario, Canada*), Brian Berenbach (*Siemens Corporation, EUA*), Martin Glinz (*University of Zurich, Suíça*), Julio Leite (PUC, Brasil), Dan Berry (*University of Waterloo, Canada*), Lucia Rapanotti (*Open University, Reino Unido*), fizeram comentários, sugestões e críticas enriquecedoras para a conclusão das pesquisas e da tese.

Finalmente o processo ELICERE foi agraciado com o prêmio *Best Student Paper* (LAHOZ; CAMARGO JÚNIOR, 2009) na conferência promovida pela INCOSE (da sigla em inglês, *International Council on Systems Engineering*). Este organismo, e este evento promovido anualmente, é um dos mais importantes da área de engenharia de sistemas no mundo.

Além destas publicações e eventos que serviram para a condução e motivação do desenvolvimento da tese, está o fato do processo ELICERE poder ser aplicado amplamente em sistemas computacionais críticos, em especial na área espacial, onde o autor tem atuação profissional no desenvolvimento do software embarcado do VLS-1, e na supervisão do serviço de atualização do sistema de apoio em solo para o lançamento deste foguete.

2 ENGENHARIA DE REQUISITOS E ENGENHARIA DE SEGURANÇA

Este capítulo apresenta os conceitos mais importantes relacionados às técnicas utilizadas para a criação do processo ELICERE, abrangendo tópicos das disciplinas de engenharia de requisitos e de engenharia de segurança aplicadas ao domínio de sistemas críticos. Considerações sobre engenharia de sistemas e problemas relacionados a requisitos e segurança em sistemas espaciais também são discutidos neste capítulo. O propósito é, além de introduzir a teoria e os temas mais relevantes aplicados durante o desenvolvimento da tese, possibilitar um entendimento comum sobre o vocabulário adotado, enfatizando que o processo proposto deve ser aplicado principalmente na fase de concepção de um sistema crítico.

Conforme mostra a Figura 2.1, o processo ELICERE se insere na intersecção de algumas das atividades da engenharia de requisitos com a de segurança, quando o levantamento das informações necessárias para a formulação dos requisitos do sistema ainda está sendo feito.

Figura 2.1 – Contexto conceitual da abordagem do processo ELICERE.

O processo é aplicável tanto na engenharia de sistemas como na engenharia de software e na engenharia de hardware, especificamente para as atividades de descoberta de requisitos voltados às questões de segurança. O processo proposto visa auxiliar a definição de metas de dependabilidade para sistemas computacionais críticos.

Alguns tópicos da engenharia, sob o ponto de vista de requisitos e segurança, são apresentados neste capítulo, de modo a permitir o entendimento da estratégia para a elicitação de metas de dependabilidade. Conceitos sobre a área espacial, tema do estudo de caso apresentado como exemplo, também são discutidos.

2.1 ENGENHARIA DE REQUISITOS

O termo Engenharia de Requisitos (ER), segundo Kotonya e Sommerville (2000) foi criado para cobrir todas as atividades envolvidas na descoberta, documentação e manutenção de um conjunto de requisitos para um sistema baseado em computador. Os autores afirmam ainda que o termo engenharia implica na utilização sistemática de técnicas que garantam que os requisitos do sistema sejam completos, consistentes, relevantes, entre outras características. Para Laplante (2004), à disciplina de engenharia de requisitos de software compete a determinação dos objetivos, funções e restrições de um sistema de software e a representação dos seus aspectos de forma acessível para modelagem e análise. Esta disciplina tem também como objetivo identificar requisitos que sejam corretos, completos e inteligíveis para clientes e desenvolvedores.

Pode-se afirmar também que o termo Engenharia de Requisitos abrange tanto os assuntos relativos à análise e especificação de sistemas de informação (negócios) como o de análise e especificação de sistemas de engenharia com base computacional.

As atividades do ciclo de vida do processo de ER podem ser representadas conforme mostra a Figura 2.2, baseando-se em um modelo espiral proposto por

Kotonya e Sommerville (2000). O processo de ER tem como produto fundamental o documento de especificação de requisitos. Este documento deve ser construído em ciclos, passando pelos processos de elicitação, análise, documentação e validação, até que um evento externo ao processo determine o final da espiral.

Figura 2.2 – Processo de ER - adaptado de Kotonya e Sommerville (2000).

O processo de ER e seu nível de detalhamento podem variar de uma organização para outra. Porém, uma organização de desenvolvimento de projetos críticos pode adaptar técnicas e práticas aplicadas a projetos da área de tecnologia da informação e vice-versa. De modo geral o processo de ER, em qualquer tipo de sistema, abrange as atividades de elicitação, análise, documentação, validação e gerenciamento de requisitos. O principal objetivo da elicitação é extrair informações sobre o problema a ser resolvido, o serviço que o sistema proposto deve disponibilizar, os requisitos de desempenho, as restrições de software e de hardware, entre outros assuntos que ajudem a conduzir a definição dos requisitos do sistema. (KOTONYA; SOMMERVILLE, 2000)

Um evento externo pode provocar o encerramento da atividade de elicitação, por exemplo, o aceite do conjunto de metas identificadas pelo engenheiro de requisitos antes do início da fase de análise. O processo pode ainda ser encerrado ao ser atingido o prazo final desta fase definido no cronograma, ou devido a restrições de recursos ou outra condição que os participantes responsáveis e interessados determinarem para sua conclusão.

2.1.1 Processo de engenharia de requisitos e os envolvidos

A forma com que o processo de engenharia de requisitos é conduzido em um projeto pode afetar diretamente o universo dos envolvidos (do termo em inglês, *stakeholders*), principalmente nas atividades de elicitação, onde a interação com participantes externos ao desenvolvimento é maior. O processo de elicitação envolve, entre outros participantes, o cliente, o engenheiro de sistema, o engenheiro de software, o gerente de projeto, o engenheiro de requisitos, e outros membros do projeto, dentro de uma visão de tecnologia, processo e cultura da organização (LAHOZ et al., 2006). Na visão de tecnologia são consideradas quais ferramentas e técnicas de ER serão utilizadas pelos engenheiros de requisitos para um determinado projeto. Na visão de processo, consideram-se como os participantes irão desempenhar suas atividades, papéis e produtos de forma mais adequada ao processo de ER. E na visão cultural, considera-se como os membros da equipe, de diferentes origens, linhas de pensamento e formas de abordagem, conduzirão o processo de ER, para determinar o domínio do problema e realizar a análise do comportamento do sistema sob a ótica da segurança, por exemplo.

No modelo de avaliação e melhoria de processo de software proposto pela ESA (ECSS-Q-80-02, 2005), a atividade de elicitação de requisitos aparece como a primeira de uma série de atividades primárias do ciclo de vida de engenharia. A Figura 2.3 mostra uma visão de quem são os envolvidos no processo de ER, e suas principais solicitações com referência à natureza dos requisitos.

Figura 2.3 – Os envolvidos no processo de ER.

No documento de engenharia de software da ESA (ECSS-E-ST-40C, 2009) é mencionada a necessidade do fornecedor de software envolver todos os participantes na atividade de engenharia de requisitos, inclusive atendendo os resultados de uma análise de segurança e dependabilidade. Os envolvidos neste processo, bem como os papéis que cada participante desempenha no processo de Engenharia de Requisitos é descrito a seguir.

Cliente: define o sistema no seu nível mais abrangente. Geralmente, em projetos críticos, é representado pelas agências governamentais ou pelos dirigentes de alto nível de um programa desta natureza. Observa o sistema dentro de um contexto estratégico ou político diferente do observado pelos outros envolvidos e pode não ter muita influência nos rumos tomados por um projeto no seu nível mais prático de desenvolvimento e acompanhamento.

Engenheiro de sistemas: capaz de entender as necessidades técnicas dos participantes no desenvolvimento do sistema. Promove um entendimento mais geral

da solução do problema e é adverso aos riscos de projeto. Usa linguagem natural para prospectar soluções e procura balancear os métodos de engenharia e gerenciamento para alcançar seus objetivos. Entende a disciplina de requisitos como necessária, mas muitas vezes, carece de uma visão metodológica desta.

Engenheiro de hardware: desempenha a função de identificar todas os requisitos e funções que se deseja atribuir ao hardware de um sistema. A engenharia de hardware envolve eletrônica que pode assumir tecnologias diferentes: eletrônica analógica, eletrônica digital, microprocessadores, microcontroladores, comunicação sem fio, entre outras tecnologias isoladas ou até mesmo combinadas, tendo cada uma, características próprias. O crescente uso de linguagens de programação para a descrição de hardware, como a HDL (da sigla em inglês, *Hardware Description Language*) tornou as funções de um engenheiro de hardware bastante semelhantes às de um engenheiro de software.

Engenheiro de software: papel importante na manipulação e validação dos requisitos. Utiliza técnicas de modelagem e de entrevista para conhecer melhor os requisitos de software. Devido à natureza das aplicações críticas, fortemente dependente de sistemas eletronicamente programáveis, o engenheiro de software assume atualmente papel fundamental na definição do sistema. Pode sugerir e negociar novos requisitos com o engenheiro de sistema, elucidando novas metas ou mesmo adaptando soluções para possíveis restrições de projeto. Pode também, devido a uma interpretação errônea ou parcial de alguma funcionalidade, desenvolver o sistema de forma a não conseguir atender às expectativas do projeto.

Engenheiro de segurança: em muitas organizações tem atribuições relativas à segurança do trabalho e do ambiente. Para sistemas computacionais, é um recurso humano pouco comum ou inexistente no mercado. Este tipo de profissional geralmente tem alguma experiência com implantação de políticas de segurança, com adoção e documentação de padrões, monitoração dos procedimentos de segurança, análise de perigos e investigação de acidentes. Pode, em muitos casos, ter um bom conhecimento do hardware e do software do projeto de um sistema, mas não ter poder suficiente para influenciar nas decisões organizacionais.

Gerente de projeto: possui visão mais pragmática de um projeto, voltando muitas vezes sua preocupação somente para o gerenciamento dos recursos e do tempo. Sofre vários tipos de pressões durante suas atividades, desde a do alto escalão da organização até a do pessoal de desenvolvimento e produção, que podem levá-lo a adotar medidas impopulares e de impacto. Utiliza técnicas de administração e gerenciamento de projeto como WBS (da sigla em inglês, *Work Breakdown Structure*).

Usuário do sistema: representado pelos clientes diretos do sistema, como operadores (técnicos e engenheiros) e equipe de hardware e de software. Também são os interessados nos requisitos do sistema mais voltados para operação e manipulação das informações e controle. Podem ter opiniões e pontos de vista muito diferentes, dependendo da função que ocupam no projeto, mas em geral são os participantes que criam e validam os requisitos de um sistema.

Engenheiro de requisitos: quem modela o problema através de um conjunto de formalismos e geralmente é um dos primeiros envolvidos a ter uma visão abrangente do contexto do sistema e do software como um todo, durante as fases iniciais de projeto. Na prática, não existe uma definição clara de seu papel em muitas organizações, levando suas atribuições a serem assumidas por algum outro tipo de engenheiro. Exerce papel estratégico na definição do projeto de um sistema por tentar conciliar os interesses dos clientes com as necessidades dos usuários. Aplica métodos e técnicas de modelagem para especificar o sistema e produzir resultados que afetarão toda a vida do projeto e sua operação.

2.1.2 Engenharia de requisitos orientada a metas

A disciplina engenharia de requisitos foi estabelecida como um campo distinto de investigação e prática da engenharia na metade dos anos 70, sendo sua utilização inicialmente concebida para sistemas de software (ROSS; SCHOMAN, 1977). Padrões como o IEEE-830 (1984) surgiram como uma forma de promover um consenso sobre o assunto. O artigo de Ross e Schoman foi um marco importante

para a consolidação da engenharia de requisitos, ao conceituar não somente o escopo da disciplina e suas perspectivas, mas também seus elementos mais promissores: dados, operações, agentes e recursos.

Segundo Kavakli e Loucopoulos (2003), no início dos anos 90, iniciou-se uma discussão sobre a ampliação do escopo desta disciplina, incorporando os aspectos dos sistemas e das organizações como um todo, como pode ser constatado por Greenspan; Mylopoulos e Borgida (1984), Loucopoulos e Karakostas (1995), Pohl (1996), Yu (1997) e Zave (1997). Nos últimos anos, o foco da Engenharia de Requisitos voltou-se para o contexto organizacional, atraindo ainda mais o interesse da comunidade da tecnologia da informação. A pesquisa nesta área é baseada na premissa de que projetar sistemas de software significa também observar o perfil de seus participantes, os objetivos do negócio, o comportamento da planta e do sistema como um todo, entre outras questões estratégicas para a organização (KAVAKLI; LOUCOPOULOS, 2003).

É importante salientar que nas diferentes definições da disciplina RE, quatro atividades são comumente identificadas em quase todas as abordagens: elicitação, negociação, especificação e validação de requisitos. A elicitação de requisitos, foco desta tese, trata do entendimento da situação técnica e organizacional do sistema a ser construído ou modificado para sua melhoria, através da descrição de suas necessidades e de suas restrições, permitindo que questões relevantes sobre o problema (sistema) sejam de conhecimento dos vários participantes (KAVAKLI; LOUCOPOULOS, 2003).

A descrição do comportamento organizacional através de metas (no inglês, *goals*), foi proposta por diversos autores e apresentada por diversas técnicas que de alguma forma, incorporaram a Engenharia de Requisitos Orientada a Metas, conhecidas como GORE. Algumas dessas técnicas foram: Metas Baseadas em Fluxos de trabalho (no inglês, *Goal-Based Workflow*) (ELLIS; WAINER, 1994), i* (YU, 1995), Desenvolvimento de Conhecimento Empresarial – EKD (no inglês, *Enterprise Knowledge Development*) (KAVAKLI; LOUCOPOULOS, 1999), técnica de Metas, Operadores, Métodos e Seleção de Regras – GOMS (no inglês, *Goals, Operators, Methods and Selection rules*) (CARD; MORAN; NEWEL, 1983), técnica

desenvolvida pelo grupo de pesquisa da Universidade de Estocolmo ISAC (no inglês, *Information Systems for Administrative Control*, que mais tarde foi alterado para *Information Systems work and Analysis of Changes*) (LUNDEBERG, 1982) e F³ (do acronismo inglês, *From Fuzzy to Formal*) (BUBENKO, 1994).

Tanto a abordagem de elicitação e modelagem de requisitos i*, como a abordagem de especificação de requisitos do método KAOS (no inglês, *Knowledge Acquisition in autOmated Specification*) (DARDENNE; VAN LAMSWEERDE; FICKAS, 1993), tomaram um forte impulso nos anos 90, quando diversos centros de pesquisas investiram no uso destas técnicas. Especificamente, a i*, criada a partir da tese de doutorado do professor Eric Yu, da Universidade de Toronto (YU, 1995) em 1995, mais tarde deu origem ao projeto Tropos (CASTRO; MYLOPOULOS, KOLP, 2002) e já foi aplicada em diversos centros de pesquisa.

Atualmente, várias iniciativas de utilização do i* estão sendo tomadas, como a do laboratório de saúde da Universidade de Namur na Bélgica com o projeto ARTHUR (PETIT; ROUSSEAU, 2001), ou mesmo na agricultura, através do Centro de Pesquisa Científica e Tecnológica do Instituto de Cultura de Trento, Itália – ITC-IRST (no italiano, *Instituto Trentino di Cultura – Centro per la Ricerca Scientifica e Tecnológica*) (PERINI; SUSI, 2001). Esta técnica está sendo adotada também na área de tráfego aéreo, especificamente no Sistema de Gerenciamento de Partidas do Controle Aéreo Britânico – NATS-DMAN (no inglês, *National Air Traffic Services – Departure Manager system*) (MAIDEN; JONES; FLYNN, 2003). O i* está sendo utilizado através do projeto RESCUE (no inglês, *Requirements with Scenarios for User-Centred Engineering*), pelos pesquisadores do Centro de Interação Homem-Computador – HCI (no inglês, *Human Computer Interaction*) da City University, em Londres (MAIDEN; JONES, 2004).

Dentro deste projeto, um estudo, ainda preliminar, foi realizado por Bush (2005), aplicado a uma modelagem do sistema NATS-DMAN. O autor apresenta uma abordagem alinhada com as pesquisas deste trabalho de tese, propondo a identificação de requisitos de segurança através da estrutura e semântica da modelagem i* juntamente com a técnica HAZOP. Apesar do caráter preliminar do trabalho de Bush, é possível identificar a contribuição que este tipo de abordagem

pode dar à comunidade científica e a viabilidade que o processo proposto por essa tese pode alcançar.

O processo ELICERE difere do trabalho de Bush principalmente por criar um processo de engenharia formalizado, um conjunto de palavras-guia específicas e por sugerir fatores de qualidade que conduzem a definição de metas de dependabilidade. No domínio dos sistemas computacionais para aplicações críticas, como da área de sistemas espaciais, ainda não foram observadas iniciativas de uso do *i** até a presente data, tornando o tema desta tese relevante para as pesquisas tanto na área de requisitos como na área de segurança.

2.1.3 Metas e requisitos

Engenharia de Requisitos diz respeito à produção de um conjunto de especificações (neste caso, para sistemas computacionais) que satisfaçam às expectativas dos participantes envolvidos e que viabilize, de alguma maneira, sua implementação, decomposição e manutenção. A engenharia de requisitos dirigida a metas preconiza que os requisitos devem inicialmente se focar em questões relacionadas ao “por que”, para só em seguida definir “o que” precisa ser feito. Esta mesma abordagem pode ser observada nos métodos dirigidos a modelo (no inglês *model driven*), que também se preocupam em identificar porque certas funcionalidades são necessárias para depois definir como estas podem ser implantadas. Isto significa que este tipo de abordagem pode permitir a identificação das razões para a implantação de uma funcionalidade do sistema e até mesmo critérios de seleção para estas alternativas. Por outro lado, os métodos convencionais de análise e projeto são focados principalmente nas funcionalidades do sistema a ser construído e suas interações com os usuários.

A pesquisa nesta área evoluiu com a introdução de agentes e de metas (ou também chamados de objetivos) (VAN LAMSWEERDE, 2000) (YUE, 1987) (FEATHER, 1987) (DARDENNE; VAN LAMSWEERDE; FICKAS, 1993). Yue (1987) sugere que a inclusão de metas nos modelos de requisitos possibilita o estabelecimento de um

critério para a avaliação dos requisitos quanto à sua completeza. O autor enfatiza que os requisitos são completos se forem suficientes para atender às metas. Dias Lopes (2002) esclarece que a pesquisa baseada em metas cresceu com os trabalhos de Mylopoulos; Chung e Nixon (1992), Dardene; Lawsweerde e Fickas (1993), Nixon (1993) e Yu (1997). Segundo o autor, a inclusão de metas nos modelos de requisitos possibilitou também o estabelecimento de um critério para a avaliação dos requisitos quanto à sua completeza.

Segundo Dardene; van Lamsweerde e Fickas (1993), metas justificam e explicam a presença de requisitos que não são necessariamente compreendidos pelos clientes. As metas podem ser usadas para atribuir responsabilidades para agentes do sistema, ajustar as restrições de projeto impostas, e finalmente prover informações para detectar e resolver conflitos emergentes dos múltiplos pontos de vista dos envolvidos (ROBINSON, 1989). Agentes são capazes também de realizar ações autônomas e ter um comportamento flexível no seu ambiente, com o intuito de atingir os objetivos para os quais foi projetado (WEISS, 2002).

A partir deste ponto, surgiram linguagens exclusivas para modelar requisitos, baseadas em objetos, agentes e metas, como o método KAOS (na sigla em inglês, *Knowledge Acquisition in Automated Specification*) (DARDENE; VAN LAMSWEERDE; FICKAS, 1993), e uma extensão do seu conceito através da técnica denominada *i** (pronuncia-se *istar* ou *iestrela*) (YU, 1995) (YU, 1997). Yu desenvolveu a técnica *i** para o contexto organizacional e para o caso particular de sistemas de informação.

2.1.4 Técnica de modelagem organizacional *i**

Eric Yu propõe a técnica de modelagem *i**, para representar as razões envolvidas na escolha de uma determinada solução de desenvolvimento de um projeto. O autor busca, através de sua técnica de modelagem, representar o por que de realizar uma determinada ação ou alternativas para o como fazer dentro das técnicas de engenharia de requisitos orientadas a metas.

O i* busca produzir uma modelagem de sistemas mais rica, capaz de reconhecer motivações, intenções, e as características de um sistema ou processo, que facilite os esforços da atividade de ER. Também busca uma descrição que auxilie a melhor compreensão do relacionamento entre os vários atores de um sistema e das razões envolvidas nos processos de decisões, permitindo a representação das várias características de modelagem que devem ser apropriadas à ER, principalmente, na fase inicial da especificação.

Segundo Alencar (1999), a modelagem i* se propõe a ser aplicada na engenharia de requisitos, na reengenharia de processo de negócios e na análise de impactos organizacionais, bem como na modelagem de processo de software. Especificamente na Engenharia de Requisitos, a técnica busca superar seu principal obstáculo: a dificuldade de compreensão aprofundada do domínio da aplicação. Segundo a autora, os modelos de requisitos que existem, normalmente quando descrevem um ambiente organizacional ou um sistema, apenas representam as entidades e as atividades, não observando as implicações que a adoção de uma solução pode acarretar, ao invés de outra. Uma representação mais precisa das motivações e razões, explicitamente capturadas em um modelo de requisitos, pode significar a identificação das necessidades estratégicas que um sistema deve atender e, conseqüentemente, definir soluções de desenvolvimento ou melhoria.

Os requisitos organizacionais de um sistema, na técnica i*, são modelados através de atores, que utilizam recursos para realizar tarefas e alcançar metas. Através dos modelos i* é possível identificar interdependências entre atores, tarefas e recursos para alcançar determinadas metas, típico de aplicações de sistemas sócio-técnicos complexos, como o do gerenciamento de tráfego aéreo (MAIDEN; JONES, 2004).

A técnica i* possui dois tipos de modelos: Dependência Estratégica e Razão Estratégica. Tanto o modelo de Dependência Estratégica – SD (da sigla em inglês *Strategic Dependency*) como o modelo de Razão Estratégica – SR (da sigla em inglês *Strategic Rationale*) possuem quatro tipos principais de nós ou componentes: metas, tarefas, recursos e metas-soft. O termo meta-soft foi adotado para descrever as metas relacionadas a requisitos não funcionais, ou metas de qualidade do sistema (do termo em inglês, *soft-goals*).

O modelo SD fornece uma rede de relações de dependências entre atores e seus componentes, descrevendo as relações externas entre estes atores e a organização ou sistema. A técnica parte do princípio de que o nível mais alto de abstração da dependência estratégica pode ser desenvolvido combinando as habilidades de um ator em satisfazer a solicitação de outro ator (YU; MYLOPOULOS, 1994).

O modelo SR fornece uma descrição das intenções dos atores e seus componentes, e as relações ou razões que os mantêm ligados (ALENCAR, 1999), descrevendo as relações de dependências internas destes atores. Os componentes podem ser conectados através de ligações de relacionamento do tipo *decomposição de tarefa*, ligações do tipo *meio-fim* e *contribuição para meta-soft*.

2.1.4.1 Modelo de dependência estratégica

O modelo de dependência estratégica (SD) é representado por um conjunto de atores e ligações entre eles. Indica uma relação de dependência entre dois atores em termos de uma rede de relacionamentos de dependências. Então, além de ator são definidos os nós ou componentes meta, tarefa, recurso e meta-soft:

Ator: entidade ativa que conduz as ações para obtenção das metas, exercitando suas habilidades. Ator refere-se genericamente a qualquer unidade cuja dependência intencional pode ser descrita. Agentes, papéis e posições são subunidades dos atores de um sistema social complexo, onde cada ator tem um sentido mais especializado.

Meta: afirmação ou objetivo que o ator, ou sistema, deve atender ou alcançar.

Tarefa: diz respeito à atividade que um ator, ou sistema, deve desempenhar para atingir alguma meta.

Recurso: entidade do sistema que disponibiliza algum tipo de informação, produto ou serviço ao ator.

Meta-soft: afirmação ou objetivo, relacionado a requisitos não funcionais, que um ator, ou sistema, deve atender ou alcançar.

Comumente, estes componentes são utilizados para caracterizar o relacionamento entre dois atores. Os atores envolvidos numa relação de dependência (*dependum*) são denominados como dependente (*dependen*), se é o ator que depende que outro ator satisfaça sua dependência, e o ator dependedor (*dependee*) é de quem se depende, ou seja, o ator responsável por satisfazer a relação de dependência.

Desta forma, como mostra a Figura 2.4, o ator que satisfaz a dependência (ator dependedor) – identificado pelo sentido da seta (representado pela letra D) atende alguma meta (meta de dependência) para o ator dependente.

Figura 2.4 – Relação de dependência entre dois atores.

É possível distinguir quatro tipos de dependências, baseados na relação de dependência (*dependum*).

- Dependência de recurso: ator que disponibiliza um recurso necessário para outro ator;
- Dependência de tarefa: ator que executa alguma tarefa de sua responsabilidade para atender um outro ator;
- Dependência de meta: ator que realiza algo para alcançar um objetivo; e
- Dependência de meta-soft: ator que realiza algo para alcançar uma meta não funcional.

É importante salientar que a dependência em qualquer um dos casos exemplificados acima pode ser de um ator com ele mesmo, isto é, uma dependência do próprio ator. No modelo SD são observados três tipos de dependência:

O Aberta (*open*): graficamente expressa por um círculo; ocorre quando os participantes, em caso de falha da relação de dependência, são afetados, mas sem consequências sérias.

Compromissada (*committed*): graficamente não possui sinal associado; ocorre quando os participantes são, significativamente, afetados em caso de falha, podendo conduzir ao questionamento da viabilidade da dependência.

X Crítica (*critical*): graficamente explicitada por meio do sinal de X; ocorre quando os participantes são seriamente afetados em caso de falhas, levando-se a uma preocupação quanto à viabilidade de toda a rede de relacionamentos e dependências.

Um ator pode ainda ser representado através de três diferentes tipos:

Agente: ator que possui manifestações físicas concretas, tal como um ser humano, muito embora esse termo ainda possa ser utilizado em relação a agentes artificiais (hardware, software ou uma combinação dos dois).

Papel: caracterização abstrata do comportamento de um ator dentro de algum contexto ou domínio especializado, ou seja, as atribuições ou funções possíveis de serem exercidas por um agente dentro de uma organização.

Posição: representa uma posição intermediária entre um agente e um papel dentro da organização, onde o agente possa desempenhar várias funções, isto é, um agente ocupando vários papéis.

2.1.4.2 Modelo de razão estratégica

Enquanto o modelo SD trata apenas dos relacionamentos externos entre os atores, o modelo SR é utilizado, dentro do contexto do sistema ou processo, para descrever as intenções e motivações dos atores, para identificar as possíveis alternativas de sua definição e investigar, mais detalhadamente, as razões existentes por detrás das dependências entre os vários atores. Esse modelo permite apresentar uma visão mais internalizada sobre as razões que foram escolhidas para a satisfação das dependências do processo, fornecendo uma estrutura para expressar e detalhar as soluções de projeto. Baseado nos mesmos tipos de componentes do modelo SD, meta, tarefa, recurso e meta-soft, possui três tipos principais de relacionamentos que os interconectam:

Meio-fim (means-ends): indica um relacionamento entre um fim (que pode ser uma meta a ser alcançada, uma tarefa a ser realizada, um recurso a ser produzido, ou uma meta-soft a ser satisfeita) e um meio para se atender a esse fim. Sugere que possam existir outros meios de alcançar o mesmo fim, exprimindo as alternativas existentes.

Decomposição de tarefas (task-decomposition): exprime o que deve ser feito para a tarefa ser realizada e pode ser de quatro tipos: sub-meta, sub-tarefa, recurso-para, ou meta-soft-para. Esse tipo de relacionamento pode interligar-se aos relacionamentos de dependência do modelo SD, quando as razões ultrapassam os limites do ator.

Contribui para meta-soft (contribution link): uma ligação *meio-fim* com uma meta-soft no fim. Pode ser usado nesta relação qualquer um dos componentes do tipo meta, tarefa e recurso para satisfazer uma meta-soft. No modelo de Razão Estratégica é importante observar que as ligações que envolvem a meta-soft requerem um atributo extra para indicar o tipo de contribuição relacionada a essa meta como, por exemplo, contribuição positiva, significando que a meta-soft influenciará positivamente, ou negativa, significando que a meta-soft influenciará negativamente.

Este modelo fornece um rico conjunto de conceitos para a modelagem de processos e as razões por detrás desses processos, bem como a possibilidade de compreensão mais aprofundada, quando comparada com as demais formas de modelagem existentes. Alencar (1999) enfatiza que os modelos *i** não vêem o sistema somente como atividades e seus fluxos, mas capturam a dimensão do por que de um processo. No relacionamento de *decomposição de tarefas*, tem-se uma descrição hierárquica dos componentes intencionais, tal qual um refinamento, onde cada componente é importante para o sucesso do projeto. É possível fazer um questionamento do por que o componente está sendo explorado, isto é, descobrir para que fim o componente está sendo requerido. Através do relacionamento *meio-fim*, o modelo SR fornece uma visão do como o componente está sendo explorado, como o componente está sendo requerido pelo sistema.

2.2 ENGENHARIA DE SEGURANÇA

A engenharia de segurança é uma disciplina que se propõe a garantir que o funcionamento de um sistema tenha um comportamento aceitável, mesmo quando ocorram falhas em seus componentes. A origem deste assunto, tratado como uma preocupação social, remonta ao início da era industrial. Tomou força após a Segunda Guerra Mundial, quando foi criada nos Estados Unidos a Comissão de Energia Atômica – AEC (em inglês, *Atomic Energy Commission*) que, em 1946,

estabeleceu um debate público sobre segurança e energia nuclear (LEVESON, 1995). Nesta mesma época, de acordo com Leveson (2003), a aviação civil tentava convencer o público ainda cético a voar, a indústria química estava lidando com plantas cada vez maiores e produtos químicos cada vez mais nocivos à saúde, o interesse da sociedade sobre os efeitos da poluição aumentava e o Departamento de Defesa americano projetava sistemas de mísseis cada vez mais letais. A partir deste momento, diferentes abordagens surgiram no sentido de resolver diferentes problemas de segurança. Técnicas e regulamentações foram criadas para áreas específicas como da aviação civil, da defesa, da energia nuclear, dos processos da indústria química, entre outros.

Da engenharia da segurança surgiu o conceito de segurança de sistemas – ou sistema seguro (do inglês, *system safety*). A primeira preocupação das diversas questões relativas à segurança de sistemas foi o gerenciamento dos perigos (no inglês, *hazards*) de um sistema: sua identificação, avaliação, eliminação e controle, através de procedimentos de análise, projeto e gerenciamento (ISAKSEN; BOWEN; NISSANKE, 1997).

Dentre as muitas definições para acidente, perigo e risco, a fim de evitar interpretações diferentes e conflitantes, adotam-se neste trabalho as seguintes definições:

Acidente: um evento indesejável e não planejado, mas não necessariamente inesperado que resulta em níveis específicos de perda de vida, ou lesões, danos a propriedade ou ambiente (LEVESON, 1995).

Perigo: um estado, ou conjunto de condições do sistema que, juntamente com outras condições do ambiente do sistema levarão a um acidente. (LEVESON, 1995). Uma situação física, como um estado do sistema ou condições ambientais, que pode resultar em um acidente (no inglês, *hazard*).

Risco: ocorre em função da probabilidade de surgir um evento perigoso e da severidade resultante de um perigo (LEVESON, 1995).

Uma das mais conhecidas análises técnicas que busca criar um entendimento sobre as questões da segurança em um sistema sob investigação é a técnica de análise

de perigos (no inglês, *hazard analysis*) (STOREY, 1996). O método de análise de perigos abrange uma série de métodos que foram criados, muitas vezes, para atender uma área específica, e outras que tiveram uma ampla aceitação pela comunidade da segurança. Dentre as diversas técnicas de análise de perigo criadas ou apresentadas por diversos autores (LEVESON, 1995) (STOREY, 1996) (ISAKSEN; BOWEN; NISSANKE, 1997) (REDMILL; CHUDLEIGH; CATMUR, 1999), destacam-se:

- Análise Preliminar de Perigos – PHA (no inglês, *Preliminar Hazard Analysis*);
- Revisões (no inglês, *Reviews and Walk-throughs*);
- Estudo de Perigos e Operacionalidades – HAZOP (no inglês, *HAZard and OPerability Study*);
- Análise dos Modos de Falha e Efeitos – FMEA (no inglês, *Failure Modes and Effects Analysis*);
- Análise dos Modos de Falha, Efeitos e Criticalidade – FMECA (no inglês, *Failure Modes, Effects and Criticality Analysis*);
- Análise de Árvore de Falhas – FTA (no inglês, *Fault Tree Analysis*);
- Análise de Árvore de Eventos – ETA (no inglês, *Event Tree Analysis*);
- Análise de Causa e Conseqüência – CCA (no inglês, *Cause –Consequence Analysis*); e
- Análise Probabilística de Risco – PRA (no inglês, *Probabilistic Risk Analysis*).

Figura 2.5 - Técnicas de análise de perigos no ciclo de vida do produto.

Estas técnicas podem ser usadas nas diversas fases do ciclo de vida de um produto, conforme ilustrado na Figura 2.5, para identificar os perigos, avaliar o risco associado e identificar medidas que podem ser tomadas para eliminar ou controlar este perigo, ou para reduzir o risco associado.

A técnica FMEA foi originalmente desenvolvida no Departamento de Defesa americano, quando foi criado o procedimento MIL-P-1629, intitulado *Procedures for Performing a Failure Mode, Effects and Criticality Analysis*, de 9 de novembro de 1949. Foi usado como uma técnica de avaliação da confiabilidade para determinar os efeitos de falhas nos equipamentos e no sistema. Estas falhas foram classificadas de acordo com seu impacto no sucesso da missão e na segurança de pessoal e equipamento. Mais tarde este padrão foi substituído pelo DOD-MIL-STD-1629A (1980). Fora do meio militar, a área espacial está entre as primeiras a aplicar em seus projetos a disciplina segurança de sistemas. Foi após o incêndio do foguete Apollo em 1967, durante testes na plataforma de lançamento em Cabo Canaveral, e que vitimou três astronautas. O acidente alertou a Agência Espacial Americana (NASA) (no inglês, *National Aeronautics and Space Administration*) para o desenvolvimento de políticas e procedimentos que tornassem as atividades aeroespaciais mais seguras. A partir desta época, um extensivo programa de segurança de sistema foi ajustado para projetos espaciais, sendo que muito do conhecimento foi absorvido dos programas da força aérea e do departamento de defesa americano. A FMEA passou a ser utilizada de forma mais abrangente no início dos anos 80, quando as indústrias automotivas americanas *Ford Motors Company*, *Chrysler Corporation* e *General Motors Corporation* começaram a utilizar o método na fabricação de automóveis e que culminou na criação do atual padrão industrial SAE-J-1739 (2000).

Os novos projetos de plantas químicas dos anos 50 e 60 eram os maiores até então construídos e operavam em temperaturas e pressões muito elevadas. O resultado foi um aumento significativo de acidentes sérios. Isto conduziu a uma abordagem mais sistemática e mais técnica com relação à segurança, conhecida como processo de segurança e de prevenção de perda (tradução livre do termo em inglês, *process safety and loss prevention*) (KLETZ, 1999). Esta abordagem procura distinguir questões como a quantificação do risco, a identificação dos perigos, auditorias, a

criação de projetos mais seguros, uma investigação mais completa dos incidentes, e o estudo mais aprofundado sobre a situação da planta antes de uma explosão, evacuação e a dispersão dos gases. A técnica HAZOP foi desenvolvida pelo *Imperial Chemical Industries* na Inglaterra no início dos anos 60, e mais tarde foi melhorada e publicada pela Associação das Indústrias Químicas, em Londres (LEVESON, 1995). A técnica tinha o intuito de identificar todos os possíveis desvios de projeto e operação de uma planta química, e todos os perigos associados a este desvio. Basicamente a técnica encoraja a criatividade, pesando todas as possibilidades em que um perigo ou um problema de operação pode ocorrer. A diferença básica entre a aplicação de HAZOP na indústria química e sua aplicação em outros campos da engenharia é a maneira como a documentação de projeto é examinada. Na indústria química o exame é conduzido através de um fluxograma, isto é um diagrama esquemático que mostra a conexão dos vasos, das tubulações e das válvulas. Nas aplicações de engenharia, pode ser usada uma representação alternativa das partes e suas interações, como um desenho mecânico, um circuito esquemático ou um diagrama de fluxo de dados. Esta mesma técnica pode ser aplicada em inúmeros níveis de um projeto (YELLOWBOOK3, 2000).

2.2.1 Técnica HAZOP

A técnica, ou método de Estudo de Perigos e Operacionalidades, conhecido como HAZOP, foi inicialmente desenvolvida nos anos 60, e é baseada no exame sistemático dos fluxos dos processos de uma planta de engenharia química, de modo a identificar e avaliar problemas que podem representar riscos a pessoas ou equipamentos, ou prejudicar a eficiência de sua operação. Atualmente seu uso foi adotado para sistemas de outras áreas em operações complexas e para sistemas de software (RAUSAND, 2004).

HAZOP é uma técnica de análise qualitativa, baseada em palavras-guias (em inglês, *guide-words*) e é conduzida por uma equipe multidisciplinar de especialistas em um conjunto de reuniões (SOUZA, 1995). As palavras-guia são aplicadas em cada fluxo de um sistema, sugerindo os possíveis desvios de operação normal, como por

exemplo, *nenhum*, *mais*, *menos* ou *reverso*. Esta análise considera, por exemplo, o que ocorre se muito ou pouco de certo elemento químico é passado através de um tanque de um processo industrial para outro. Para isto, a equipe HAZOP segue o estudo, por meio das recomendações de condução da análise, e se for julgado necessário, sugere instruções de melhoria no intuito de mitigar ou detectar um risco ou uma falha.

O método HAZOP é principalmente indicado quando da implantação de novos processos na fase de projeto ou de modificação, tanto de grande como de pequeno porte. O ideal é que o estudo seja desenvolvido antes da fase de detalhamento e construção, evitando com isso que modificações tenham que ser feitas depois que o processo já esteja em operação. O caráter multidisciplinar da equipe de trabalho HAZOP faz com que a criatividade individual seja estimulada, melhorando a compreensão dos problemas das diferentes áreas e das interfaces do sistema.

O estudo é feito de forma sistemática, levando em consideração alguns tópicos importantes na condução do processo como nós-de-estudo (no inglês, *study nodes*), que são os pontos do processo que serão analisados nos casos em que possivelmente ocorram desvios; intenção de projeto/operação (no inglês, *design intent*), que define o comportamento específico de certo aspecto do projeto ou de sua operação; desvio (no inglês, *deviation*), que representa o deslocamento da intenção de projeto ou operação; perigo (no inglês, *hazard*), que representa uma situação, seguida de algum evento iniciador, que pode levar a um acidente; palavras-guia, que são palavras simples utilizadas para qualificar os desvios de intenção de projeto, e para guiar e estimular o grupo HAZOP, durante o processo de estudo.

Em termos gerais, HAZOP é bastante semelhante a qualquer outra técnica de análise de perigos, como a da PHA ou da FMEA. Entretanto, a análise realizada por HAZOP é feita através de palavras-chave, ou palavras-guia, que orientam o raciocínio dos grupos de estudo multidisciplinares, focando os perigos mais significativos para o sistema. Cada palavra-guia é aplicada às variáveis identificadas no processo (pressão, temperatura, fluxo, composição, nível, etc.) identificando os

desvios que podem ocorrer caso a condição proposta pelas palavras-guia seja pertinente.

2.2.2 Palavras-guia HAZOP

Segundo Redmill; Chudleigh e Catmur (1999) a palavra-guia é uma palavra ou frase que expressa e define um tipo específico de desvio de operação de uma intenção de projeto. É parte chave do estudo HAZOP, usada como entrada para o estudo, extração de idéias e fomento das discussões, de modo a maximizar as chances de identificar e estudar todos os perigos potenciais da planta analisada.

As palavras-guia são aplicadas aos parâmetros de processo estabelecidos pela intenção de projeto e operação, em todos os nós de estudo da planta em análise, procurando observar seus possíveis desvios. Assim, as palavras-guia são utilizadas para levantar questões como: “O que aconteceria se o fluxo fosse mais...?” ou “E se o fluxo fosse reverso de?” Podem ser aplicadas tanto a parâmetros de processo mais gerais (por exemplo, reagir, transferir) como a parâmetros mais específicos (por exemplo, pressão, temperatura).

Diversos tipos de palavras-guia para processos industriais podem ser utilizados, podendo ocorrer certas variações, dependendo do tipo da aplicação. A seguir é apresentado um conjunto de palavras-guia e sua interpretação para sistemas genéricos e para sistemas eletronicamente programáveis, os chamados PES (no inglês, *Programmable Eletronic System*) (REDMILL; CHUDLEIGH; CATMUR, 1999).

Não (Nenhum): completa negação da intenção de projeto; nenhuma parte da intenção é alcançada; nada acontece; nenhum dado ou sinal de controle fornecido.

Mais (mais alto): acréscimo quantitativo; mais dados são fornecidos do que o esperado.

Menos (mais baixo): decréscimo quantitativo, menos dados do que o esperado.

Além de: acréscimo qualitativo, onde toda a intenção de projeto é alcançada em conjunto com uma atividade adicional.

Parte de: decréscimo qualitativo onde somente parte da intenção de projeto é alcançada; informação passada está incompleta.

Reverso (ao contrário de): oposto lógico da intenção; informação geralmente não confiável.

Diferente: substituição completa, onde nenhuma parte da intenção de projeto é alcançada; informação completa, mas incorreta.

Quando é relevante examinar a temporização do processo estudado, outras palavras-guia são necessárias:

Cedo: algo acontece mais cedo em relação ao tempo de intenção do projeto; fluxo de informação ocorre antes do que foi projetado.

Tarde: algo acontece mais tarde em relação ao tempo de intenção do projeto; fluxo da informação ocorre depois do que foi projetado.

Antes: algo acontece antes da seqüência da intenção do projeto.

Depois: algo acontece depois da seqüência da intenção do projeto.

As palavras-guia, aplicadas com os parâmetros gerais, dão origem aos desvios de intenção de projeto ou operação. Estes desvios podem até ter mais de um significado para um mesmo parâmetro. Souza (1995) cita, por exemplo, a aplicação da palavra-guia “mais” ao parâmetro “reação” gera um desvio potencial “mais reação”, podendo significar que a reação está ocorrendo a uma velocidade maior ou que uma maior quantidade de reagentes está reagindo. Mesmo significando que ambas resultam em maior quantidade de produto, pode ser necessário algumas modificações das palavras-guia. Também é importante observar que em determinadas situações, alguns desvios potenciais são eliminados devido a limitações físicas dos parâmetros de processo em análise. Pode ser o caso do parâmetro temperatura, que possui possibilidade de análise praticamente só com as palavras-guia “mais” e “menos”. É prerrogativa da equipe que aplica HAZOP estabelecer quais combinações são significativas e quais não produzem efeitos no sistema.

No caso específico de sistemas eletrônicos digitais, foco da análise de perigos conduzida nesta tese, o estudo deve ser endereçado a todas as partes como

hardware, o software que é executado nele, e outros componentes associados como sensores e atuadores. Para ilustrar este tipo de aplicação Redmill; Chudleigh e Catmur (1999) apresentam alguns exemplos de aplicação de palavras-guia para hardware digital e sistemas de comunicação, levando em conta diversos atributos de seus componentes como a capacidade de condução do fio e protocolo de comunicação. A Tabela 2.1 mostra a matriz de atributos versus interpretação de algumas palavras-guia do exemplo de Redmill; Chudleigh e Catmur (1999).

Tabela 2.1 – Interpretação de atributos por palavras-guia.

Atributo	Palavra-guia	Interpretação
Capacidade de condução do fio	NENHUM (Nenhuma intenção foi alcançada)	Falta fio ou está quebrado.
	MAIS (Uma quantidade maior)	Alta voltagem ou alta corrente.
	MENOS (Uma quantidade menor)	Corrente muito baixa.
	ALÉM DE (Atendeu solicitação com resultados adicionais)	Interferência ou Interferência Eletro-Magnética.
	REVERSO (O oposto da solicitação foi atendido)	Comportou-se como uma linha de transmissão.
	OUTRO QUE (Outro resultado que o proposto)	Fio errado levando a um sinal errado
Atributo	Palavra-guia	Interpretação
Protocolo de comunicação	NENHUM (Nenhuma intenção foi alcançada)	Nenhum protocolo implementado.
	ALÉM DE (Atendeu solicitação com resultados adicionais)	Outro controle de sinal. (outro <i>hand-shake</i>).
	PARTE DE (Somente parte da intenção do projeto é alcançada)	Resposta ou comando parcial.
	CEDO (Em relação ao tempo do relógio)	Erro relativo ao tempo do relógio (tempo mais cedo).
	DEPOIS (Depois relacionado à ordem ou seqüência)	Erro relativo a seqüência esperada (depois da ordem correta).

Fonte: (REDMILL; CHUDLEIGH; CATMUR, 1999).

Deve ficar claro que um conjunto de palavras-guia genéricas pode ser útil em uma primeira abordagem para qualquer tipo de sistema, incluindo os eletronicamente programáveis. Entretanto é preciso ficar claro que palavras-guia podem ser interpretadas de forma diferente dependendo do ambiente ou contexto em que se insere o estudo. O tipo de desenvolvimento industrial e o ciclo de vida do sistema, por exemplo, podem sugerir diferentes tipos de abordagens para as palavras-guia e sua interpretação.

2.2.3 Técnica FMEA

A técnica de Análise de Modos de Falha e Efeitos (FMEA) foi desenvolvida no final dos anos 40, como um procedimento de segurança militar norte-americano (conhecido como procedimento MIL-P-1629) para determinar as falhas e efeitos no sistema e nos equipamentos e posteriormente, nos anos 60, foi utilizada na indústria aeroespacial durante o programa Apollo. As falhas eram classificadas de acordo com o seu impacto no sucesso da missão e na segurança do pessoal, selecionando componentes individuais ou funções do sistema e investigando seus possíveis modos de falha.

Este método pode ser aplicado tanto para o desenvolvimento de um produto (visão de processo) como para um produto já finalizado (visão de produto). As etapas e a maneira de realização da análise são as mesmas, diferenciando-se somente quanto ao objetivo. FMEA primeiramente é realizada de forma qualitativa, podendo ser aplicada em vários estágios do desenvolvimento, apesar de muitas vezes ser utilizada ainda nos estágios iniciais de um projeto para determinar seu nível de integridade de segurança (STOREY, 1996). O objetivo é estabelecer um número que corresponda à probabilidade geral do produto operar sem falha durante uma quantidade de tempo específica, ou alternativamente, que o produto operará durante certo tempo entre falhas (LEVESON, 1995). Para isto é necessário que se defina como e com qual frequência os componentes do produto podem falhar, sendo então a análise estendida para avaliar os efeitos de tais falhas. Nesta fase se procede a uma análise quantitativa, estabelecendo valores da confiabilidade dos componentes

e do sistema. É analisada a probabilidade de falha do sistema ou subsistema, através do cálculo de probabilidades de falhas individuais de seus componentes, bem como na determinação de como poderiam ser reduzidas estas probabilidades, inclusive pelo uso de componentes com confiabilidade alta ou pela verificação de redundâncias de projeto.

A técnica FMEA determina a probabilidade de falha de componentes individuais, raramente envolvendo uma investigação sobre falhas decorrentes de falhas de múltiplos componentes ou se o componente operou com sucesso. Hammer (1972) acredita que o uso de FMEA se mostra bastante útil com relação às falhas advindas das características funcionais dos equipamentos, condições ambientais adversas e erros humanos com relação a procedimentos operacionais.

2.2.4 Palavras-guia no contexto do HAZOP e FMEA

Palavras-guia podem ser interpretadas de maneira diferente para diferentes tipos de aplicações, em diferentes estágios do ciclo de vida de um produto. Elas devem ser adequadas de modo a refletir um entendimento claro de quais são os possíveis desvios de intenção de uso dos componentes do sistema investigado, dentro de um determinado projeto. Apesar de uma possível controvérsia sobre a definição de um conjunto completo de palavras-guia, suficiente para abranger todos os desvios possíveis de intenção, deve-se definir um conjunto mínimo e abrangente, tanto quanto possível, de forma a permitir uma exploração criativa dos perigos de operação do sistema estudado.

Redmill; Chudleigh e Catmur (1999) afirmam que este tipo de estudo para sistemas programáveis ainda carece de melhores práticas e que o uso de interpretações muito específicas tendem a mascarar uma exploração criativa. Acrescenta ainda que, durante o processo de investigação dos perigos, podem surgir questões que obriguem a reavaliação do conjunto de palavras-guia usado e, conseqüentemente, uma nova discussão pode ser iniciada para a inclusão de alguma nova interpretação ou mesmo de uma nova palavra-guia.

No contexto do processo para aplicações espaciais, a definição das palavras-guia se baseou no estudo de operabilidade e risco HAZOP, e também nos potenciais modos de falha, descritos nas diversas abordagens da técnica de análise de modos de falhas e seus efeitos FMEA. Tanto as técnicas HAZOP como FMEA deram origem a diversas variações de abordagens, como a proposta por Pumfrey (2000), e conhecida como SHARD/LISA (no inglês, *Software Hazard Analysis and Resolution in Design/ Low Level Interaction Safety Analysis*), ou Software FMEA (da sigla, SFMEA) adaptado por Lutz e Woodhouse (1996). Em ambos os casos são feitas adaptações de uso de palavras-guia dos modelos tradicionais de análise de perigos.

As palavras-guia convencionais do HAZOP e Software HAZOP incluem um grupo de palavras que expressam e definem tipos específicos de desvios de intenção de projeto (no inglês, *deviation from design intent*) relacionados à PES. As palavras-guia de HAZOP que exploram potenciais desvios são: *Não, Mais, Menos, Parte de, Oposto, Diferente, Cedo, Tarde, Antes e Depois*.

Como analisado por Conmy (2005), SHARD/LISA são mais apropriadas para desvios de hardware/software, enquanto SFMEA é usada para verificar requisitos de software, mais especificamente, requisitos de software em naves espaciais.

A técnica SHARD examina os desvios de fluxo de informações, iniciando com as saídas do sistema ou de suas funções, e trabalhando no sentido das entradas. A técnica LISA examina os desvios de eventos de tempo, como interrupções, e recursos físicos usados na operação do sistema. Já a técnica SFMEA (LUTZ; WOODHOUSE, 1996, 1997) foi também aplicada a requisitos de software para veículos espaciais e analisa eventos de software e fluxo de dados categorizados como falhas de eventos e de dados. Por outro lado, SHARD/LISA utiliza palavras-guia, baseadas em uma série de análises de diferentes sistemas computacionais, agrupadas conforme a classe da falha, ou seja: *Omissão, Comissão, Cedo, Tarde e Valor*.

Lutz e Woodhouse (1996) iniciaram sua análise de FMEA focada em software, observando primeiramente os efeitos das falhas e depois identificando suas causas. Classificaram as palavras-guia baseadas em fluxo de dados e fluxo de eventos. Para

fluxo de dados identificaram as palavras-guia *Falta de Dados*, *Dados Incorretos*, *Dados Duplicados*, *Temporização Errada dos Dados*. Para fluxo de eventos identificaram as palavras-guia *Término Anormal*, *Omissão*, *Lógica/Evento Incorreto*, *Ordem/Temporização Errada*.

Tendo como referência estes trabalhos, foram classificadas nesta tese as palavras-guia em dois grupos: Fluxo de Dados e Fluxo de Eventos e Serviços.

Palavras-guia relativas à Fluxo de dados:

Dado Insuficiente/Ausente: perda ou falta de mensagens; ausência de entrada de sensor; falta de entrada ou saída, falha no recebimento de dados necessários; perda de dados devido a falha de hardware, falha do processo de software ou no sensor para enviar os dados necessários para o correto funcionamento do modelo de software.

Dado Incorreto: dado ruim ou qualquer recurso que não descreva corretamente o estado da espaçonave ou seu ambiente operacional; erro em valores relativos a rotinas de acionamento de disparadores (*triggers*); parâmetros errados relativos aos limites de atraso de tempo; parâmetros errados para comandos certos, sinais espúrios ou inesperados em comandos de saída.

Erro na Temporização do Dado: dado disponibilizado muito tarde para ser usado; ou para ter a precisão necessária (tarde); dado disponibilizado cedo para ser usado ou para ter a precisão necessária (cedo); dado obsoleto usado na decisão de controle (no inglês, *data age*); dados espúrios, inadvertidos ou falhos que ocorre somente com algumas entradas.

Dado Duplicado: dado saturado ou transbordamento de dados (no inglês, *overflow*).

Valor: informação entregue com um valor errado; falha na detecção de algum valor esperado ou não.

Palavras-guia relativas à Fluxo de Eventos e Serviços:

Término Suspensão/Anormal: evento em aberto, pendurado; em um impasse (no inglês, *deadlock*) neste ponto do processo.

Lógica/Evento Incorreto: o comportamento está errado, a lógica está errada; a lógica está invertida; considerações erradas sobre o estado e sobre pré-condições; a intenção ou o requisito não reflete a implementação.

Omissão: ocorre uma falha de um evento ou serviço, mas o processo continua executando; saltos (no inglês, *jumps*) e desvios (no inglês, *skips*); um componente não está respondendo a uma entrada; produz uma saída não esperada; o serviço nunca é entregue; uma comunicação simples é omitida; muitas (ou até todas) iterações consecutivas são omitidas.

Comissão: o serviço é entregue quando não é requerido; uma comunicação esperada é repetida quando não deveria ser; ocorre uma comunicação completamente inesperada.

Temporização: evento ocorre no tempo errado; ou na ordem errada; evento ocorre muito cedo (premature: o sistema não estava no modo apropriado para receber ou processar o evento); muito tarde, a seqüência dos eventos é incorreta; um evento que deveria preceder outro não ocorre como deveria; eventos iterativos ocorrem intermitentemente ao invés de regularmente; eventos que deveriam ocorrer somente uma vez ocorrem iterativamente.

2.3 SISTEMAS CRÍTICOS DA ÁREA ESPACIAL

Um marco histórico no desenvolvimento espacial aconteceu na metade do século passado, quando em 1957 a nave espacial russa Sputnik foi lançada em órbita. Desde então avanços significativos da tecnologia ocorreram, como a exploração da Lua e as viagens a diversos planetas do sistema solar. Naves já pousaram em Vênus e Marte, já conseguiram orbitar o planeta Júpiter, as luas de Saturno e, recentemente, iniciaram-se projetos mais ambiciosos como o programa Roseta, que pretende pousar em um cometa. Outro projeto ambicioso é o da Estação Espacial Internacional – ISS (no inglês, *International Space Station*), que criou um programa de longa permanência do homem no espaço, envolvendo diversos países da Europa, Estados Unidos e Rússia, tendo também o Brasil como participante.

Mais recentemente, um dos mais notáveis avanços na tecnologia espacial foi proporcionado com a utilização da computação eletrônica e do software. Não obstante estes recursos não terem sido desenvolvidos especificamente para a área de espaço, a capacidade tecnológica que oferecem foi rapidamente assimilada, e revolucionou, entre vários aspectos, a flexibilização dos projetos espaciais. Porém o uso destes recursos alternou entre ocorrências de falhas potenciais e grandes sucessos como evidenciados pelo projeto do Voyager 2 (STARK; SWINED; TATNALL, 2004).

A tecnologia digital tem criado uma revolução silenciosa na maioria dos campos da engenharia, mas a engenharia de sistemas, e as técnicas de engenharia de segurança de sistemas, não mantiveram o mesmo passo. De acordo com Leveson (2004) e Johnson (2003) os sistemas digitais estão mudando a natureza dos acidentes. Muitas das abordagens para prevenir acidentes que trabalharam com componentes eletromecânicos – como a replicação de componentes para proteger contra falhas de componentes individuais – são ineficazes em controlar acidentes que surgem do uso de sistemas digitais e de software.

Falha na cultura de segurança, problemas organizacionais, deficiências técnicas e uma inadequada engenharia cognitiva, desempenharam papel importante nos acidentes aeroespaciais recentes, como o da explosão do lançador Ariane 5 no voo realizado em 1996, a perda de contato da nave SOHO em 1998; a perda da nave Mars Climate Orbiter também em 1999, a destruição do foguete Mars Polar Lander em algum momento entre as fases de entrada e pouso no ano seguinte, a satelização em uma órbita incorreta e sem uso do satélite militar americano Milstar pelo lançador Titan IV B-32/Centaur em 1999. Em alguns destes acidentes, os recursos disponíveis de engenharia de sistema foram insuficientes para atender as necessidades do projeto. Em outros, o processo seguido foi falho, como na decomposição dos requisitos de sistema para requisitos de software, ou na coordenação e comunicação entre parceiros e equipes do projeto.

2.3.1 Sistemas espaciais

Um sistema espacial pode ser caracterizado através da visão do projeto de um veículo espacial. Quando se observa o sistema espaçonave, estão se considerando diversos elementos funcionais ou subsistemas. Podem ser incluídos, de forma mais genérica sistema estação de solo, sistema carga útil, e sistema veículo.

No sistema estação de solo, por exemplo, podem ser identificados diversos subsistemas e atores como sistema de apoio em solo para lançamento do veículo, sistema de apoio em solo para controle do satélite, plataforma de lançamento, umbilicais, sistema de meteorologia, de rastreamento do veículo, de segurança de voo, de recuperação de carga, de aquisição de dados, entre outros. Em um nível mais detalhado, por exemplo, no sistema de apoio em solo para lançamento do veículo podem-se observar subsistemas como o suprimento de energia, de controle, de fluidos do veículo, etc. Como atores humanos é possível identificar, entre outros, o operador do sistema em solo, operador de controle do satélite, operador de segurança do lançamento do veículo e operador do radar de rastreamento.

No sistema carga útil estão identificados atores, como o subsistema satélite e o subsistema de experimentos embarcados. No subsistema satélite, por exemplo, estão identificados atores humanos e artificiais como o operador de telecomando, sensor inercial de posição e atitude do satélite, sensor de temperatura, tubeiras móveis, computador de bordo, sistema de energia.

No sistema veículo estão identificados dois tipos principais de veículos: foguetes tripulados e não tripulados. As naves não tripuladas incluem lançadores (de cargas úteis, como satélites) e veículos de sondagem. Nos lançadores de satélite podem-se observar atores como sensor inercial de posição e atitude, sistema de atuadores, sistema de estágios do veículo, computador de bordo, sistema de telemetria a bordo, sistema de baterias, propulsão, entre outros.

Existem ainda classes de sistemas mais específicas que, segundo o ponto de vista computacional, poderiam ser classificadas como sistemas de informação, sistemas

embarcados e sistema de comando e controle (KOTONYA; SOMMERVILLE, 2000). Cada uma dessas classes de sistema possui um conjunto de propriedades emergentes que se manifestam quando seus subsistemas são integrados, tratando não somente das propriedades que dizem respeito às suas funcionalidades, mas também propriedades relacionadas a requisitos de qualidade, como confiabilidade, capacidade de manutenção, desempenho, facilidade de uso, segurança, etc.

De modo geral, o propósito da missão de um veículo espacial é de por no espaço equipamentos ou pessoas para cuidar de atividades que não podem ser realizadas na Terra. Pisacane e Moore (1994) afirmam que o projeto de cada equipamento é fortemente influenciado pela missão específica, seu tempo de vida útil, de como são os meios de lançamento deste veículo, incluindo seu ambiente, e o ambiente espacial onde este veículo irá transitar.

A decomposição da engenharia de um sistema espacial pode ser feita de diversas formas e abordagens, levando-se em conta os objetivos e os requisitos da missão primeiramente. Segundo Stark; Swinerd e Tatnall (2004), os objetivos da missão consideram o ponto de vista dos diversos usuários do sistema, como por exemplo, os usuários técnicos e gerenciais, e restrições tanto políticas como financeiras. Os requisitos de missão levam em consideração questões como desempenho, confiabilidade, custo, vida útil, entre outros. A partir dos objetivos e requisitos de missão, o sistema é decomposto nos requisitos do sistema veículo, que são desdobrados em segmento veículo e segmento solo. Cada um dos elementos do sistema interage entre si, e uma das metas principais dos projetistas de foguetes é obter um rendimento máximo entre eles, de modo a cumprir os objetivos da missão proposta.

Os requisitos do sistema veículo observam qual será a órbita a ser alcançada, qual a massa do veículo, sua capacidade de propulsão (energia), sua operação e a configuração do veículo. O sistema veículo leva em conta o volume da carga, o ambiente pela qual esta será submetida e a sua distribuição de sua massa. O segmento solo leva entre outras considerações, como será o apoio operacional em terra, o processamento dos dados do veículo, as atividades de segurança em solo. Estes dois segmentos possuem diversos subsistemas como, por exemplo, o de

potência, de estrutura, o eletrônico, o de controle de atitude, o térmico, e o de comunicação.

2.3.2 A engenharia de sistemas e de software espacial

Stark; Swined e Tatnall (2004) citam do Dicionário de Ciência e Tecnologia (em inglês, *Chambers Science and Technology Dictionary*), como definição de engenharia de sistemas espaciais, o processo lógico de atividades que transformam um conjunto de requisitos advindos dos objetivos de uma missão específica para uma descrição completa de um sistema que preenche os objetivos de forma ótima. Pisacane e Moore (1994) definem a engenharia de sistemas espaciais como um processo de desenvolvimento que consiste primeiramente em traduzir as necessidades do sistema em requisitos de sistema, estes requisitos em especificação de subsistemas, estas especificações em projeto de subsistemas, que possam ser fabricados e testados, e finalmente estes subsistemas integrados, tanto o veículo como seu lançador e sua operação de lançamento. Todos estes passos devem ser efetivados por diversos grupos dentro de um cronograma e de um custo estimado previamente no processo, sem perder o foco da missão ou do sistema como um todo.

2.3.2.1 Engenharia de sistemas espacial

A forma com que o processo de engenharia de sistema é conduzido varia dependendo de como o programa espacial de um país ou organismo é definido. Geralmente é dividido em fases distintas e estas fases podem ainda sofrer adaptações, dependendo de como este programa espacial está sendo gerenciado. Partes de um programa espacial podem, por exemplo, ser terceirizados para outras empresas e instituições, fazendo com que determinadas fases e processos do programa sejam revistos. Isto vale tanto para a engenharia de sistemas como para a engenharia de software. O tempo gasto para a realização de um projeto espacial,

consumido em boa parte em atividades da engenharia, pode ser de vários anos, como no caso da missão Rosetta da ESA para estudo dos pequenos corpos do Sistema Solar, os asteróides e cometas. O projeto foi aprovado em 1993, lançado em 2003 e tem previsão para alcançar o cometa Wirtanen em 29 de novembro de 2011.

Tatnall; Farrow e Francis (2004) apresentam cinco fases de um processo de engenharia de sistemas espacial: (A) estudo de viabilidade, (B) definição detalhada, (C/D) desenvolvimento, manufatura, construção e teste, (E) operação da missão e análise dos dados. A Associação Brasileira de Normas Técnicas (ABNT) publicou em 2002 a norma ABNT-NBR-14857-2 (2002) que fornece uma visão e os requisitos de gerenciamento de um programa espacial, com o objetivo global de otimizar desempenho, custos e cronogramas, e de minimizar riscos de acordo com as normas da ISO e dos padrões da ESA. A ESA propõe um ciclo de desenvolvimento de produtos espaciais em sete fases (ECSS-M-ST-10C, 2009): (0) Análise de Missão /Identificação das Necessidades; (A) Viabilidade; (B) Definição Preliminar; (C) Definição Detalhada; (D) Produção/Teste de Qualificação em Terra; (E) Utilização; (F) Descontinuidade (descarte). O ciclo de vida do produto também abrange o contexto do software no ciclo de desenvolvimento do produto (ECSS-E-ST-40C, 2009).

(0) Análise de Missão /Identificação das Necessidades: engloba questões relativas à identificação e análise das necessidades da missão, e deve permitir:

- A identificação e caracterização da intenção da missão;
- Sua expressão em termos de necessidades, desempenho esperado e as metas de dependabilidade e segurança;
- Avaliação das restrições de operação, em particular do ambiente físico e operacional;
- A identificação dos possíveis conceitos do sistema, com ênfase no grau de inovação e qualquer aspecto crítico; e
- A avaliação preliminar dos dados do gerenciamento do projeto (organização, custos e cronograma).

Ao final desta fase deve ocorrer uma Revisão de Definição de Missão – MDR (no inglês, *Mission Definition Review*).

(A) Viabilidade: resulta na finalização da expressão das necessidades da fase 0, com propostas de solução para as seguintes necessidades observadas:

- A caracterização e quantificação dos elementos críticos (técnicos e econômicos);
- O estabelecimento de uma Árvore Funcional contendo os componentes do sistema e suas principais atividades;
- A exploração dos vários possíveis conceitos do sistema, através de modelos que determinem suas características e os elementos críticos;
- A identificação das restrições relativas a custo, cronograma, organização, utilização, produção e descontinuidade e margens estimadas; e
- Execução das atividades de engenharia de sistemas relacionada a software, definindo quais requisitos são de atribuição do software.

Ao final desta fase deve ser conduzida uma Revisão Preliminar de Requisitos – PRR (no inglês, *Preliminary Requirements Review*).

(B) Definição Preliminar: esta fase deve abranger, entre outros itens:

- A definição precisa da solução técnica escolhida durante a fase A;
- A identificação do que será desenvolvido internamente ou terceirizado;
- A avaliação das técnicas, tecnologias e meios para a implementação;
- A avaliação dos custos de confecção, produção e operação;
- A avaliação da confiabilidade e segurança;
- A avaliação do impacto ambiental;
- A garantia de coerência entre os requisitos especificados para os componentes, tanto em alto nível como em um nível detalhado;
- A decomposição das tarefas e as interfaces externas e internas do sistema; e
- Análise de Requisitos e Projeto de Arquitetura de Software.

Durante o decorrer desta fase deve ser realizada a Revisão de Requisitos do Sistema – SRR (no inglês, *System Requirements Review*) e a Revisão Preliminar de Projeto – PDR (no inglês, *Preliminary Design Review*).

(C) Definição Detalhada: além do estudo da solução detalhada e da definição do sistema, esta fase deve permitir:

- A decisão definitiva de fazer ou comprar o produto, se necessário;
- A confirmação das condições de configuração, teste e qualificação, e a iniciação dos métodos e meios de produção e verificação;
- O início da avaliação ou qualificação das tecnologias, bem como da atividade de avaliação de parceiros (em inglês, *procurement*); e
- Projeto e Implementação de Software.

Ao final desta fase deve ocorrer a Revisão Crítica de Projeto – CDR (em inglês, *Critical Design Review*).

(D) Produção/Teste de Qualificação em Terra: fase que vai até o fim do desenvolvimento do sistema, e deve permitir:

- A definição da qualificação do produto, dos componentes e do sistema como um todo;
- A qualificação do processo de apoio em solo;
- A produção do material de software e componentes necessários para a obtenção dos resultados experimentais (modelo de qualificação);
- A confirmação dos métodos de qualificação, procedimentos e meios de produção e verificação, permitindo a realização das tarefas de manufatura, montagem, integração e verificação;
- Verificação e validação do software; e
- Entrega e aceitação do software.

Durante esta fase é conduzida a Revisão de Qualificação – QR (em inglês, *Qualification Review*) e a Revisão de Aceitação – AR (no inglês, *Acceptance Review*).

(E) Utilização: esta fase compreende a campanha de lançamento, o lançamento e a aceitação em voo dos elementos do sistema espacial, se necessário. Corresponde também a operação e manutenção do sistema.

Durante esta fase é realizada a Revisão de Preparação para Operação – ORR (no inglês, *Operational Readiness Review*) e revisão de Preparação para Voo – FRR (no inglês, *Flight Readiness Review*).

(F) Descontinuidade: esta fase cobre todos os eventos de fim de vida do produto ou seu descarte. Geralmente ela é preparada durante a fase de utilização do sistema, e é especificado para cada produto em questão. Um inventário completo é revisado com respeito ao seu uso futuro ou sua disposição para avaliação do seu impacto ambiental.

2.3.2.2 Engenharia de software espacial

Dentre os padrões de modelos de processo de desenvolvimento de software surgidos nos últimos anos, destacam-se o modelo da ISO/IEC-12207 (2001) e o padrão ISO/IEC-15504-2 (2003). O padrão 12207 descreve processos para desenvolvimento e manutenção de software, fornecendo um conjunto essencial de atividades que deverão ser completadas para obter um produto de software. Já o padrão 15504, derivado do modelo de maturidade organizacional do SPICE (da sigla em inglês, *Software Process Improvement and Capability dEtermination*) (SPICE, 1995), é utilizado na avaliação de projetos e organizações, de forma a verificar o nível em que essas se encontram, em termos de maturidade dos processos utilizados e das práticas organizacionais no desenvolvimento de projetos de software.

Seguindo essa tendência de padronização de processo de software, em 1994 a ESA adotou um novo conjunto de padrões, preparado pela ECSS (da sigla em inglês, *European Cooperation for Space Standardization*), e fortemente baseado no modelo de ciclo de vida do padrão 12207. O ECSS é um esforço cooperativo entre a ESA,

as agências espaciais dos países europeus e a indústria européia, com o objetivo de desenvolver e manter padrões relacionados a programas espaciais, no sentido da redução de custos e da melhoria da qualidade, através de um entendimento comum e do estabelecimento de técnicas e padrões (JONES et al., 2002).

A estrutura do padrão do ECSS pode ser vista através de três ramos principais: o de gerenciamento de Projeto (M), o da Garantia de Produto (Q), e o da Engenharia (E). Especificamente tratando-se de software, o padrão ECSS-E-ST-40C (2009) cobre todos os aspectos da Engenharia de Software Espacial. Ele é dividido em dois segmentos: espaço e solo. O segmento espaço (computador embarcado, sistema de manipulação de dados e sistemas de órbita e atitude), assim como o segmento de solo (sistemas de controle de missão, simuladores, sistemas de dinâmica de voo, ferramentas de análise de voo, sistemas de dados de estações terrestres, como processadores de telemetria e telecomando e redes de comunicações), contém software com uma considerável complexidade. O padrão define requisitos para as atividades de engenharia, associadas com diferentes processos de desenvolvimento de software. Esse padrão, entretanto, não pode ser mencionado sem referenciar o padrão ECSS-Q-ST-80C (2009). Este padrão define os requisitos de garantia de produto de software para o desenvolvimento e manutenção de software para sistemas espaciais, envolvendo requisitos relevantes a processos de forma individual. Adicionado a isto, os requisitos desse padrão auxiliam a garantir a qualidade, tanto do processo de desenvolvimento como do produto de software final. Como método para avaliação do processo de software para as atividades espaciais, a ESA produziu, em conformidade com os padrões ISO/IEC-15504, o S4S (da sigla em inglês, SPICE for Space) (LAHOZ; BURGARELI; MOURA, 2004).

O S4S é uma estrutura de processo (do inglês, *framework*) cujo objetivo é prover instrumentos e informações necessárias para avaliar a maturidade do processo de software (*Capability Determination Mode*), prover uma base sólida para melhoria do processo (*Process Improvement Mode*) e garantir que as práticas de software espacial (*ECSS Compliance Mode*) estão sendo seguidas de acordo com o estabelecido nos padrões ECSS. Como entrada para o desenvolvimento do modelo de avaliação S4S, utiliza-se o padrão ECSS, assim como nos resultados deste modelo, onde se leva em consideração requisitos específicos da área espacial. Para

evidenciar a importância dos requisitos relacionados com segurança, novos processos foram incorporados ao modelo de avaliação S4S dentro da categoria de processo de suporte. Esses processos cobrem as atividades requeridas para garantir a segurança do sistema e as atividades de verificação e validação independentes do sistema, mais especificamente o processo de garantia da segurança e dependabilidade (no inglês, *safety and dependability assurance process*) e o processo independente de verificação e validação – IV&V (da sigla em inglês, *independent verification and validation*).

Os padrões ECSS-E-ST-40C, ECSS-Q-ST-80C e o método de avaliação de processo S4S da ESA definem uma estrutura para engenharia, qualidade e avaliação e melhoria de processo de engenharia de software, e prescrevem práticas básicas para atingir certos níveis de maturidade. Esses modelos representam o processo de software sob a perspectiva funcional, mas não apresentam procedimentos definidos, como, por exemplo, a sequência de atividades básicas que o processo deve realizar. Isso fica a cargo da organização que implementa os processos sugeridos pelo padrão.

É importante ressaltar que as revisões são marcos importantes tanto na interação entre clientes e fornecedores, como para o processo de engenharia de requisitos. Na norma de software da ESA, a ECSS-E-ST-40C (2009), é comentado que um planejamento cuidadoso deve ser feito no interesse de descrever todas as restrições e interfaces do processo de engenharia de software no nível de sistema, mas ao mesmo tempo deve permitir ao fornecedor uma liberdade necessária para prover e gerenciar suas atividades e tarefas para o processo. Como regra geral, a participação da equipe de software neste tipo de projeto começa geralmente durante a Revisão de Requisitos de Sistema, quando são produzidas as informações relativas aos requisitos de missão que o software deve atender, e vai até sua entrega efetiva ao cliente, na Revisão de Aceitação. Os fornecedores de software participam ainda da Revisão de Preparação Operacional, quando são efetuadas as últimas verificações e ajustes do sistema para voo.

Na engenharia de software, as fases de Definição Detalhada e de Produção/Testes de Qualificação abrangem uma boa parte do esforço de trabalho desta disciplina

durante o ciclo de vida de desenvolvimento do sistema. Os processos de engenharia de requisitos de software e de arquitetura consomem uma grande, e muitas vezes subestimada, quantidade de esforço no desenvolvimento de software para sistemas espaciais. Como resultado da complexidade dos requisitos funcionais e de desempenho relacionados aos sistemas computacionais críticos, uma ênfase especial também deve ser empregada nas fases anteriores à fase de Definição Detalhada. O ideal é que a engenharia de software possa participar com profissionais de poder de decisão, desde a fase de Análise de Missão, principalmente para o software de voo, onde as funções projetadas para a missão espacial podem ser críticas.

Também é preciso salientar que a manutenção de software pode criar restrições adicionais a um projeto, visto que o tempo de vida operacional de um sistema pode exceder o tempo de vida útil de seus equipamentos computacionais, como no caso de um satélite espacial. O software que vai embarcado em voo é o único componente que pode sofrer uma manutenção e reparo, e até mesmo um re-projeto depois de lançado o veículo. Em casos extremos, o sistema de software é re-projetado, desenvolvido e carregado um novo aplicativo no veículo em voo (ECSS-E-ST-40C, 2009). Esta mesma caracterização vale para o software de solo. Uma outra questão interessante é que o processo de manutenção de um projeto pode perdurar por anos, dependendo do processo de qualificação de um veículo. Isto significa que a manutenção pode ser vista, durante o passar do tempo, como uma atualização do projeto, devido tanto às alterações impostas pelos avanços tecnológicos como pelas próprias mudanças de projeto.

2.3.3 Foguetes espaciais no Brasil

Em 1965, foram realizadas, em cooperação com a NASA, as primeiras atividades da área específica de veículos espaciais no Brasil. Naquela década, foi construído o Centro de Lançamento da Barreira do Inferno – CLBI, em Natal, Rio Grande do Norte, de onde aconteceram vários lançamentos com especialistas da NASA e, também, do Centro Aeroespacial Alemão (DLR, no alemão, *Deutsches Zentrum für*

Luft-und Raumfahrt) (RIBEIRO, 1999). Em 1965, o atual Instituto de Aeronáutica e Espaço – IAE lançou o primeiro foguete de sondagem brasileiro, Sonda I, dando início a uma família de foguetes: Sonda II, III e IV; VS-30 e VS-40. No final da década de setenta, o IAE já possuía tecnologia de foguetes de sondagem, que permitiu iniciar o desenvolvimento de um veículo controlado, seguindo uma trajetória previamente programada no solo. Para realizar o lançamento destes veículos, foi criado, além do CLBI em 1965, o Centro de Lançamento de Alcântara (CLA), no Maranhão, em 1984. Localizado nas proximidades da linha do equador (2°18'Sul) em frente ao Oceano Atlântico, numa região de estabilidade geológica e climatológica, o Centro dispõe de vasta extensão de áreas livres para a implantação de novas plataformas de lançamento, permitindo a operação de foguetes de vários portes e categorias.

Finalmente, no dia 2 de dezembro de 1997, o projeto de um primeiro veículo lançador de satélites, chamado VLS-1, se concretizou com o lançamento do primeiro protótipo, composto de quatro estágios, utilizando propelente sólido e com missão de executar um voo controlado que colocasse pequenos satélites de dados ambientais e de sensoriamento remoto em órbita baixa. O projeto do VLS-1 significou uma grande evolução em termos de capacidade de desenvolvimento de sistemas computacionais para aplicações espaciais no Brasil. A solução computacional tinha como função, na fase de pré-voo, atender a inúmeros requisitos do sistema como, interpretar e executar comandos emitidos pelo banco de controle em solo. A fase de voo tinha como requisito executar os algoritmos de navegação e controle, comandar a sequência de eventos de voo, desde a decolagem até a separação do quarto estágio e formatar e fornecer dados via telemetria, referentes às funções desempenhadas em voo (CASTRO LEITE, 1999). O veículo não cumpriu sua missão devido à falha no acendimento de um dos quatro motores do primeiro estágio do veículo.

O voo do segundo protótipo do VLS-1, denominado V02, também a partir do CLA, ocorreu no dia 11 de dezembro de 1999. Os primeiros 55,8 segundos de voo foram exatamente conforme o planejado até a ignição do segundo estágio. Aos 55,9 segundos de voo houve uma explosão que provocou a liberação irregular dos motores do primeiro estágio e a separação prematura entre o segundo e o terceiro

estágio. O segmento constituído dos motores do terceiro e quarto estágios, dos módulos superiores, inclusive a coifa contendo o satélite, permaneceram em voo, porém de forma não controlada, isto é, em trajetória balística. Aos 119 segundos, seguindo a programação prevista no software de missão, o computador de bordo comandou a ignição do motor do terceiro estágio, que funcionou em condições normais pelo tempo previsto, porém seguindo uma trajetória anômala. Aos 189 segundos, a equipe de segurança de voo comandou a tele-destruição ao notar que a trajetória seguida pelo conjunto e, conseqüentemente, seu ponto de impacto, estava se deslocando para fora da área de segurança.

Em 22 de agosto de 2003, o terceiro protótipo do VLS-1, o V03, foi destruído por um incêndio, durante os preparativos para seu lançamento. Foi criada uma Comissão Técnica de Investigação que, após 172 dias, concluiu um Relatório de Investigação (SERVIÇO PÚBLICO FEDERAL, 2004) detalhado sobre as possíveis causas do acidente. Possivelmente uma descarga de corrente elétrica acionou o motor A, do primeiro estágio do foguete, ocasionando uma forte explosão e destruindo todo o veículo, a torre móvel de integração e vitimando fatalmente 21 técnicos do Instituto de Aeronáutica e Espaço – IAE. No capítulo cinco deste relatório de investigação, são feitas recomendações para o chamado Retorno Ao Voo (RAV), propondo mudanças consideradas essenciais, principalmente nas questões de segurança. O relatório sugere o fortalecimento do programa e, conseqüentemente, faz recomendações para seu sucesso, para cumprir assim seus objetivos e as expectativas da sociedade brasileira, além de uma homenagem aos 21 profissionais que perderam a vida neste acidente.

Em 2005, a AEB e o CTA (atualmente denominado Comando-Geral de Tecnologia Aeroespacial) anunciaram o Programa Cruzeiro do Sul (em referência às cinco estrelas da constelação Cruzeiro do Sul). A nova família de lançadores, Figura 2.5, é composta pelos veículos Alfa, Beta, Gama, Delta e Épsilon, que serão construídos com o intuito de atender tanto às missões espaciais propostas no PNAE (Programa Nacional de Atividades Espaciais), como a possíveis acordos internacionais para missões de lançamento.

O programa prevê uma evolução gradativa dos veículos com relação a capacidades de transporte de carga útil e terá como maior desafio o desenvolvimento e fabricação de motores à propulsão líquida de médio e grande porte. O primeiro veículo do programa, o VLS Alfa é um lançador de três estágios e é baseado no VLS-1. Ele utiliza o 1º e o 2º estágios do VLS-1 e incorpora um novo propulsor a propelente líquido em substituição aos 3º e 4º estágios do VLS-1.

Figura 2.6 – Família de lançadores do programa Cruzeiro do Sul.

Fonte: (MORAES et al., 2006).

O programa tem previsão de lançamento do veículo de maior porte (VLS Épsilon) em 2022, ano do Bicentenário da Independência. O programa possibilitará ao Brasil a independência no transporte espacial de satélites de pequeno a grande porte (VILLAS BOAS, 2006) (MORAES; et. al., 2006).

2.3.4 Software em foguetes espaciais no Brasil

As primeiras iniciativas com relação ao desenvolvimento de sistemas computacionais em veículos espaciais no Brasil, nos projetos do IAE ocorreram nos anos 80, com o objetivo de desenvolver e testar um sistema de referência inercial

em foguetes de sondagem, que viesse a atender às necessidades de controle do futuro veículo lançador de satélites de pequeno porte (MOURA; DOMICIANO; FERNANDES, 1998). Esses foguetes de sondagem, chamados Sonda, não possuíam controle ativo da trajetória, ou seja, sua trajetória não era controlada (balística) e as pré-condições para o voo-nominal eram fornecidas pela orientação da plataforma de lançamento. A evolução da complexidade se deu na série 4 deste veículo, com a incorporação de um computador com software embarcado, com a capacidade de pilotar o engenho, ou seja, manobrá-lo através de mudança de orientação de seu corpo ao redor do próprio centro de gravidade.

Já o projeto do VLS significou uma grande evolução em termos de capacidade de desenvolvimento de software para aplicações espaciais no Brasil. O software embarcado deveria atender aos inúmeros requisitos do sistema, como interpretar e executar comandos emitidos pelo Banco de Controle na fase de pré-voo, executar os algoritmos de navegação e controle e comandar a sequência de eventos de voo, desde a decolagem até a separação do quarto estágio. Além destes, tinha também como requisito formatar e fornecer dados de telemédidas referentes às funções desempenhadas em voo. Para atender às características deste tipo de aplicação de missão crítica, de tempo real e de ambiente adverso, foi adotado, em 1994, um padrão de desenvolvimento de software do Departamento de Defesa norte-americano, a Norma DOD-MIL-STD-2167A (1988). Esse padrão serviu como um guia para delineamento e registro das atividades de requisitos e as demais atividades a serem realizadas, estabelecendo critérios a serem aplicados durante o desenvolvimento, a aquisição e o suporte do sistema de software. Nos três protótipos fabricados, o software embarcado obedeceu aos procedimentos de desenvolvimento e manutenção de software propostos em seu projeto e, até onde pode ser observado, com relativo sucesso, mesmo sem uma abordagem metodológica para a segurança, especificamente para este item do projeto.

O padrão 2167A, cancelado em 1988, foi substituído pelo padrão militar americano DOD-MIL-STD-498 (1994), posteriormente cancelado em 1998. Atualmente os projetos de software da NASA adotam, como referência, o modelo de ciclo de vida de software proposto pelo padrão da ISO/IEC-12207 (2001). Neste sentido, o uso de padrões internacionais que indicam práticas maduras de engenharia de software, pode facilitar o desenvolvimento de processos relativamente difíceis e caros, quando

implantados a partir de iniciativas isoladas. Os projetos de software do IAE seguem esta tendência de uso do modelo de referência de ciclo de vida de software adotado tanto pela ESA como pela NASA para as futuras versões de seus projetos de software. Estes padrões reúnem um conjunto de convenções, requisitos técnicos ou práticas mais estáveis que são muito úteis como referência, principalmente quando se trata de tópicos controversos. Além dos benefícios citados, a adoção de um padrão reconhecido, visa facilitar também a obtenção de uma futura certificação de software, na busca de excelência dos produtos gerados para as atividades de espaço.

2.4 CONSIDERAÇÕES

Este capítulo apresentou alguns dos principais conceitos relacionados ao processo ELICERE, tanto das disciplinas de engenharia de requisitos, como da engenharia de segurança aplicadas ao domínio de sistemas críticos. O propósito foi de introduzir a teoria e os temas mais relevantes que foram aplicados durante o estudo da tese e possibilitar um entendimento comum do vocabulário adotado. Os conceitos sobre engenharia espacial abordados nesta tese estão ligados aos aspectos mais genéricos dos elementos que compõem um sistema espacial, em particular o de um veículo espacial. Apresentam apenas uma conceituação básica sobre as duas engenharias e um breve histórico do desenvolvimento de projeto de foguetes no Brasil. Qualquer tentativa de aprofundamento no assunto exigiria outras explicações e informações que estariam fora do escopo do trabalho.

3 ELICERE: O PROCESSO DE ELICITAÇÃO DE METAS DE DEPENDABILIDADE APLICADO A SISTEMAS CRÍTICOS

O processo ELICERE foi idealizado para emprego no desenvolvimento de projetos críticos, especificamente nas atividades de elicitação de metas relacionadas a requisitos não funcionais como segurança, confiabilidade entre outros fatores de dependabilidade. O processo deve ser aplicado nas fases iniciais de um projeto, quando são feitos o levantamento de informações sobre o sistema e sua missão, a identificação das necessidades dos usuários, das regulamentações e padrões a serem adotados e o levantamento de outras informações relacionadas à concepção e às características do sistema a ser construído.

Nestas fases iniciais de desenvolvimento de um projeto, ainda estão sendo definidos e descritos os objetivos ou metas que o sistema proposto deve atender, bem como os equipamentos que serão utilizados no sistema. Geralmente, as metas e os equipamentos descritos estão muito focalizados nas funcionalidades que o sistema deve atender, isto é, o que o sistema deve fazer, e não o que o sistema não deve fazer, ou como o sistema deve se comportar para alcançar um determinado grau de confiança no seu funcionamento, ou seja, de dependabilidade. É justamente neste contexto que o processo ELICERE se propõe a atuar. Criando um conjunto de atividades a serem realizadas na fase de Análise de Missão, conduzem à discussão e a análise das funcionalidades e objetivos de um sistema crítico, o processo permite identificar fragilidades ou necessidades relacionadas à dependabilidade que devem ser observadas no início do projeto.

O processo ELICERE se originou da necessidade da definição e melhoria dos processos de engenharia de requisitos e de segurança para o projeto do Veículo Lançador de Satélites (VLS) do Instituto de Aeronáutica e Espaço (IAE). Durante a condução do projeto observou-se que o mesmo poderia ser utilizado de forma mais abrangente, para outros tipos de sistemas críticos. O processo tem atenção especial voltada para o sistema computacional de missão crítica, que geralmente tem diversas atribuições como o controle, a monitoração e a ativação de eventos. Este

tipo de sistema demanda a produção de requisitos dentro de critérios rigorosos de garantia da qualidade, de forma completa, e sem ambigüidades, capazes de garantir o sucesso da missão já na primeira versão do produto. Dentro da visão voltada principalmente à melhoria da confiança no funcionamento do sistema, a garantia da qualidade deve ser baseada na confiança depositada no serviço a ser fornecido pelo sistema (DE LEMOS; VERÍSSIMO, 1989).

A proposta é estimular, o mais cedo possível, a análise das metas, recursos e tarefas a serem utilizados no desenvolvimento de um sistema, de modo que sejam discutidas e propostas novas metas relacionadas a requisitos não funcionais, que colaborem na proposta de soluções e alternativas para alcançar o sucesso da missão com segurança e manter a integridade do sistema.

O processo busca ainda, através da construção de passos simples e objetivos, auxiliar na integração das equipes participantes do desenvolvimento do sistema, quer sejam engenheiros de requisitos de sistema ou engenheiros de requisitos de software. Melhorar a comunicação entre estas equipes, no que tange à produção de modelos e argumentos para a discussão das fragilidades e das condições inseguras do sistema, ajuda na definição das metas de dependabilidade relacionadas à segurança, capacidade de sobrevivência, confiabilidade e adequação do sistema, entre outras.

As metas a serem propostas poderão ser tanto metas de produto como metas de processo. As metas de produto deverão ser aquelas focadas principalmente em atributos de dependabilidade, como segurança, robustez e confiabilidade. Poderão ser também direcionadas às questões de qualidade em geral, como atributos de correção, precisão e portabilidade. As metas de processo deverão ser relacionadas às questões de melhoria do processo de desenvolvimento, de verificação e validação do produto e de práticas seguras de desenvolvimento.

Neste capítulo são apresentadas as atividades do processo ELICERE e detalhes sobre como ele deve ser conduzido. Também, no intuito de explicar como o processo deve ser utilizado, são apresentados alguns exemplos ilustrativos de sistemas computacionais da área espacial.

3.1 MODELO DO PROCESSO

Os passos do processo ELICERE de elicitação de requisitos de dependabilidade podem ser resumidos conforme mostra a Figura 3.1. Estes passos são detalhados nas seções seguintes deste capítulo e compreendem as seguintes macro-atividades:

- Modelar o sistema com i^* ;
- Aplicar análise de perigos.

Figura 3.1 – Processo ELICERE de definição de metas de dependabilidade.

Modelar o sistema com i^* : nesta primeira macro-atividade do processo é criada uma representação inicial do sistema, através do uso dos modelos propostos pela técnica i^* . Através dos modelos i^* e de seus componentes meta, recurso, tarefa e meta-soft, é possível detalhar os objetivos da missão, os recursos, as tarefas e, caso já existam, as metas de qualidade do sistema. Estes componentes de representação da técnica de modelagem i^* serão utilizados posteriormente em uma análise dos

prováveis e potenciais riscos que o sistema a ser construído poderá correr. Para esta atividade, serão utilizados os diagramas i* de representação de Dependências Estratégicas – SD (na sigla em inglês, *Strategic Dependencies*) e o de Razões Estratégicas – SR (na sigla em inglês, *Strategic Rationales*).

Aplicar análise de perigos: nesta macro-atividade são primeiramente aplicadas palavras-guia nos componentes do sistema SD e SR, utilizando-se o recurso de palavras-guia baseadas principalmente na técnica HAZOP e FMEA para a identificação dos potenciais perigos do sistema. Esta atividade do processo permitirá obter uma importante caracterização dos principais pontos a serem explorados em termos de melhoria da dependabilidade do sistema. Subseqüentemente à aplicação das palavras-guia, um questionário deverá ser respondido, que resultará na proposta de um conjunto de metas relacionadas à dependabilidade.

O desdobramento natural da aplicação do processo será um conjunto de recomendações, em forma de metas-soft, que auxiliarão as equipes de desenvolvimento do sistema na definição de requisitos não funcionais, de forma a minimizar os perigos e riscos identificados.

3.2 SEQUÊNCIA DAS ATIVIDADES DO PROCESSO

O processo ELICERE tem como objetivo fornecer um conjunto de atividades da garantia de produto para sistemas críticos, que atendam às suas exigências com relação às questões de dependabilidade, como segurança, confiabilidade e disponibilidade. A linguagem de representação de processo SPEM (da sigla em inglês, *Software Process Engineering Metamodel*) (SPEM, 2002) já adotada para o desenho de processos de garantia da qualidade de software (LAHOZ, 2004), foi escolhida por apresentar uma visão dinâmica das atividades do processo, isto é, uma representação do processo com relação à seqüência dos passos a serem executados.

Apesar de ser usado para a modelagem de processos de software, como a linguagem de meta-modelos da UML (da sigla em inglês, *Unified Modeling Language*), o SPEM é uma linguagem de fácil representação e possui um conjunto mínimo de recursos de modelagem necessários para descrever um processo de desenvolvimento. O SPEM foi criado pela OMG (da sigla em inglês, *Object Management Group*), organização aberta e sem fins lucrativos que se propõe a integrar vários padrões e tecnologias relacionadas a sistemas de tempo real, sistemas especialistas, sistemas embarcados, entre outros, em vários ramos do conhecimento, incluindo o de modelagem de negócios, governança, saúde, robótica e espaço (OMG, 2008).

O SPEM, além de incorporar os ícones padrões da UML, criou representações para as outras entidades de um processo: atividades, produtos de trabalho, ferramentas de apoio, etc. A linguagem SPEM auxiliou na representação da dinâmica do processo ELICERE de elicitação, no entendimento dos passos a serem dados e nos produtos a serem utilizados durante a execução do processo. A Figura 3.2 apresenta a seqüência de atividades do processo ELICERE, seus produtos de trabalho de entrada e saída, bem como o responsável pela atividade dentro do processo.

Apresentação sucinta das atividades do processo ELICERE:

1) Coletar informações: a primeira atividade, a cargo do engenheiro de requisitos, é identificar informações a respeito do sistema. Devem ser utilizadas técnicas convencionais e úteis de elicitação de requisitos, como análise dos envolvidos, entrevistas, observação e estudo da documentação, como descritas por Lauesen (2002). A técnica de análise dos envolvidos procura responder quem são os envolvidos, seus objetivos, contribuições e riscos na definição e uso do sistema a ser construído. A técnica de entrevistas é utilizada para se adquirir conhecimento do domínio do problema, através da extração de idéias e informações sobre o que é o sistema e da identificação dos conflitos entre os interessados. Emprega-se a observação para a análise dos usuários em seu ambiente de trabalho, de suas atividades diárias e dos usuários em seu ambiente de trabalho, de suas atividades diárias e dos processos de trabalho que utilizam. O estudo da documentação é um modo de verificação cruzada da informação extraída da técnica de entrevista. É uma

forma rápida de se obter informação de base de dados de projeto. O analista estuda os documentos como formulários, arquivos, “logs” e outras documentações do sistema pré-existente.

Figura 3.2 – Seqüência de atividades do processo ELICERE.

2) Modelar Sistema: a partir das informações coletadas, são identificados os atores, as metas, os recursos, as tarefas e, se possível, as metas-soft pré-definidas para o sistema a ser construído. Devem então, ser modelados os relacionamentos dos atores com as metas. A caracterização das metas, ou objetivos de missão, são os itens que primeiramente são definidos no planejamento de um sistema. Sua identificação, inclusive, ajudará na definição dos recursos e das tarefas relacionadas ao atendimento destas metas. Logo após, são identificados e modelados os recursos do sistema, visto que estes são os componentes que permitirão viabilizar a obtenção das metas. Finalmente, são detalhadas as tarefas, que efetivamente permitirão que os recursos possam ser utilizados para atender as metas estabelecidas. As tarefas podem ser de todos os tipos, desde acionamento de *jumpers* no sistema de acionamento de eventos, até a programação de extensos algoritmos de controle. Dentro desta abordagem para a construção dos modelos i^* , devem ser elaborados os modelos SD, onde são representados primeiramente os relacionamentos entre os atores e suas metas. Em seguida, é feito um diagrama para representar os relacionamentos entre atores e os recursos envolvidos para atingir as metas. Segue-se o desenho dos modelos SR, que descrevem o comportamento interno de cada ator, focando principalmente nas tarefas necessárias para atingir as metas e identificando novamente os recursos relacionados e os atores externos envolvidos. Especificamente para as tarefas, devem ser criados tantos modelos SR quantos forem necessários para representar cada conjunto de atividades específicas de forma clara e compreensível, e de maneira a representar o comportamento detalhado de cada ator.

3) Aplicar palavras-guia: aplicar aos componentes i^* , meta, recurso, tarefa e meta-soft de cada modelo, as palavras-guia do processo ELICERE. No processo, para cada componente da técnica i^* , existe um conjunto específico de palavras-guia e uma interpretação de apoio, para auxiliar a condução desta atividade pelo engenheiro de requisitos. Aplicar palavras-guia significa basicamente verificar qual seria seu efeito sobre o componente i^* analisado, em termos de desvio de intenção de projeto. Poderia ser a discussão, por exemplo, de qual seria o perigo decorrente da ausência de um recurso do sistema que está sendo analisado. As palavras-guia devem ser aplicadas, de preferência, primeiro para metas, depois para recursos, em seguida para tarefas. Acredita-se que esta ordem promoverá um entendimento

gradual das principais relações de dependência do sistema. Esta ordenação leva em conta que as metas, ou objetivos de missão, são os itens primeiramente definidos no planejamento de um sistema. Ela permitiria, inclusive, a identificação dos objetivos mais críticos a serem eventualmente priorizados e quais os recursos e tarefas relacionadas que deverão sofrer prioritariamente a aplicação de palavras-guia. Logo após, são aplicadas as palavras-guia aos recursos do sistema, visto que estes são os componentes que permitirão viabilizar a obtenção das metas. Em seguida, são aplicadas as palavras-guia às tarefas, que efetivamente permitirão que os recursos possam atender às metas estabelecidas.

4) Aplicar questionário: o questionário deve ser aplicado logo após a interpretação das palavras-guia, para cada componente do modelo do sistema estudado. O questionário tem o objetivo de levar à identificação dos modos de falha, do perigo esperado, e da severidade da falha do componente estudado, em relação à palavra-guia. O questionário visa também esclarecer os prováveis tipos de falha, quer sejam de hardware, software ou humana.

5) Identificar metas-soft: última etapa do processo busca definir quais são as metas não funcionais, qualitativas e quantitativas, necessárias para minimizar potenciais riscos identificados durante a aplicação do processo ELICERE nos componentes do sistema, de modo a melhorar a sua confiança no funcionamento. O produto resultante do processo são as metas-soft identificadas, e sua caracterização devem levar em conta um conjunto de fatores de dependabilidade de referência, que auxiliarão na condução da escolha das metas-soft mais adequadas.

O ELICERE então fornece um processo organizacional que auxilia as atividades da garantia de produto e de processo, no que tange às questões da dependabilidade do sistema, capaz de atender às exigências estratégicas e tecnológicas de um mercado em constante evolução, podendo inclusive, levar a organização que o empregue a alcançar níveis mais elevados de maturidade e capacitação (LAHOZ, 2004).

Uma pré-condição para o desempenho das atividades de modelagem, análise das palavras-guia e aplicação do questionário é a capacitação e experiência do engenheiro de requisitos na área do domínio da aplicação. Além de possuir um

mínimo de conhecimento na área, o engenheiro de requisitos precisará da assessoria de especialistas das diversas áreas do conhecimento da aplicação, para ser capaz de identificar, entre os vários componentes representados no modelo, quais são os mais prioritários para a análise de palavras-guia e para a aplicação do questionário. Deverá também receber orientação de especialistas para avaliar as metas de dependabilidade definidas, a fim de adequá-las às restrições de projeto e a viabilidade técnica de realizá-las.

A proposta de se criar diversos modelos ao invés de um único diagrama contendo todas as visões do sistema é recomendado, devido ao fato de que um único modelo ficaria extenso e complexo dificultando seu entendimento e a condução da aplicação das palavras-guia. Recomenda-se que na aplicação do processo ELICERE se faça um diagrama separadamente dos atores e suas metas, atores e seus recursos, e atores e suas tarefas. Se necessário, deve-se acrescentar a este conjunto de diagramas combinações dos componentes i* como, por exemplo, atores com metas e recursos ou atores com metas e tarefas, ou atores recursos e tarefas, atores com metas e metas-soft, entre outros.

3.3 APLICAÇÃO DO I* NO PROCESSO ELICERE

Subsistemas, processos, atividades, pessoas, estrutura organizacional e restrições de toda natureza devem ser mapeados de forma a identificar todas as entidades que direta ou indiretamente afetam e interagem com o sistema. O uso da técnica de modelagem i* tem o objetivo de representar quais são os atores envolvidos, quais são as metas a cumprir, os recursos associados e as tarefas a serem realizadas por cada ator. Muito comumente, estas informações podem ser encontradas a partir de dados de alto nível do sistema, como os contidos no documento de especificação de requisitos de sistema, ou mesmo das informações obtidas durante a fase de análise de viabilidade do projeto, onde são discutidas as principais soluções técnicas e caracterizados os elementos críticos do projeto.

O emprego da técnica i^* para o processo ELICERE abrange a utilização do modelo de Dependências Estratégicas (SD) e do modelo de Razões Estratégicas (SR). O modelo de Dependências Estratégicas é usado para descrever as relações de dependências externas entre os participantes do sistema, os chamados atores. Atores, neste contexto, são representados por entidades concretas ou abstratas. Podem ser entidades concretas, como seres humanos desempenhando papéis ou, ainda, sistemas eletrônicos ou computacionais, ou entidades abstratas, representando organizações, agentes autônomos inteligentes e outras entidades não tangíveis. O modelo de Razões Estratégicas é usado para descrever o comportamento interno dos atores do sistema, seja através da descrição dos interesses ou das atividades que eles realizarão, seja para representar e avaliar as possíveis razões e alternativas de projeto. No domínio das aplicações críticas, onde o grau de automação e complexidade é geralmente significativo, a representação de atores muitas vezes é feita através de subsistemas que assumem as mais variadas atribuições e funções. Por exemplo, para o modelo de Dependências Estratégicas de um sistema computacional de bordo de um veículo lançador de satélites, é possível identificar vários atores como controlador digital embarcado, sistema de apoio em solo, sensor inercial, telemetria, atuadores, cabos umbilicais, plataforma de lançamento, suprimento de energia, sistema de segurança, entre outros.

Sistemas podem ainda ser subdivididos em subsistemas, como no caso do exemplo anterior, o controlador digital, que pode ainda ser subdividido em subsistemas de hardware (computador de bordo, controladoras, cabos e conexões) e subsistemas de software (software aplicativo de bordo, dispositivo inicializador do computador, protocolos de comunicação).

Atores podem também representar agentes humanos, como o caso de operadores do sistema de apoio em solo, de segurança, de telemetria, entre outros. Cada um deles possui um conjunto de tarefas, recursos internos, metas e metas-soft que devem ser analisados para avaliar sua possível interação com o controlador digital para descrição e posterior análise para a avaliação da dependabilidade.

A primeira atividade do processo ELICERE é coletar informações para auxiliar na criação dos modelos do sistema a ser construído. A coleta de informações deve

permitir a representação do sistema em um alto nível de abstração, identificando o ambiente de atuação do sistema a ser construído, os recursos e as tarefas necessárias para a condução do sistema.

O modelo SD de metas contém o relacionamento entre atores e metas (metas inter-atores) e representa as metas estratégicas definidas pelo cliente, ainda em termos de objetivo de missão. Este modelo cria uma visão de alto nível do sistema, mostrando os objetivos a serem alcançados pelo projeto, equivalente, por exemplo, à definição em alto nível das metas de negócio do sistema a ser construído (BASILI, 2007).

Em seguida é criado um modelo SD para a representação do relacionamento entre os atores e os recursos envolvidos no sistema a ser construído (recursos inter-atores). Os recursos devem estar sempre que possível, relacionados a uma meta a ser cumprida – o meio para se atingir um fim, apresentada no modelo SD de metas. Através deste novo modelo será possível identificar e representar tanto os dispositivos físicos (hardware) envolvidos, como as informações necessárias para o atendimento às metas associadas aos recursos.

Nos modelos SD desenvolvidos, são representadas também as tarefas e as metas-soft externas aos atores, caso seja necessário. O importante é buscar a representação de todos os componentes do sistema identificados durante o levantamento inicial de informações do sistema para a posterior análise de palavras-guia.

Logo após, são criados modelos SR tanto quanto forem necessários, para identificar as tarefas a serem desempenhadas pelos atores (tarefas intra-atores). As tarefas intra-atores descrevem o que o ator precisa realizar para atender uma ou mais metas do sistema. A utilização deste tipo de modelo tem como objetivo representar as funções tanto lógicas como de atuação física do sistema. São representados neste diagrama rotinas ou programas de software, comandos de atuação física dos componentes eletro-eletrônicos, entre outras funcionalidades.

As palavras-guia e o questionário aplicados para os modelos identificam metas de dependabilidade relacionadas tanto às questões de qualidade de software (por exemplo, correção, disponibilidade operacional, desempenho, previsibilidade, estabilidade), hardware (por exemplo, confiabilidade e disponibilidade), como de processo (por exemplo, métodos empregados, normas e padrões, processos de verificação e validação). As metas-soft também poderão conduzir na busca de melhores alternativas ou razões de projeto e implementação, que terão efeito nas atividades de engenharia do sistema.

Além das metas de dependabilidade, a análise dos modelos servirá tanto para identificar e minimizar conflitos entre metas, definir os limites onde o sistema deve atuar, bem como consolidar quais são os atores responsáveis por atingir determinadas metas.

Com estas três visões, metas e recursos inter-atores e tarefas intra-atores, se permitirá, de forma mais sistematizada, a identificação dos elementos mais críticos do sistema a serem analisados pelas palavras-guia e conseqüentemente a priorização das melhorias a serem propostas após a aplicação do questionário.

3.3.1 Modelo SD para o processo ELICERE

O modelo de Dependências Estratégicas (SD) representa um ator ou um conjunto de atores que possui algum tipo de ligação de dependência entre si. Um ator pode ser identificado na técnica HAZOP, como um nó de estudo e suas relações de dependência como ligações. Através do modelo SD será possível identificar e analisar as interfaces entre os atores, de modo a permitir observar relações críticas ou mesmo vulneráveis que devem ser tratadas no processo de elicitação de metas de dependabilidade. Em um caso específico, como o do exemplo de um veículo lançador de satélites, os principais nós ou entidades, que podem ser representadas através da técnica de modelagem *i**, são os sistemas Solo, Carga Útil e Veículo.

Para se ter uma visão mais detalhada dos modelos, deve-se observar cada sistema isoladamente. Ainda, em nível mais detalhado, cada subsistema extraído do sistema Solo, por exemplo, poderia ser representado como um sistema autônomo, dependendo do ângulo de visão e da análise que se pretende ter de uma determinada relação de dependência. O que pode auxiliar, segundo a visão de dependabilidade, é não perder de vista o fato de que sempre existirá alguma meta que o faz dependente ou que satisfaz a dependência de outro sistema. Dessa forma se obtém uma visão de alto nível das relações de dependência entre os atores dentro do sistema proposto.

3.3.1.1 Tipos de dependências do modelo SD para o processo ELICERE

Segundo o conceito proposto pela técnica i^* e utilizado para o processo ELICERE, atores podem ser representados como subsistemas ou entidades que se relacionam entre si, participantes do sistema, assumindo a figura de papéis ou agentes, podendo ter um comportamento controlável ou não. Possuem algum tipo de relação de dependência para alcançar a realização dos objetivos ou metas de sistema. Individualmente, os atores podem até atender seus objetivos imediatos ou primários, mas no contexto do sistema como um todo, eles possuem uma relação de dependência (*dependum*) com outros atores que faz com que se tornem ora fornecedores (*dependee*), ora solicitantes (*dependor*) de serviços de outro ator. Metas, tarefas e recursos são necessários para alcançar os objetivos do sistema. Metas-soft também poderão estar definidas a priori, tanto para atender aspectos técnicos, como qualitativos e gerenciais do sistema.

Na dependência de meta, um ator (ou subsistema) depende de outro para que um determinado objetivo ou intenção seja satisfeita. Este tipo de representação de vínculo ou dependência entre subsistemas pode ser expresso em diferentes níveis de abstração. Podem-se abranger objetivos de alto nível, como metas estratégicas ou de missão, e de baixo nível, como atividades técnicas ou funcionais de um ator com relação ao outro ou ao sistema como um todo. As metas da dependência SD diferem das metas SR que representam a descrição do comportamento interno do

ator ou subsistema. As metas estratégicas ou de missão devem descrever o comportamento entre atores com respeito ao cumprimento dos objetivos da missão ou dos objetivos técnicos de uma determinada tarefa atribuída ao ator.

Como exemplo de meta de missão, a Figura 3.3 mostra a meta “comandar eventos”, entre o Controlador Digital (CD) e o sistema de Segurança e Atuação (SA), a meta “fornecer telemédidas” entre o CD e o sistema de Telemetria Embarcada (TE), e a meta “medir tempo durante o voo” do próprio CD.

Figura 3.3 – Metas de missão do CD com o próprio CD, com o SA e com o TE.

O ator TE depende do ator CD para realizar a meta “fornecer telemédidas”. A direção do “D” na linha de conexão entre os atores determina quem é o ator responsável por satisfazer a relação de dependência, ou seja, o ator dependedor (*dependee*) CD. O mesmo acontece com o ator SA, que depende do ator CD para atingir a meta “comandar eventos”. Finalmente, como a meta (*dependum*) “medir tempo durante o voo”, possui um tipo de relação de dependência de um lado só, não existe o ator dependente (*dependor*). Esta meta está somente relacionada com o próprio ator dependedor.

A dependência recurso é representada pelos recursos físicos ou lógicos dos atores envolvidos. Recursos físicos, como interfaces de comunicação entre os atores, e recursos lógicos, como dados ou eventos que transitam entre os atores, estão vinculados a metas e tarefas a serem cumpridas. Se possível, deve-se identificar se é crítico para o sistema o fornecimento de tal recurso, isto é, se a não satisfação desta relação de dependência de recurso entre dois atores provocará a perda da missão, por exemplo. Com isto, na próxima fase da aplicação do processo ELICERE será possível indicar quais componentes do sistema são mais prioritários para a aplicação das metas de dependabilidade que poderão ajudar a garantir a satisfação desta relação de dependência.

Como mostrado na Figura 3.4, existe uma relação de dependência entre o ator CD e o ator sistema de Apoio em Solo do veículo (AS), durante a fase de pré-vôo, que é satisfeita através dos recursos “comando preparação SI” (comando de preparação do Sistema Inercial), “hardware OK”, “testes_entradas e testes_saída”, “resposta_teste”, “perfil de vôo”, “Navegação OK”, e “interface_serial”.

Figura 3.4 – Recursos utilizados para a comunicação do CD com o AS.

A dependência tarefa pode ser representada pelas inúmeras atividades que um ator deve realizar para alcançar uma determinada meta. A descrição de uma tarefa nesta fase inicial de projeto pode não estar precisa ou ainda ser incompleta. O mais importante é, durante o processo de elicitação proposto, conseguir identificá-las mesmo ainda com um alto grau de subjetividade para poder discutir, em seguida, os possíveis riscos que levem a tarefa ou o sistema a um não funcionamento, por exemplo. Somente a partir daí será possível identificar metas-soft para melhoria do nível de dependabilidade deste sistema.

No caso específico da relação de dependência entre o sistema Sensor Inercial (SI) e o CD, para a satisfação da meta “iniciar sensor inercial”, são identificadas as seguintes tarefas: “realizar alinhamento inicial”, “calibrar SI”, “estabilizar SI”, e “realizar o alinhamento de precisão”, como representado na Figura 3.5.

Figura 3.5 – Relação de dependência de tarefas entre os atores SI e CD.

Na dependência de meta-soft, um sistema depende de outro para que um determinado objetivo não funcional seja satisfeito ou de forma a se atingir metas qualitativas do sistema. Uma meta-soft é similar a uma meta, a não ser pelo fato de que os critérios de sucesso podem não ser estabelecidos a priori. Podem ser

elaborados em termos de métodos e estratégias qualitativas a serem adotadas a fim de perseguir um objetivo.

As metas-soft “criar interface visual amigável”, “consistir comandos”, “manter ações terminais distantes” e “responder tempo máximo 3 seg”, apresentadas na Figura 3.6, descrevem objetivos não funcionais da relação de dependência entre o sistema de Apoio em Solo (AS) e o Operador do sistema de Apoio em Solo (Op_AS), de forma a garantir que a meta “receber comandos Op_AS” (meta a ser cumprida pelo Op_AS) e a meta “receber resposta AS” (meta a ser cumprida pelo AS) existentes entre estes dois atores, seja satisfeita de forma adequada.

Figura 3.6 – Relação de dependência de meta-soft entre os atores AS e Op_AS.

3.3.1.2 Relações de dependência do modelo SD

A técnica i* oferece vários tipos de relações de dependência entre os atores do modelo SD. O processo ELICERE utiliza principalmente as dependências

compromissadas e as críticas. As dependências comprometidas são aquelas em que o dependente é afetado pela falha na relação de dependência, mas não significativamente. Isto significa que o objeto da relação de dependência (*dependum*) entre dois atores não foi afetado de modo irreversível, permitindo ainda a continuidade do relacionamento ou do funcionamento do sistema. No exemplo da Figura 3.7, o ator CD tem como meta “fornecer telemédidas” para o ator sistema de Telemetria Embarcada (TE), que os enviará ao sistema de Telemetria em Solo (TS) para atender a meta “fornecer dados análise pós-vôo”.

Figura 3.7 – Relação de dependência comprometida entre os atores CD, TE e TS.

Caso o ator dependente TS não tenha a sua relação de dependência completamente satisfeita, isto é, se houver perda parcial destas informações transmitidas pelo TE, isto não afetará irreversivelmente a relação entre os atores e o não atendimento da meta. Dependendo do percentual de perda das informações enviadas ao TE, o perfil real do voo realizado poderá ser recomposto.

As dependências críticas são aquelas em que o dependente é significativamente afetado com a falha na relação de dependência. É representado pela inclusão de um “X” no lado onde o ator dependedor, significando que a meta, recurso ou tarefa

associada à relação entre os atores, foi afetada de modo irreversível ou de modo que não permita a continuidade do relacionamento ou do funcionamento do sistema.

A Figura 3.8 ilustra a dependência crítica relacionada ao recurso “dados segurança vôo” entre os atores sistema de Telemetria a Bordo (TE) e o ator sistema de Segurança em Solo (SS) para atender à meta “fornecer dados segurança de vôo”. Sem a informação proveniente do TE o SS não obtém os dados necessários para auxiliar na tomada de decisão de tele-destruição pelo Operador do sistema de Segurança em Solo (Op_SS). Por outro lado o CD necessita do recurso (crítico) telecomando, a ser enviado pelo SS para destruir o veículo em caso de violação dos requisitos de segurança de missão.

Figura 3.8 – Relação de dependência crítica entre os atores SS, TE e CD.

3.3.2 Modelo SR para o processo ELICERE

O modelo de Razões Estratégicas (SR) descreve o comportamento interno de um ator ou subsistema, apresentando também possíveis alternativas para atingir suas metas, os recursos internos que o ator vai utilizar, as tarefas e subtarefas

necessárias para obtenção da meta a ser alcançada e as meta-soft relacionadas ao ator interno. Na visão deste trabalho, o modelo SR permite, através da investigação mais aprofundada das relações internas do subsistema ator, identificar mais facilmente os componentes sujeitos aos perigos ou riscos potenciais, que possam conduzir o subsistema a um estado de falta de confiança no funcionamento. A Figura 3.9 mostra um exemplo de representação das tarefas internas do ator CD com relação à meta “apontar veículo”.

Figura 3.9 – Tarefas internas do ator CD com relação à meta “apontar veículo”.

Para atingir tal objetivo o ator CD precisa realizar um conjunto de tarefas como “iniciar apontamento”, definindo o instante que se dará o cálculo do apontamento e as condições iniciais para operação desta manobra do voo do veículo lançador.

Em seguida, deve ser realizada a tarefa “calcular parâmetros orbitais”, isto é, calcular parâmetros circulares, como vetor raio, ângulo da trajetória, excentricidade, entre outros. Em seguida é necessário “definir plano de voo”, calculado através dos dados relativos à posição e velocidade. A tarefa “analisar modo falha” identificará se o raio impulsional de injeção é maior que o apogeu da órbita atual deste

determinado instante do voo, bem como o instante de ignição. Finalmente a tarefa “calcular ângulo de apontamento” para a obtenção dos ângulos de Euler em relação ao triedro de navegação.

É importante destacar também que no modelo SR pode-se observar, além do comportamento interno de um ator, sua interação com outros subsistemas. Muitas vezes é preciso representar a relação de dependência deste subsistema com outro subsistema, ou mesmo com todo o sistema, principalmente quando alguma tarefa do modelo SR depende de algum recurso externo ao seu controle.

A Figura 3.10 mostra um exemplo de como alguns dos componentes do modelo SR do CD interagem entre si.

Figura 3.10 – O modelo SR do ator CD com relação à tarefa “fornecer dados telemetria”.

Os atores sistema de Telemetria em Solo (TS) e o ator sistema de Segurança em Solo (SS), precisam dos dados de vôo que o TE adquire do CD para envio a Terra. O recurso “dados vôo” é gerado pelo CD, através da tarefa “fornecer dados telemetria”, que possui duas subtarefas: “adquirir dados”, e “formatar dados”. A tarefa “adquirir dados” possui as subtarefas, “condicionar eventos”, “condicionar processamento de controle”, “condicionar entradas de controle” e “condicionar dados pressão”. O recurso “dados segurança vôo” é subproduto do recurso “dados vôo”.

3.3.2.1 Tipos de relacionamento do modelo SR

Sob a perspectiva do processo ELICERE e do uso da técnica i^* , o modelo SR representa como os atores (subsistemas) se comportam internamente, ou como é a interação destes com o resto do sistema, utilizando os mesmos componentes do modelo SD: meta, meta-soft, tarefa e recurso. Os principais tipos de relacionamentos que interconectam os nós propostos pelo i^* , *meio-fim* e *decomposição de tarefas*, são utilizados no processo ELICERE.

Os relacionamentos *meio-fim*, que indicam um relacionamento entre um fim – que pode ser representado por uma meta – e um meio para se atender este meta será adotado principalmente para a decomposição das metas em tarefas.

Como ilustração do relacionamento *meio-fim*, a Figura 3.11 mostra dois exemplos de metas-fim “analisar raio ignição impulsional” e “analisar instante impulso” do sistema veículo lançador. A primeira tem como relacionamento-meio as tarefas “tiro no apogeu” e “tiro normal”.

Uma destas alternativas deverá ser utilizada para se atingir a meta “analisar raio de ignição impulsional” a partir do recurso “raio de ignição”. A segunda meta-fim tem como relacionamento-meio as tarefas “modo APD” (o instante de ignição é estabelecido como sendo o do término da manobra de basculamento) e “modo APP” (recalcula o instante de ignição para a fase descendente da trajetória). Uma destas

duas tarefas deverá ser utilizada a partir da informação (recurso) do “instante ignição” do terceiro estágio do veículo.

Figura 3.11 – Metas meio da tarefa “analisar modo falha” e suas tarefas fim.

Os relacionamentos de *decomposição de tarefa* são utilizados para representar o que deve ser feito para que uma tarefa seja realizada. Eles podem também interligar-se aos relacionamentos de dependência do modelo de Dependências Estratégicas (SD), quando as razões ultrapassam os limites do respectivo subsistema ou ator.

Como exemplo de uma relação de *decomposição de tarefa*, a Figura 3.12 mostra a *decomposição da tarefa* “executar basculamento” em quatro tarefas: “parar o movimento angular do veículo”, “efetuar rotação em rolamento”, “efetuar rotação em arfagem” e “manter veículo em atitude até spin-up”. A tarefa “executar basculamento” necessita do recurso “ângulo rolamento FI” e “ângulo arfagem” para alcançar a meta “bascular veículo”.

Figura 3.12 – Decomposição de tarefa “executar basculamento”.

Após a modelagem do sistema em termos de modelo SD, para a obtenção da visão estratégica e de missão do sistema, e do modelo SR, para a obtenção da visão interna do comportamento de seus atores, o próximo passo é avaliar o grau de dependabilidade através do uso de palavras-guia adaptadas das técnicas de análise de perigos HAZOP e FMEA.

3.4 PALAVRAS-GUIA PARA O PROCESSO ELICERE

A aplicação de técnicas de análise de perigos baseadas em HAZOP e FMEA em componentes, funções ou fluxos relacionados a sistemas computacionais (hardware, software, ou firmware - software residente em memória não-volátil, como ROM, EPROM, ou memória flash), são fortemente recomendadas para aplicação em sistemas críticos da área aeroespacial e de defesa (NASA-GB-8719.13, 2005) (ECSS-Q-ST-30-02C, 2009) (MOD-DEF-STAN-058, 2000). Tanto a NASA como a ESA recomendam a utilização das técnicas FMEA e FMECA (do inglês, *Failure Modes and Effects Analysis* e *Failure Modes, Effects and Criticality Analysis*) para uso na análise do projeto funcional e físico de um sistema espacial. O Departamento de Defesa Britânico, além de estimular o uso de HAZOP para identificar potenciais perigos em um projeto, apresenta um guia de como aplicar esta técnica de forma eficiente e eficaz.

Variações das abordagens HAZOP e FMEA, como CHAZOP (no inglês, *Computer HAZOP*) (NIMMO; NUNNS; EDDERSHAW, 1987), SHAZOP (no inglês, *Software HAZOP*) (BURNS; PITLADO, 1993), SFMEA (no inglês, *Software Failure Modes and Effects Analysis*) (LUTZ; WOODHOUSE, 1996) e SHARD/LISA (no inglês, *Software Hazard Analysis e Resolution in Design/ Low-level Interaction Safety Analysis*) (PUMFREY, 2000), foram criadas como novas formas de condução das atividades da engenharia de segurança, utilizando o recurso de palavras-guia (no inglês, *guidewords*) para a análise e para a devida qualificação dos desvios de intenção de projeto e operação.

A aplicação de palavras-guia em análise de perigos de um sistema é feita através do uso de uma palavra ou frase que expresse ou defina um tipo específico de desvio de intenção de projeto. A identificação do desvio de intenção, segundo Redmill; Chudleigh e Catmur (1999) é baseada na experiência de interpretação de quem utiliza as palavras-guia pré-determinadas que focam nos problemas que possam ocorrer na planta analisada ou um possível desvio de projeto. É aplicada a um determinado atributo do sistema, investigando quais as conseqüências, por exemplo, se esse atributo fosse maior, ou faltasse, ou mesmo tivesse sido projetado errado.

Desta maneira, no processo ELICERE, o uso de palavras-guia na análise de perigos do sistema a ser construído, auxiliarão na identificação e priorização das fragilidades do sistema, e na definição de metas que venham a minimizar estes problemas.

O processo ELICERE apresenta um conjunto adequado de palavras-guia para aplicação na análise de perigos de sistemas críticos, auxiliando a identificação de questões de dependabilidade relevantes, e que devem ser avaliadas para a melhoria do sistema ainda na sua fase de concepção. As palavras-guia específicas para a elicitación de requisitos propostas pelo ELICERE foram destinadas para serem aplicadas principalmente aos componentes do sistema identificados como relação de dependência (*dependum*) na técnica de modelagem *i**, ou seja, aqueles componentes que procuram satisfazer uma relação de dependência. A aplicação das palavras-guia nos componentes *dependum* do modelo de um sistema crítico (meta, recurso, tarefa e meta-soft), se deve ao fato de que um dos maiores problemas da análise de perigos de sistemas complexos, é justamente a dificuldade de extrair o comportamento interativo dos seus componentes (LAHOZ et al., 2000). Aplicar as palavras-guia nestes componentes, também chamados pela técnica HAZOP de “nós de estudo”, permitem analisar as inter-relações de dependência do modelo e caracterizar os potenciais perigos envolvidos com os atores do sistema.

É importante ressaltar que no processo ELICERE, as palavras-guia são aplicadas primeiramente para meta, seguida de recurso e tarefa. Considerando que as palavras-guia de recurso e tarefa são necessárias para a obtenção das metas, infere-se que os desvios de intenção das metas podem ser provocados pelos desvios de intenção de recursos ou de tarefas. Uma representação semântica relacionando as palavras-guia de meta com as de recurso e tarefa é apresentada pelo processo, de forma a facilitar a identificação dos principais perigos observados na não obtenção de uma meta.

Para conduzir a análise das palavras-guia e a identificação dos seus inter-relacionamentos de meta, recurso e tarefa, foi desenvolvido um questionário que conduz e classifica as conseqüências advindas de um mau funcionamento do sistema, seja devido ao não atendimento de uma meta ou de um recurso, por exemplo. O questionário associado às palavras-guia ajuda a identificar os possíveis

desvios ou distúrbios de projeto e operação, classificando o impacto deste desvio no comportamento do sistema, seu grau de severidade e quais seriam as metas de dependabilidade necessárias para minimizar este desvio.

O questionário auxilia também na escolha de quais componentes do sistema (nós de estudo) e respectiva interpretação em termos das palavras-guias, devem ser analisados com mais profundidade pelo processo. A idéia é, através de parâmetros estabelecidos durante o início desta atividade do processo, selecionar os componentes do sistema com maior potencial de perigo para análise mais detalhada e extração das metas de dependabilidade, que permitirão o desdobramento em requisitos não funcionais.

As palavras-guia e o questionário aplicados aos componentes *dependum*, que são os componentes que interagem com os atores do sistema, permitem, em última estância, analisar o comportamento e as fragilidades de cada ator envolvido nesta relação, quer sejam daqueles que o sistema não tem a gerência da funcionalidade, como os produtos COTS (sigla do inglês, *Comercial of The Shelf*), quer sejam dos atores cujos componentes e funcionalidades são desenvolvidas internamente pela organização.

3.4.1 Adequação das palavras-guia para o processo ELICERE

A utilização de palavras-guia para o processo de elicitação de metas de dependabilidade está direcionada principalmente para o universo de situações de perigo em sistemas computacionais críticos. Isto inclui problemas relacionados ao funcionamento de dispositivos de hardware, de dispositivos de software e problemas de operação do sistema. Dentro deste universo, a escolha das palavras-guia foi fortemente dirigida à aplicação em sistemas que incluem dispositivos eletronicamente programáveis, os chamados PES (no inglês, *Programmable Eletronic System*).

Atualmente os PES são largamente empregados em sistemas críticos ou em sistemas que tem implicações rigorosas com relação à segurança, como na área de

transporte aéreo, metroviário, marítimos, sistemas médicos, geração de energia, de defesa e espacial. São considerados para esta classificação principalmente os sistemas eletrônicos, eletromecânicos, de recepção, comunicação e transmissão de dados, que utilizam intensivamente software para controle de processo, atuação de dispositivos, leitura de sensores, bem como atividades relacionadas à sincronização, monitoração e ativação de eventos em tempo real.

A definição das palavras-guia do processo ELICERE se baseou nas do estudo de operabilidade e perigos, HAZOP, e dos potenciais modos de falha, descritos nas diversas abordagens da técnica de análise de modos de falhas e seus efeitos, FMEA. Tanto o HAZOP como o FMEA deram origem a diversas variações de abordagens como SHARD, LISA e SFMEA. Tanto o SHARD/LISA como o SFMEA utilizam abordagens mais atualizadas do uso de palavras-guia dos modelos tradicionais de análise de perigos.

Para a aplicação no processo de eliciação de metas de dependabilidade, a escolha das palavras-guia foi baseada nos possíveis desvios de intenção de projeto no uso dos componentes da linguagem i*. A identificação de desvios de meta, recurso, e tarefa foram interpretadas e inter-relacionadas no contexto de PES para aplicações críticas, de modo a abranger, de forma sistematizada, todas as possíveis situações de perigo.

A interpretação das palavras-guia do componente meta leva em consideração os desvios de intenção de projeto, relacionados aos objetivos ou metas de um ator, ou às metas da missão, ou do sistema como um todo. Como meta é uma abstração de alto nível do requisito de um sistema, não cumprir uma meta, ou mesmo cumprir de forma incorreta, poderá provocar o insucesso de toda a missão. O insucesso na obtenção de uma meta pode estar diretamente relacionado a problemas ocasionados pelos recursos ou pelas tarefas necessárias para cumprir esta meta. Isto significa que para identificar perigos associados a uma meta (palavras-guia de meta), é necessário identificar perigos associados a recursos e a tarefas (palavras-guia de recurso e palavras-guia de tarefa). É necessário analisar quais são os perigos de recurso e de tarefa de uma meta que devem ser observados para que, no

final da análise, possam se definir prioritariamente metas-soft de recurso ou de tarefa que permitirão que a meta seja alcançada completamente.

A Tabela 3.1 mostra a correspondência desenvolvida das palavras-guia ELICERE aplicadas à meta tendo como referência as palavras-guia clássicas HAZOP, apresentadas por Redmill; Chudleigh e Catmur (1999) e pela norma de defesa britânica MOD-DEF-STAN-058 (2000).

Tabela 3.1 – Palavras-guia para meta.

Redmill/ MoD – Def Std058	Processo ELICERE
No	Meta Não Atingida
More, Less	
Part of, Reverse	Meta Incorreta
Other than	
As well as	Meta Adicional
Early, Late	
Before, After	Meta Fora do Tempo/Ordem

Algumas possíveis interpretações das palavras-guia para a componente meta são:

Meta Não Atingida: meta não atendida; meta não cumprida. Recurso ausente, incorreto, adicional, provocando a não obtenção da meta. Término anormal de uma tarefa ou omissão de uma tarefa provocando a não obtenção da meta. Uma incorreção de tarefa ou uma tarefa executada fora do tempo ou de ordem ocasionando a falta da meta.

Meta Incorreta: a meta alcançou um resultado além, parcial ou aquém do proposto, o objetivo não está completo, não está claro que foi alcançado o objetivo, meta alcançou um resultado oposto ao proposto, (provavelmente) outra meta foi atingida,

a meta alcançou um resultado diferente do proposto. Um recurso ausente ou incorreto teve como resultado uma meta incorreta. Recurso adicional ou fora do tempo ou da ordem incorreta. Término anormal de uma tarefa ou omissão de uma tarefa não permitiu que a meta fosse atingida corretamente. Uma incorreção em uma tarefa ou uma tarefa executada fora do tempo ou fora da ordem gerou uma meta incorreta.

Meta Adicional: a meta alcançou um resultado adicional ao que foi proposta. Um recurso incorreto ou adicional provocou uma meta adicional. Uma incorreção em uma tarefa gerou uma meta adicional.

Meta Fora do Tempo/Ordem: meta alcançada mais cedo ou mais tarde do que proposta, meta alcançada muito antes do que foi proposta, meta alcançada muito depois do que proposta. Um recurso fora do tempo ou da ordem fez com que uma meta fosse atingida no tempo ou na ordem não estipulada. Uma incorreção em uma tarefa ou uma tarefa fora do tempo ou da ordem levou a atingir uma meta fora do tempo ou da ordem estipulada.

A interpretação do componente recurso leva em consideração os desvios de intenção de projeto relacionados aos dispositivos físicos (equipamentos) ou lógicos (fluxo de dados) que interagem com o relacionamento entre os atores em estudo.

A Tabela 3.2 mostra a correspondência desenvolvida entre as palavras-guia aplicadas a Recurso no processo ELICERE e as apresentadas pelo SFMEA (LUTZ; WOODHOUSE, 1997), SHARD/LISA (PUMFREY, 2000) e as palavras-guia HAZOP clássicas discutidas por Redmill; Chudleigh e Catmur (1999) e pelo Departamento de Defesa Britânico (MOD-DEF-STAN-058, 2000).

Tabela 3.2 – Palavras-guia para recurso.

SFMEA	SHARD/LISA	Redmill MoD – Def Std058	Processo ELICERE
Absent Data	Omission Total	No	Recurso Ausente
	Comission	More Reverse	
Incorrect Data	Omission Partial	Less Part Of	Recurso Incorreto
	Value	Other Than	
Duplicate Data	Comission Repetition	As Well As	Recurso Adicional
Timing of Data Wrong	Early	Early Before	Recurso Fora do Tempo/Ordem
	Late	Late After	

Algumas possíveis interpretações para as palavras-guia do componente recurso são:

Recurso Ausente: não foi provido o recurso. Falha no hardware, perda ou falta de mensagens, ausência de valores de entrada de um sensor; falta de valores de entrada ou saída, falha no recebimento de dados necessários. Perda de dados devido à falha de hardware. Falha no sensor para enviar os dados. Dado não entregue/ detectado; não apresenta nenhuma comunicação.

Recurso Incorreto: dado ruim ou qualquer recurso que não descreva corretamente o recurso do sistema ou seu ambiente operacional. Sinais espúrios ou inesperados na saída de um dispositivo. Erro em valores relativos a rotinas de acionamento de disparadores (no inglês, *triggers*). Parâmetros errados, relativos aos limites de atraso de tempo, recurso não entregue totalmente, parte dos dados de comunicação está perdida, estrutura de dados incompleta, falta de alguns pacotes de dados em uma sequência. Recurso foi maior ou menor do que o requerido, só parte do recurso foi

oferecido, oferecido recurso oposto, outro recurso oferecido, informação entregue com valor errado.

Recurso Adicional: oferecido recurso adicional. Dado saturado, dado duplicado, ou transbordamento de dados (no inglês, *overflow*); recurso oferecido quando não é requerido, um dado proveniente de uma comunicação esperada está repetido quando não deveria estar.

Recurso Fora do Tempo/Ordem: dispositivo de hardware com acionamento aquém ou além do especificado. Dispositivo de hardware com acionamento fora da ordem especificada. Dado disponibilizado muito tarde para ser usado ou para ter a precisão necessária, dado disponibilizado cedo para ser usado ou para ter a precisão necessária, dado obsoleto usado na decisão de controle, dados espúrios, inadvertidos ou falhos que ocorrem somente com algumas entradas, recurso fornecido antes do tempo requerido, recurso fornecido depois do tempo requerido, ocorre a seqüência ABDC em uma seqüência de eventos que deveria ser ABCD, recurso fornecido com atraso, recurso fornecido antecipadamente, recurso fornecido depois.

A interpretação do componente tarefa leva em consideração os desvios de intenção de projeto relacionados às instruções comandadas diretamente por dispositivos físicos, bem como por módulos e programas computacionais, e representam as atividades ou tarefas de um ator ou sua forma de interação com outro ator.

A Tabela 3.3 mostra a correspondência desenvolvida entre as palavras-guia ELICERE aplicadas à tarefa e as apresentadas pelo SFMEA (LUTZ; WOODHOUSE, 1997), SHARD/LISA (PUMFREY, 2000), Redmill; Chudleigh e Catmur (1999) e pela norma de defesa britânica (MOD-DEF-STAN-058, 2000).

Tabela 3.3 – Palavras-guia para tarefa.

SFMEA	SHARD/LISA	Redmill MoD – Def Std058	Processo ELICERE
Halt/Abnormal Termination	Omission Partial	Part Of	Término Anormal da Tarefa
Omission	Omission Total	No	Omissão da Tarefa
Incorrect Logic/Event	Comission Repetition/Spurius	More Less Reverse Other Than As Well As	Incorreção na Tarefa
Timing/Order	Early	Early Before After Late	Tarefa Fora do Tempo/Ordem
	Late		

Algumas possíveis interpretações para as palavras-guia escolhidas para o componente tarefa são:

Término Anormal da Tarefa: tarefa interrompida de forma anormal, evento aberto, suspenso ou em impasse (no inglês, *deadlock*) em algum ponto do processo, tarefa não é executada até o final, não foi provida a tarefa até seu correto encerramento.

Omissão da Tarefa: falha de um evento ou serviço, mas o processo continua executando. Saltos (no inglês, *jumps*) e desvios (no inglês, *skips*), um componente não está respondendo a uma entrada ou produz uma saída não esperada, o serviço nunca é entregue. Uma comunicação simples é omitida, muitas (ou até todas) iterações consecutivas são omitidas, a tarefa não é provida, um componente não está respondendo a uma entrada. O serviço nunca é entregue, uma comunicação simples é omitida, muitas (ou até todas) iterações consecutivas são omitidas, a tarefa não é fornecida.

Incorreção na Tarefa: o comportamento está errado, a lógica está errada. A lógica está invertida; considerações erradas sobre o estado e sobre pré-condições, a intenção ou o requisito não reflete a implementação. Uma comunicação esperada é repetida quando não deveria ser; ocorre uma comunicação completamente inesperada, fornecida parte da tarefa; a tarefa fornecida menos vezes do que requerida. A tarefa faz somente parte do que deveria, a tarefa é fornecida mais vezes do que requerida, a tarefa faz além do proposto ou faz o oposto do que deveria. Outra tarefa é fornecida, a tarefa executa algo diferente do proposto, a tarefa faz além do requerido.

Tarefa Fora do Tempo/Ordem: tarefa ocorre no tempo errado, ou na ordem errada, evento ocorre muito cedo (premature: o sistema não estava no modo apropriado para receber ou processar o evento) ou muito tarde; a seqüência dos eventos é incorreta. A tarefa é provida quando não é requerida. Um evento que deveria preceder outro não ocorre como deveria, eventos iterativos ocorrem intermitentemente ao invés de regularmente, eventos que deveriam ocorrer somente uma vez ocorrem iterativamente. A tarefa acontece antes do tempo requerido, a tarefa é fornecida antes ou depois do tempo requerido, fornecida antes ou depois da seqüência requerida, tarefa fornecida com uma freqüência maior ou menor do que o estipulado, o serviço é entregue quando não é requerido.

As palavras-guia de Meta-soft podem também ser utilizadas no processo ELICERE para definir e aprimorar metas-soft pré-existentes, ou em uma reavaliação das metas de dependabilidade identificadas pelo próprio processo.

A interpretação das palavras-guia do componente Meta-soft leva em consideração os desvios de intenção de projeto relacionados aos objetivos ou metas não funcionais de um ator ou do sistema como um todo. A interpretação toma por base os atributos relacionados à qualidade de um ator, ou seja, atributos não-funcionais e suas características qualitativas ou quantitativas, bem como os desvios ou erros de atendimento aos requisitos não funcionais que um ator tem que cumprir ou atender.

A Tabela 3.4 mostra a correspondência desenvolvida entre as palavras-guia ELICERE aplicadas ao componente meta-soft e as palavras-guia clássicas HAZOP, apresentadas por Redmill; Chudleigh e Catmur (1999) e pela norma de defesa britânica (MOD-DEF-STAN-058, 2000).

Tabela 3.4 – Palavras-guia para meta-soft.

Redmill/ MoD - Def Std058	Processo ELICERE
No	Meta-soft Não Alcançada
More, Less	
Part of, Reverse	Meta-soft Incorreta
Other than	
As well as	Meta-soft Adicional
Early, Late	
Before, After	Meta-Soft Fora do Tempo/Ordem

A interpretação das meta-soft foi apresentada de forma genérica devido ao variado número de atributos de qualidade que podem se relacionar com metas de qualidade.

Algumas possíveis interpretações para as palavras-guia escolhidas para o componente meta-soft são:

Meta-soft Não Alcançada: meta-soft não atendida. Meta qualitativa ou quantitativa não cumprida. Nenhuma meta-soft proposta foi alcançada.

Meta-soft Incorreta: a meta-soft alcançou um resultado além do proposto. Meta-soft alcançou um resultado aquém do proposto. Meta-soft alcançou um resultado parcial ao que foi proposto. O objetivo qualitativo ou quantitativo não foi completado, não está claro que foi corretamente alcançado o objetivo. A meta-soft alcançou um

resultado oposto ao que foi proposto, provavelmente outra meta-soft foi atingida, a meta-soft alcançou um resultado diferente do que foi proposto.

Meta-soft Adicional: a meta-soft alcançou um resultado adicional ao que foi proposto. Meta qualitativa ou quantitativa adicional ao que foi proposto. Outra meta que não a proposta foi atingida.

Meta-soft Fora do Tempo/Ordem: a meta-soft alcançada mais cedo do que foi proposta. A meta-soft alcançada mais tarde do que foi proposto, meta-soft alcançada muito antes do que foi proposto. A meta-soft foi alcançada muito depois do que foi proposto.

No processo ELICERE foi definida uma estrutura semântica para formalizar o inter-relacionamento das palavras-guia de metas com as palavras-guia de recursos e de tarefas. Partindo do pressuposto de que o desvio de intenção de uma meta está diretamente relacionado ao desvio de um recurso ou tarefa, foi desenhada uma árvore para cada palavra-guia de meta com as palavras-guia de recursos e tarefas relacionadas. Este tipo de conexão semântica facilita o entendimento dos possíveis perigos desdobrados do não cumprimento de uma meta e, conseqüentemente, na condução da análise de palavras-guia e na definição de metas de dependabilidade.

3.4.2 Estrutura semântica das palavras-guia

Quando o modelo de um sistema crítico é submetido às palavras-guia do processo ELICERE ele ainda se encontra em um alto nível de abstração. Como os desvios de metas, recursos e tarefas estarão sendo analisados de forma preliminar, é necessário organizar quais palavras-guia podem estar inter-relacionadas, identificando quais desvios podem provocar a não satisfação de uma meta específica de um ator do sistema. A fim de facilitar esta etapa do processo, relacionando as palavras-guia entre si e a aplicação do questionário, foi proposta uma estrutura semântica, baseada em uma árvore de componentes e

relacionamentos entre as palavras-guia. Esta estrutura se baseou no método gráfico de representação da análise de árvore de falhas – FTA (da sigla em inglês, *Fault Tree Analysis*), onde o evento topo identifica o perigo e o evento folha (ou básico) determina sua causa. Nesta estrutura hierárquica proposta, uma palavra-guia de meta (componente meta) caracteriza-se como evento topo da árvore de perigos. No segundo nível da árvore estão as palavras-guia de recurso (componente recurso) e de tarefa (componente tarefa), eventos-base da árvore.

A identificação dos desvios de intenção do componente meta do sistema é o primeiro evento a ser interpretado pelo processo ELICERE, e que será desdobrado em desvios do componente recurso e desvios do componente tarefa. Desta forma, nesta fase de aplicação do processo é feito o relacionamento de desvio de meta em desvios de recursos e desvios de tarefas, identificando os perigos relacionados ao não atendimento desta meta analisada. A organização semântica das palavras-guia em uma estrutura em árvore considerou o modelo de classe como um sistema em série.

O modelo em série “puro”, sem redundância, foi adotado devido ao fato de que somente um desvio de intenção, isto é, identificação de uma palavra-guia relacionada (recurso ausente, por exemplo), já é suficiente para que se caracterize um perigo. Isto se deve também ao fato de que na fase do projeto onde é aplicado o processo ELICERE ainda não foi definida a arquitetura do sistema. Mesmo que fosse adotado também o modelo em paralelo, haveria situações em que seria necessário mais de um desvio de intenção de componentes (recurso ausente e incorreção na tarefa, por exemplo) para se caracterizar um perigo.

Foram criadas quatro árvores de perigos da classe meta e seus relacionamentos com os componentes recurso e tarefa. O propósito final desta atividade é auxiliar quem estiver executando o processo ELICERE a direcionar a aplicação dos questionários para as palavras-guia mais adequadas e identificar principalmente metas-soft para os desvios de recurso e tarefa.

A primeira árvore de palavras-guia, Figura 3.13, apresenta a árvore da classe “meta não atingida” e os desvios de recurso e tarefa relacionados. A não obtenção da meta significa que ou ocorreu um desvio de intenção do componente recurso, ou um desvio de intenção do componente tarefa. Neste caso, todos os tipos de relacionamento de recurso e de tarefa estão incluídos como possíveis desvios de intenção de projeto. Isto quer dizer que uma meta não alcançada pode ser advinda de um recurso ausente, incorreto, adicional ou ainda fora do tempo ou da ordem. O mesmo se aplica ao relacionamento entre meta e tarefa. A meta não alcançada pode ser advinda da omissão de uma tarefa, de uma incorreção, de seu término anormal ou de sua execução fora do tempo.

Figura 3.13 – Árvore de palavras-guia de “meta não atingida”.

A segunda árvore de palavras-guia, Figura 3.14, apresenta a árvore da classe “meta incorreta” e os desvios de recurso e tarefa relacionados. Neste caso, todos os tipos de relacionamento de recurso e de tarefa também estão incluídos como possíveis desvios de intenção de projeto. Uma meta incorreta pode ser advinda de um recurso ausente, incorreto, adicional ou ainda fora do tempo ou da ordem. A meta incorreta pode também ser advinda da omissão de uma tarefa, de uma incorreção, de seu término anormal ou de sua execução fora do tempo.

Figura 3.14 – Árvore de palavras-guia de “meta incorreta”.

A terceira árvore de palavras-guia, como mostra a Figura 3.15, apresenta a árvore da classe “meta adicional” e os desvios de recurso e tarefa relacionados. Neste caso, somente o relacionamento de recurso adicional, recurso incorreto e incorreção na tarefa estão incluídos como possíveis desvios de intenção de projeto.

Figura 3.15 – Árvore de palavras-guia de “meta adicional”.

A quarta e última árvore de palavras-guia, Figura 3.16, apresenta a árvore da classe “meta fora do tempo/ordem” e os desvios de recurso e tarefa relacionados. Neste caso, somente o relacionamento de recurso fora do tempo ou da ordem, incorreção na tarefa e tarefa fora do tempo/ordem estão incluídos como possíveis desvios de intenção de projeto.

Figura 3.16 – Árvore de palavras-guia de “meta fora do tempo/ordem”.

As árvores de palavras-guia direcionam a aplicação dos questionários adequados ao sistema estudado, e ajudam a interpretar as palavras-guias mais adequadas para a meta em questão. O questionário se propõe a identificar o tipo e os modos de falha advindos da palavra-guia e sua interpretação, conduzindo a indicação do pior nível de perigo esperado, do maior grau de severidade que este perigo pode provocar, de qual risco e a probabilidade de sua ocorrência que o sistema pode aceitar. Finalmente, a definição de metas-soft (qualitativas ou quantitativas) deve ser uma etapa adicionada ao processo para garantir um nível satisfatório para a dependabilidade do sistema.

3.4.3 Questionário das palavras-guia

A aplicação de um questionário como técnica padronizada de coleta de dados para condução da análise das palavras-guia se torna necessário para que, ao final desta etapa do processo ELICERE, sejam fornecidos subsídios suficientes para a definição de metas-soft qualitativas e quantitativas para os componentes de maior perigo potencial. A aplicação do questionário pode ser feita pelo engenheiro de requisitos em conjunto com algum membro da equipe técnica ou gerencial do projeto, que tenha um conhecimento mínimo do item investigado, de forma a proporcionar respostas que auxiliem na análise através de palavras-guia e das conseqüentes propostas de melhoria das metas de dependabilidade.

Antes da aplicação do questionário, as lideranças da equipe de engenharia de sistemas e de software, juntamente com a gerência do projeto, devem definir algumas premissas para a condução do questionário. Estas premissas devem ser, por exemplo, os tipos de falhas consideradas na análise, uma categorização para os possíveis níveis de perigo e uma classificação para a severidade das falhas. Também se deve definir uma matriz associando esses níveis de severidade a uma probabilidade de ocorrência, a fim de auxiliar na definição de metas-soft do risco considerado aceitável para este tipo de perigo, e qual é a probabilidade aceitável para este tipo de risco.

As seguintes questões devem ser respondidas para cada palavra-guia analisada:

- Qual é o tipo da falha?
- Qual o modo ou os modos de falha decorrentes desta interpretação?
- Qual é o pior nível de perigo esperado decorrente desta interpretação?
- Qual é a severidade da falha decorrente desta interpretação?
- Qual é a meta-soft relativa ao risco aceitável que se pretende correr para este componente do sistema?
- Qual é a meta-soft relativa à probabilidade aceitável de ocorrência para este componente do sistema?

- Quais são as metas-soft (quantitativas e qualitativas) relacionadas a fatores de qualidade que devem ser incorporadas para este componente do sistema?

A análise do sistema através das palavras-guia e do questionário associado indica, entre outras questões, os potenciais modos de falhas, os perigos advindos da falha e o risco aceitável. Esta identificação advém da interpretação da palavra-guia e da conseqüente verificação do atendimento, ou não, de sua condição de dependência. Condição de dependência no sentido de verificar se a relação entre um ator e outro ator, através de suas metas, recursos, tarefas ou metas-soft pode ser afetada por cada perigo associado à palavra-guia aplicada. É verificado, através de cada palavra-guia, qual o perigo decorrente da não satisfação da relação de dependência i^* que está sendo analisada.

3.4.3.1 Item tipo da falha

Neste item, a identificação do tipo da falha. O contexto de falha neste trabalho é o de defeito (do inglês, *fault*) que segundo Camargo Júnior (2002) corresponde a um problema interno de um componente, seja de hardware ou software. No que diz respeito ao hardware, o termo pode corresponder a um problema na especificação, na implementação, desgaste do componente ou a distúrbios externos. Já com relação ao software são considerados apenas os dois primeiros aspectos, especificação e implementação. O termo falha corresponde à causa hipotética do erro.

O tipo da falha deve ser baseado em uma classificação que permita agrupar as falhas segundo algum critério. Os critérios deverão ser definidos previamente de modo a facilitar a agregação dos itens do questionário em tipos de falha de origem comum. Além de facilitar o entendimento e a classificação dos tipos de falhas observados no sistema, este conjunto, ao final da análise das palavras-guia de todo o sistema, possibilitará o agrupamento dos componentes em falhas de mesma natureza, auxiliando na definição de metas gerais de dependabilidade para o sistema.

Uma classificação genérica de tipos de falhas abrange o tipo de falha de sistema e o tipo de falha entradas impróprias. O tipo de falha de sistema pode ser desmembrado em falha de hardware e falha de software. O tipo de falha entradas impróprias abrange as falhas provocadas pelo ambiente e pelo ser humano. A atribuição de falha de hardware e de software é mais facilmente observável do que atribuições relativas a falhas provocadas pelo ambiente ou por ser humano. A classificação do tipo de uma falha poderá variar dependendo do nível de detalhamento das informações que se dispõem sobre o sistema a ser construído.

A falha identificada como de alto nível será classificada como falha de sistema (SI) quando, dependendo das informações obtidas, não estiver claro ainda como será a solução de projeto (envolvendo somente hardware, ou software, ou somente intervenção humana). No caso de sistema crítico da área espacial temos, por exemplo, uma falha advinda da falta do acionamento de um atuador pirotécnico de separação de estágio de um foguete. Na fase de concepção do sistema pode não estar definido se este ator (atuador) será um dispositivo inteligente (microcontrolador) ou meramente mecânico (eletro-pirotécnico).

Como falha de hardware (HW) será classificada principalmente a falha identificada como de um componente físico do sistema. Por exemplo, o rompimento de um cabo, a queima de um circuito elétrico, a falha mecânica de um sensor, a quebra de uma válvula ou atuador.

Como falha de software (SW) será classificada principalmente a falha identificada de um componente lógico do sistema. Por exemplo, problemas de mapeamento da memória do computador, processamento relacionado à temporização, falta de parâmetros, comando incorreto, saída incorreta, valores fora da faixa especificada, satisfação de uma condição através de algum caminho inesperado, erro de lógica de programação.

Como falha de entradas impróprias (EI) será classificada a falha identificada como de alto nível quando, dependendo das informações obtidas, não for possível classificar o tipo de entrada no sistema: informação, sinal ou qualquer tipo de alimentação imprópria ao sistema. Por exemplo, se não ocorreu o acionamento do

telecomando de destruição do veículo devido a uma interferência de sinal de rádio não especificado.

Como falha de ambiente (AB) será classificada a falha identificada como aquelas decorrentes de componentes externos ao sistema veículo. Por exemplo, a existência de radiação gama no sistema elétrico do veículo, interferência eletromagnética, flutuação de potência, ventos acima dos tolerados pelo sistema de controle do foguete, condições de voo fora dos valores esperados pelo sistema.

Como falha do operador (HU) será classificada a falha de qualquer ser humano que interaja com o sistema. Pode ser uma intervenção incorreta do operador ou usuário do sistema. Por exemplo, a mudança de uma seqüência dos procedimentos de ativação dos sistemas durante a fase de pré-lançamento de um veículo, o acionamento da tele-destruição de um foguete em condições normais de voo, o acionamento de comandos fora do tempo correto durante a manutenção de um satélite em órbita, falha em responder uma condição de erro em um tempo específico.

3.4.3.2 Item modo de falha

Neste item é identificado o modo de falha, ou os modos de falha, decorrentes de perigos interpretados pela palavra-guia aplicada ao componente do sistema. Deve ser descrito o dano que pode ser causado no sistema, sob o enfoque da palavra-guia. Esta informação, por si só, não provê condições suficientes para identificar tanto os modos de falha única como os modos de falha combinados. Assim como usado na técnica FMEA, esta informação é importante para padronizar os modos de falhas bem conhecidos do sistema em questão (LEVESON, 95).

No caso específico da área espacial, como em sistemas interativos, falhas podem ser decorrentes, por exemplo, de disfunções relacionadas a recursos críticos interagindo entre atores, ou a tarefas de tempo real interagindo entre atores. Devem

ser detalhados tantos modos de falha quantos forem necessários no item do questionário.

A Figura 3.10, apresentada anteriormente, mostra o recurso “dados vôo” satisfazendo a relação de dependência entre os atores CD, TE, TS e SS. O modo de falha relacionado à palavra-guia “recurso ausente” deve ser interpretado de modo a descrever o que significa a falta do recurso “dados vôo” na relação de dependência para cada um destes atores.

Somente para a interpretação dos modos de falha (recurso ausente) entre o ator CD e o ator TE tem-se:

- Modo de falha de hardware (HW1): o hardware do CD não está disponível/funcionando;
- Modo de falha de hardware (HW2): o hardware do TE não está disponível/funcionando;
- Modo de falha de hardware (HW3): O meio de transmissão (cabo) do recurso não enviou os dados de vôo do CD para o TE;
- Modo de falha de software (SW1): a tarefa “fornecer dados telemetria”, embarcada no CD, não está operante;
- Modo de falha de software (SW2): a tarefa “adquirir dados” do sistema não está operante; e
- Modo de falha de software (SW3): o protocolo de transmissão dos dados de telemetria entre o CD e o TE não está operante.

A identificação dos modos de falha para cada palavra-guia aplicada permitirá classificar quais são as possíveis razões que podem ter levado o sistema a uma situação de perigo e até mesmo priorizar quais são os componentes do sistema (meta, recurso, tarefa ou mesmo meta-soft) que deverão receber metas-soft. Como o questionário é aplicado no início do projeto, deve ser descrito neste item pelo menos o erro ou defeito provocado pelo desvio de intenção de projeto do componente em relação à palavra-guia aplicada.

3.4.3.3 Item pior nível de perigo esperado

O item nível de perigo está relacionado à identificação do pior perigo associado ao modo ou aos modos de falha observados no item anterior. O perigo esperado está associado à possibilidade de uma avaria, ou disfunção (do inglês, *failure*), levar o sistema a desempenhar uma de suas funções incorretamente (Camargo Júnior, 2002).

Deverá ser descrita a consequência advinda do pior modo de falha e a possível não satisfação de uma relação de dependência por parte do componente analisado. O nível de perigo identificado estabelece o impacto no sucesso da missão, ou da meta estabelecida, auxiliando na classificação de seu impacto em conjunto com o grau de severidade da falha. Esta classificação pode ser baseada em norma, e pode abranger os seguintes níveis:

- Perda da vida, ou dano a pessoas (pessoal operacional ou população sobrevoada);
- Severos danos ao ambiente (gases tóxicos, destruição de mata, poluição da água);
- Danos à propriedade pública;
- Danos à propriedade privada;
- Perda das facilidades de lançamento (veículo lançador e/ou carga útil e/ou instalações no solo);
- Perda da missão;
- Degradação dos objetivos da missão ou de seu desempenho;
- Perda do sistema; e
- Insatisfação do usuário (baixa qualidade do sistema).

Para responder a esta pergunta, durante a análise da palavra-guia, deve-se ter como referência como o sucesso da missão e sua segurança podem ser postos em perigo. Em última instância, deve-se analisar qual o perigo associado à falha do componente, se este não realizasse sua relação de dependência. As consequências

podem ser classificadas de acordo com padrões específicos da área em questão do sistema estudado e até relacionados ao grau de severidade apresentado previamente no questionário.

Ainda no exemplo da Figura 3.10, a aplicação da palavra-guia “meta não realizada” para a meta “fornecer dados análise pós-vôo”, sugere que não será possível cumprir o objetivo de enviar, via telemetria de bordo, as informações provenientes do vôo a partir do ator TE para o ator TS. Neste caso, o pior nível de perigo esperado para ausência do dado é a degradação dos objetivos da missão, partindo do pressuposto de que esta meta não põe em risco a meta principal do sistema, que é colocar o satélite em uma órbita da Terra. Os dados provenientes da telemetria do veículo durante o vôo não serão enviados para a estação em solo (TS), limitando o acompanhamento da trajetória somente aos dados provenientes das estações de radares destinadas a rastrear o vôo do foguete.

3.4.3.4 Item severidade da falha

O item severidade da falha identifica a consequência observada quando do não cumprimento da relação de dependência analisada. Caso a análise da palavra-guia indique um modo de falha, associado a um nível de perigo, qual será a severidade desta falha? É feita a interpretação da palavra-guia de forma a classificar a magnitude que a falha poderá acarretar. Este item pode ser definido também como o nível do impacto do perigo analisado. A severidade deve ser classificada de acordo com uma escala, com critérios definidos em cada projeto, que depende da natureza da aplicação (solo ou embarcada, por exemplo) e sua política de risco. Esta classificação pode ser baseada em padrões já existentes, podendo abranger os seguintes itens:

- Catastrófico: perda de vidas humanas, danos permanentes ou temporários à saúde, perda do sistema, perda de interface de vôo tripulado, perda das facilidades em solo, danos ambientais severos;

- Crítico: incapacidade física temporária, doença ocupacional provisória, dano maior da interface de voo do sistema, dano maior das facilidades em solo, dano maior de propriedades públicas e privadas, danos ambientais maiores;
- Maior: completa perda de missão, maior degradação da missão; e
- Menor ou Negligenciável: degradação menor da missão ou qualquer outro efeito.

3.4.3.5 Item meta-soft

Este item do questionário tem como objetivo identificar, perante toda a análise de palavras-guia já realizada, quais são as metas-soft, qualitativas e quantitativas, mais adequadas a garantir a dependabilidade do sistema. Metas-soft podem ser definidas como as ações ou objetivos relacionados à garantia de produto de um sistema, organização ou negócio, ou mesmo referentes à garantia de processo. Garantia do processo implica em metas que promovam a melhoria dos processos de desenvolvimento, verificação e validação do sistema sob análise, de forma a garantir a confiança do funcionamento do sistema.

No processo ELICERE as metas-soft são utilizadas principalmente para descrever as expressões de mais alto nível dos requisitos não funcionais de dependabilidade do sistema a ser construído. Podem ser objetivos relacionados à segurança, disponibilidade, confiabilidade, entre outros atributos relacionados à confiança no funcionamento, expressando também as razões para as soluções de dependabilidade propostas nesta fase de concepção do sistema.

Primeiramente deve ser atribuída uma meta relacionada ao risco aceitável que o componente analisado pode correr sem que se perca a confiança no funcionamento do sistema ou que este trabalhe de forma a manter suas propriedades funcionais mínimas.

Em seguida, deve ser atribuída uma meta relacionada à probabilidade aceitável de ocorrência do risco que o sistema está disposto a correr, com relação à palavra-guia

associada ao componente sob análise. Através destas duas atribuições será possível direcionar, priorizar, e de certa forma quantificar, o esforço necessário para manter o componente dentro de uma faixa tolerável de risco.

Finalmente, devem ser definidas metas-soft qualitativas e quantitativas de dependabilidade para o processo, estabelecendo um relacionamento entre metas e fatores de qualidade. É desejável que durante a execução do processo ELICERE seja disponibilizado um conjunto de fatores de qualidade que devem ser observados durante esta última atividade, que é a definição de metas-soft obtidas a partir da análise dos componentes i^* (metas, recursos, tarefas e metas-soft). Este conjunto de fatores irá permitir a formulação de metas mais objetivas e claras que futuramente se transformarão em requisitos não funcionais.

A seguir são apresentadas as características principais das metas-soft a serem definidas para os componentes analisados pelo processo ELICERE.

a) Meta-soft “risco aceitável”

A escala de risco aceitável representa o grau de risco que cada componente analisado, com relação à palavra-guia, deverá suportar durante o funcionamento do sistema. Como todos os perigos são passíveis de ocorrer, e sempre existe uma limitação de recurso e de tempo para resolvê-los, esta escala deve ser definida no sentido de se identificar os mais prioritários e os menos prioritários, em termos da tolerância que o projeto do sistema pode aceitar. Para a definição do risco em um nível aceitável devem-se considerar orientações de agências reguladoras, padrões e normas específicas, bem como definir uma nomenclatura que permita que os envolvidos na definição do sistema, como engenheiros de segurança, de sistema e de software, possam discutir com base no mesmo entendimento do significado de cada categoria de risco.

Para cada sistema que se submeter ao processo ELICERE deve ser estabelecida previamente uma escala de graduação de risco para a aplicação nos componentes submetidos à análise das palavras-guia. A escala de risco abrangerá uma magnitude

que vai desde um risco alto até um risco baixo para garantir que os objetivos de missão do veículo se cumpram. Ao se atribuir ao componente analisado uma categoria de risco alto significará que o risco deste item é intolerável em qualquer circunstância. Um risco baixo significará que o risco é aceitável, porém deve ser monitorado.

A Tabela 3.5 mostra um comparativo entre definições de risco aceitável que podem ser utilizadas como referência e a interpretação qualitativa destes riscos pelo ELICERE. Uma das referências é a do padrão militar de segurança americano DOD-MIL-STD-882D (2000), que define uma classificação de risco que abrange as categorias Alto, Sério, Médio e Baixo. Outra referência, que usa uma classificação semelhante é a do princípio ALARP (no inglês, *As Low As Reasonably Practicable*) (STOREY, 1996).

Esta atribuição de risco aceitável, feita durante a aplicação do questionário associado às palavras-guia, poderá ajudar a indicar práticas e técnicas de dependabilidade para o projeto de forma a garantir que o componente analisado (meta, recurso, tarefa ou meta-soft) permaneça dentro de sua faixa de risco aceitável.

Tabela 3.5 – Comparativo de risco aceitável.

Nível	Mil-Std 882D	Interpretação qualitativa do risco aceitável (ELICERE)	Princípio ALARP
I	Alto	Inaceitável em qualquer circunstância. O risco não pode ser justificado, exceto em circunstâncias extraordinárias.	Intolerável
II	Sério	Tolerável somente se a redução do risco for impraticável ou seu custo for desproporcional ao benefício obtido.	Indesejável
III	Médio	Tolerável se medidas de contenção do risco forem adotadas.	Tolerável, sob análise
IV	Baixo	Aceitável, porém necessita ser monitorado. É necessário garantir que o risco se mantenha neste nível.	Aceitável.

Fonte: (DOD-MIL-STD-882D, 2000).

Também pode ser apresentada uma interpretação quantitativa do risco aceitável, levando-se em conta, por exemplo, a quantificação dos objetivos gerais de segurança adotado, por exemplo, no caso espacial, pelo AEB, através do Regulamento Técnico Geral da Segurança Espacial (AEB-RTSE-1, 2007).

b) Meta-soft “probabilidade aceitável de ocorrência”

Neste item é obtida a probabilidade aceitável de ocorrência do risco que o sistema pode correr, com relação à palavra-guia associada ao componente sob análise. De preferência deve-se definir a probabilidade aceitável baseada em uma matriz onde cada risco aceitável seja distribuído em faixas com relação à severidade da falha analisada pela palavra-guia versus uma probabilidade de ocorrência.

A categorização da severidade da falha (linha), definida previamente, deverá ser mapeada com relação a um conjunto de probabilidades de ocorrência (coluna), que deverá também ser definido previamente (qualitativa e quantitativamente). Normas de segurança podem ser adotadas como referência para a construção das faixas de probabilidade de ocorrência do risco, utilizando faixas como Improvável, Remota, Ocasional, Provável e Frequente (DOD-MIL-STD-882D, 2000) e que variam em números de falhas por ano de menor que 10^{-1} (frequente) a 10^{-6} (improvável), como o exemplo da Tabela 3.6.

Tendo em mente que o processo se aplica a sistemas em fase de concepção, em que as características e as funcionalidades do projeto não são ainda totalmente identificadas pode-se, mediante o grau de severidade estabelecido e o risco aceitável que forem atribuídos ao componente durante a aplicação do questionário, estabelecer faixas de probabilidade de ocorrência onde o risco deverá ser mantido. Cada faixa poderá ser associada a um conjunto de recomendações qualitativas e quantitativas de metas de dependabilidade.

Tabela 3.6 – Matriz de probabilidade aceitável de ocorrência do risco.

		Probabilidade de Ocorrência				
Severidade		Impro- vável	Remo- to	Ocasio- nal	Prová- vel	Fre- quente
	Catastrófica	IV	III	II	I	I
	Crítica	IV	III	II	II	I
	Maior	IV	IV	III	II	I
	Menor	IV	IV	III	III	II
Falhas por ano			10^{-6}	10^{-3}	10^{-2}	10^{-1}

FONTE: adaptada de (DOD-MIL-STD-882D, 2000).

As faixas também podem ser associadas com o nível de integridade da segurança, chamado comumente de SIL (do inglês, *Safety Integrity Level*) (IEC-61508-1, 1998). Cada faixa associada a um SIL poderá indicar técnicas, métodos, processos e recomendações apropriadas para que o processo de desenvolvimento do sistema analisado tenha um determinado rigor e que atenda às meta-soft definidas nesta análise.

c) Meta-soft “fatores de dependabilidade”

O processo ELICERE apresenta um conjunto básico de fatores como referência na condução de uma primeira definição de metas-soft do sistema analisado, que, no entanto, pode não ser suficiente para todas as peculiaridades dos projetos que aplicarem o processo. Este conjunto de fatores relacionados à qualidade, específicos para dependabilidade, está baseado no modelo proposto por Firesmith (2004, 2006) e por diversos autores, como os classificados por Romani (2007). Estes fatores são adequados para serem aplicados a sistemas computacionais de missão crítica ou de segurança crítica, como sistemas computacionais de aplicações espaciais. Isto significou a inclusão de fatores relacionados a hardware, software e outros que interagem com o sistema computacional ou, que possuem algum tipo de relação de dependência com este tipo de sistema.

As meta-soft de dependabilidade podem ser divididas em três conjuntos de fatores: capacidade de defesa, de adequação e de qualidade:

c1) Conjunto do fator “capacidade de defesa” (no inglês, *defensability*): diz respeito à forma com que o sistema crítico ou componente do mesmo se defende de acidentes e perigos. Pode ser dividido nos seguintes subfatores:

- Robustez (do inglês, *robustness*): capacidade com que um componente do sistema continua a funcionar apropriadamente sob uma dada condição ou circunstância anormal ou condições ambientais estressantes (RUS; KOMI-SIRVIO; COSTA, 2003).
- Segurança contra falhas acidentais (no inglês, *safety*): capacidade com que um componente do sistema previne o sistema contra acidentes que provoquem danos à saúde, propriedade e ambiente (FIRESMITH, 2007) ou resultem em falha de missão (NASA-GB-8719.13, 2005).
- Segurança contra falhas maliciosas ou intencionais (do inglês, *security*): capacidade com que um componente do sistema previne o sistema contra atitudes maliciosas e intencionais (FIRESMITH, 2007). Habilidade do sistema proteger-se contra intrusão acidental ou deliberada (SOMMERVILLE, 2001). Atributos do software que evidenciam a capacidade de evitar o acesso não autorizado, acidental ou deliberado a programas e dados (ABNT-NBR-ISO/IEC-9126, 1994).
- Capacidade de sobrevivência (no inglês, *survivability*): capacidade com que um componente do sistema provê serviços essenciais de missão a despeito de eventos maliciosos ou acidentais (FIRESMITH, 2007). Habilidade de uma aplicação de um sistema baseado em comunicação por computador de continuar satisfazendo a determinados requisitos críticos (por exemplo, requisitos para segurança, confiabilidade, responsividade em tempo-real e exatidão) em face de condições adversas (RUS; KOMI-SIRVIO; COSTA, 2003).
- Tolerância à falhas (no inglês, *fault tolerance*): atributo que evidencia a capacidade do sistema em manter o nível de desempenho especificado nos

casos de falhas no software ou de violação nas interfaces especificadas (CAMARGO JÚNIOR; ALMEIDA JÚNIOR; MELNIKOFF; 1997).

- Recuperabilidade (no inglês, *recoverability*): atributo que evidencia a capacidade do sistema em restabelecer seu nível de desempenho e recuperar os dados diretamente afetados em caso de falha, e o tempo e esforço necessário para tal (ABNT-NBR-ISO/IEC 9126, 1994).

c2) Conjunto do fator “adequação” (no inglês, *soundness*): diz respeito à forma com que o sistema crítico ou componente do mesmo está adequado ao uso. Pode ser dividido nos seguintes subfatores:

- Correção (no inglês, *correctness*): modo com que um componente do sistema satisfaz sua especificação e atende aos objetivos de missão e segurança (FIRESMITH, 2007).
- Disponibilidade Operacional (no inglês, *availability*): modo com que um componente do sistema permanece operacional e disponível para uso num determinado instante de tempo (FIRESMITH, 2007). Habilidade de um item estar em um estado para executar uma função requerida sob dadas condições em um dado instante de tempo, assumindo que os recursos externos são fornecidos (ECSS-P- 001B, 2004).
- Previsibilidade (no inglês, *predictability*): capacidade do sistema, ou dos componentes em prever seu comportamento para um dado conjunto de entradas em um dado estado do sistema (FIRESMITH, 2007).
- Confiabilidade (no inglês, *reliability*): capacidade do sistema, ou um componente operar sem falhas sob dadas condições normais, durante certo período de tempo (FIRESMITH, 2007). A habilidade de um item executar suas funções requeridas sem falha ou defeito para a duração de seu perfil de vida (MOD-DEF-STAN-00-40, 2003).
- Estabilidade (no inglês, *stability*): é um subtipo do fator confiabilidade, e diz respeito ao modo com que um serviço de missão crítica de um sistema continua a ser provido durante um dado período de tempo sob dado perfil operacional, a despeito de qualquer falha provocada durante sua entrega ou na identificação do evento causador da falha (FIRESMITH, 2007).

- Auto-descrição (no inglês, *self-description*): capacidade do sistema de permitir uma maior facilidade de seu entendimento em futuras manutenções, diminuindo a possibilidade de introdução de novos erros (CAMARGO JÚNIOR; ALMEIDA JÚNIOR; MELNIKOFF, 1997).
- Completeza (no inglês, *completeness*): capacidade do sistema em evitar que exista omissão em relação a algum aspecto de sua aplicação que pode levar o sistema a atingir um estado inseguro (CAMARGO JÚNIOR; ALMEIDA JÚNIOR; MELNIKOFF, 1997).
- Consistência (no inglês, *consistency*): capacidade do software de verificar erros que possam levar o sistema a uma situação insegura (CAMARGO JÚNIOR; ALMEIDA JÚNIOR; MELNIKOFF, 1997).

c3) Conjunto do fator “qualidade” (no inglês, *quality*): diz respeito a outras características de qualidade de produto que podem influenciar a sua dependabilidade. Pode ser dividido nos seguintes subfatores:

- Testabilidade (no inglês, *testability*): capacidade com que o sistema ou um componente facilita a criação e a execução de testes de software (por exemplo, testes que causam falhas devido a defeitos não aparentes) (FIRESMITH, 2007). Atributos do software que evidenciam o esforço necessário para validar o software modificado (ABNT-NBR-ISO/IEC 9126, 1994).
- Capacidade (no inglês, *capacity*): capacidade do sistema, ou componente poder manipular com sucesso em um instante de tempo ou em um intervalo de tempo um número máximo de objetos (FIRESMITH, 2007).
- Eficiência (no inglês, *efficiency*): capacidade de usar de forma eficiente os recursos por parte do sistema ou componente (por exemplo, o mínimo de consumo de um recurso) (FIRESMITH, 2007).
- Desempenho (no inglês, *performance*): capacidade do sistema, ou componente operar com relação às suas restrições temporais de projeto (FIRESMITH, 2007). Refere-se aos aspectos de temporização que são fundamentais num sistema crítico (CAMARGO JÚNIOR; ALMEIDA JÚNIOR; MELNIKOFF, 1997). Desempenho é tipicamente decomposto em subfatores

como retardo, latência, tempo de resposta, capacidade de escalonamento e taxa de transferência.

- Retardo (no inglês, *jitter*): variação do tempo entre o período projetado para uma ação ser executada e o período que esta ação realmente acontece (FIRESMITH, 2007).
- Latência (no inglês, *latency*): tempo que o sistema ou componente leva para executar uma tarefa específica fim a fim (FIRESMITH, 2007).
- Tempo de resposta (no inglês, *response time*): tempo que o sistema ou componente leva para iniciar a resposta a uma requisição do cliente para um determinado serviço ou para permitir o acesso a um recurso (FIRESMITH, 2007).
- Capacidade de escalonamento (no inglês, *schedulability*): capacidade do sistema ou de um componente, modelar e programar o comportamento de seus eventos e de suas tarefas de forma determinística (FIRESMITH, 2007).
- Taxa de transferência (no inglês, *throughput*): número de vezes que o sistema ou o componente é capaz de completar uma operação ou prover um serviço dentro de uma unidade de tempo especificada (FIRESMITH, 2007).
- Usabilidade (no inglês, *usability*): capacidade do sistema de facilitar a interação humana e ser capaz de alcançar seus objetivos específicos de uso, dentro de um determinado contexto e para um determinado grupo de usuários (FIRESMITH, 2007).
- Previsibilidade (no inglês, *predictability*): capacidade de se conhecer o comportamento determinístico de um sistema ou componente (FIRESMITH, 2007).
- Manutenção (no inglês, *maintainability*): facilidade com que é possível fazer mudanças no sistema ou em um componente, executada sob condições, procedimentos e recursos indicados (FIRESMITH, 2007). Habilidade de um item sob dadas condições de uso, estar retido em, ou recuperado para, um estado no qual possa executar uma manutenção sob dadas condições e usando procedimentos e recursos indicados (ECSS-P- 001B, 2004).

- Capacidade de expansão (no inglês, *expandability*): facilidade de melhoria ou expansão do sistema ou um componente, de forma a atender futuras metas e significantes mudanças de requisitos (FIRESMITH, 2007).
- Simplicidade (no inglês, *simplicity*): característica do sistema de facilitar a avaliação da sua segurança (CAMARGO JÚNIOR; ALMEIDA JÚNIOR, MELNIKOFF, 1997).
- Modularidade (no inglês, *modularity*): capacidade do sistema de possuir um alto grau de acoplamento, isto é, interdependência entre seus módulos e baixa coesão, isto é, o módulo inclui duas ou mais funções independentes (CAMARGO JÚNIOR; ALMEIDA JÚNIOR; MELNIKOFF, 1997).
- Portabilidade (no inglês, *portability*): capacidade e esforço requerido pelo sistema para ser transferido de um ambiente de hardware e/ou software para outro (KITCHENHAM; PFLEEGER, 1996).
- Precisão (no inglês, *accuracy*): capacidade do sistema de gerar resultados ou efeitos corretos ou conforme acordados (CAMARGO JÚNIOR; ALMEIDA JÚNIOR; MELNIKOFF, 1997).
- Rastreabilidade (no inglês, *traceability*): capacidade com que o sistema permite com que todos os requisitos gerais de segurança sejam perfeitamente mapeáveis na sua especificação e na sua respectiva implementação (CAMARGO JÚNIOR; ALMEIDA JÚNIOR, MELNIKOFF; 1997).

3.5 CONSIDERAÇÕES

Para a correta aplicação e interpretação do processo ELICERE deve-se ter em mente que meta não é requisito. Enquanto meta é geralmente alguma propriedade do sistema em um alto nível de abstração, requisitos devem possuir características objetivas e testáveis (KOTONYA; SOMMERVILLE, 2000). Entretanto, apesar de que metas ou metas-soft não sejam conceitualmente requisitos, elas podem levar à identificação de um requisito ou requisitos não funcionais, que devem ser definidos para melhorar, ou até mesmo garantir a confiança no funcionamento do sistema.

Nos primeiros estágios de definição do sistema a ser construído, fase do desenvolvimento do projeto onde o processo ELICERE se insere, não se deve desprezar a utilidade de uma meta, dentro de uma abordagem sistemática, para descobrir e estruturar requisitos. Segundo van Lamsweerde; Darimont e Massonet (1995), metas devem ser refinadas em requisitos, que devem ser formulados em termos de estados controlados por algum agente individual do sistema. As metas, inclusive, justificam e explicam a presença de alguns requisitos que não são necessariamente compreendidos pelos clientes (DARDENNE; VAN LAMSWEERDE; FICKAS, 1993).

Para a obtenção de um resultado satisfatório, é preciso, dentro do processo ELICERE, recomendar metas bem objetivas e sem ambigüidade, suficientemente específicas para que, a partir da definição de uma meta-soft possa-se escolher requisitos de dependabilidade adequados ao sistema. Depois de completado o questionário com a identificação de metas-soft qualitativas e quantitativas, a identificação de requisitos não-funcionais será uma consequência natural da análise das palavras-guia aplicadas aos componentes do sistema modelado em i*. Dentro desta proposta, a derivação de metas-soft em requisitos não funcionais será possível observando-se principalmente um padrão de objetivos (metas) entre os componentes do sistema e as necessidades às quais esses objetivos devem atender (requisitos).

O mapeamento de metas em requisitos poderá ser feito, por exemplo, através de técnicas como a decomposição de metas em requisitos não funcionais proposta por Mylopoulos; Chung e Nixon (1992) ou pela técnica de decomposição dos interesses (em inglês, *concerns decomposition*) em requisitos não funcionais (SOMMERVILLE, 2001).

Segundo Kotonya e Sommerville (2000), as metas ou interesses que são identificados durante o processo de elicitação, correspondem geralmente a objetivos estratégicos do sistema, como os objetivos críticos do negócio (padronização através de normas, por exemplo) ou podem refletir características essenciais, como segurança. Os autores citam ainda que as metas devem ser consideradas no contexto de cada estágio da elicitação de requisitos, inclusive dentro de um

processo de prioridade. Neste caso específico, o mapeamento de metas em requisitos abrange primeiramente a identificação da meta (ou interesse) com relação ao conjunto de metas ou mesmo requisitos propostos previamente. Isto quer dizer, identificar a meta do usuário e seu requisito associado a um conjunto prévio. No livro é detalhado, conforme a Tabela 3.7, o relacionamento entre necessidades/interesse dos usuários e sua derivação em requisitos não funcionais.

Tabela 3.7 – Relacionamento entre interesses e requisitos não funcionais.

Necessidade do usuário	Interesse do usuário	Requisito não funcional
Função	Fácil de usar	Usabilidade
	Autorização de acesso	Segurança contra falhas maliciosas
	Probabilidade de falha	Confiabilidade
Desempenho	Resposta rápida	Tempo de resposta
	Utilização do recurso	Eficiência
	Controlar eventos	Capacidade de escalonamento
Mudança	Fácil de reparar	Capacidade de manutenção
	Fácil de mudar	Flexibilidade
	Fácil de expandir	Capacidade de expansão

Fonte: (KOTONYA; SOMMERVILLE, 2000).

É importante salientar que para este processo de derivação atingir um nível objetivo e claro do requisito, podem ser necessários desdobramentos, isto é, uma meta pode derivar um conjunto de submetas, até se alcançar um nível de granularidade de meta que possa ser mapeado em um requisito, ou conjunto de requisitos detalhados e atômicos.

Deve-se ficar claro que a proposta do processo ELICERE não engloba a validação ou a manutenção das metas-soft identificadas. Essas atividades podem e devem também serem realizadas em conjunto com o ELICERE, a fim de avaliar quais as metas de dependabilidade são mais adequadas e prioritárias para o projeto e como mantê-las. É preciso aplicar o processo ELICERE e as atividades de validação e manutenção em todas as vezes que alterações de projeto forem incorporadas.

4 ESTUDO DE CASO DO PROJETO V-ALFA

O estudo de caso descreve um veículo lançador de satélite hipotético, intitulado V-ALFA e, portanto, não pertence a algum foguete ora em desenvolvimento no Brasil. Os objetivos apresentados baseiam-se em dados coletados de material não reservado do projeto do primeiro veículo, de uma série de foguetes a serem desenvolvidos dentro do programa espacial brasileiro, intitulado Cruzeiro do Sul (MORAES et al., 2006) (VILLAS BOAS, 2006) e não compreendem um conjunto completo de objetivos de missão. Outras informações são frutos também da pesquisa e da experiência do autor desta tese na área de sistemas computacionais para aplicações críticas.

Como objetivos da missão do V-ALFA no estudo de caso, são apresentados apenas os mais relevantes, de forma condensada, partindo do pressuposto que algumas soluções sistêmicas já tenham sido tomadas, baseadas em projetos anteriores ou mesmo em limitações tecnológicas e de mercado.

Para utilização no estudo de caso foram selecionados alguns componentes que representam tanto os relacionamentos dos sistemas de apoio em solo, como aqueles a bordo do veículo. Uma ênfase foi dada ao sistema de controle de bordo do foguete devido às suas características de missão crítica e à diversidade de tarefas e recursos necessários para o cumprimento da missão do V-ALFA.

Foram representados subsistemas, como o sistema de apoio em solo, que executa todas as monitorações e verificações para o lançamento do veículo até o acionamento do comando de ignição dos motores do primeiro estágio; o operador do sistema de apoio em solo, o sistema de telemetria embarcado e em solo, e o sistema de segurança em terra, e o operador de segurança em solo.

De modo geral, um sistema de controle de bordo é representado por sensor, atuador, controlador e o sistema físico a ser controlado (BLETSOS, 2004). A Figura 4.1 mostra uma visão geral de um sistema de controle (CASTRO LEITE, 2002) para manter na trajetória planejada um veículo dinamicamente instável, como o V-ALFA.

Figura 4.1 – Exemplo de sistema de controle de um veículo espacial.

Fonte: (CASTRO LEITE, 2002).

O veículo representa o sistema físico a ser controlado. O sensor representa os dispositivos para medidas das variáveis a serem monitoradas. O atuador representa os dispositivos que produzem força ou torque para a modificação do comportamento das variáveis controladas e, portanto, do sistema físico. O controlador representa os cálculos de comparação entre o valor medido e o valor desejado das variáveis a serem controladas e o tratamento matemático adequado dado a estes erros para gerar comandos para que os atuadores executem as ações de controle.

4.1 O VEÍCULO LANÇADOR DE SATÉLITES V-ALFA

Nesta seção são descritos os principais objetivos de missão do veículo espacial V-ALFA, enfocando principalmente as atribuições relacionadas ao seu sistema computacional.

O veículo espacial V-ALFA representa um foguete de três estágios que tem como missão colocar em órbitas baixas (com inclinação de 400 km de altitude aproximadamente) uma carga útil (satélite) com massa acima de 400 kg, numa faixa de inclinação de 0 a 100 graus, a ser lançado a partir do Centro de Lançamento de Alcântara, no Maranhão.

O primeiro estágio do V-ALFA será composto por quatro propulsores a propelente sólido, fixados em torno do segundo estágio (configuração conhecida como *strap-ons*), formado por um único motor a propelente sólido, maior e central ao eixo do veículo. O terceiro e último estágio, será composto por um motor a propelente líquido de 75kN de empuxo.

O veículo será controlado nos seus três eixos (arfagem-Y, guinada-Z e rolamento-X) durante as fases propulsadas de 1°, 2° e 3° estágios, sendo que no 3° estágio ele será estabilizado de maneira a se posicionar para a ejeção da carga útil em órbita pré-determinada. As informações necessárias ao controle do veículo são provenientes de uma plataforma inercial do tipo “*strapdown*”, que se comunicará com o controlador digital através de um barramento serial em uma frequência de 100 Hz.

A pilotagem do foguete na fase propulsada será através do controle do vetor de empuxo, utilizando para isto um sistema de tubeiras móveis em todo o voo controlado. O primeiro estágio, constituído por quatro tubeiras móveis, apresentará controle em torno dos três eixos do veículo. O segundo estágio será composto por uma tubeira móvel que proporciona torque somente em torno dos eixos de arfagem e guinada, portanto, o controle em rolamento será feito por um sistema independente, equipado com quatro tuberias fixas. Para o terceiro estágio, embora seja um propulsor único, haverá um único sistema de controle que, devido às características do propulsor líquido, é formado também por um sistema de quatro pequenas tubeiras, que através de uma combinação de seu acionamento, é capaz de promover a atuação nos três eixos do veículo.

Para minorar os erros de trajetória provocados pelas dispersões propulsivas, variação de massa e de arrasto, perturbações externas, entre outros fenômenos, será utilizado um algoritmo de guiamento na fase propulsada do terceiro estágio, que permitirá colocar o foguete na posição adequada para que uma das órbitas pré-estipuladas a bordo seja alcançada.

O controlador digital é formado pela integração de um computador de bordo robustecido, suas unidades de interfaces e o software embarcado. Ele realizará, em

tempo real, os cálculos necessários para a condução do veículo até seu devido posicionamento para ejeção da carga útil. Este sistema ficará situado na baia de equipamentos do terceiro estágio, e será responsável por gerenciar os dados de entrada e saída do veículo.

As informações de entrada do controlador digital, fornecidas pelo sistema de apoio em solo, são:

- Atribuição dos estados e modos do sistema durante a fase de pré-vô;
- Identificação dos testes a serem realizados para verificação das redes elétricas do veículo durante a fase de pré-vô;
- Perfil de trajetória do vôo a ser carregado na fase de pré-vô; e
- Comando de iniciação do vôo (NAV-ON);

Os incrementos dos ângulos de atitude do veículo (ΔAlfa , ΔBeta , $\Delta\text{Epsilon}$) e incrementos de aceleração (ΔV_x , ΔV_y , ΔV_z) do veículo, são fornecidos pelo sensor inercial (pré-vô e vôo).

Informações de saída do controlador digital:

- Diagnóstico da iniciação do computador durante a fase de pré-vô, resultado dos testes de preparação das redes elétricas do veículo durante esta fase, fornecidas para o sistema de apoio em solo;
- Comandos de ativação da seqüência de eventos e de controle de vôo fornecido para os sistemas de atuadores e do seqüenciador de eventos; e
- Monitorações dos eventos e informações de controle do veículo com seus respectivos tempos durante a fase de vôo, fornecidos para o sistema de telemetria embarcada.

Todas as informações do veículo serão depositadas no barramento serial de dados, para serem transmitidos na fase de pré-vô entre o controlador digital, o sistema de apoio em solo e o sensor inercial; e na fase de vôo entre o controlador digital, o sensor inercial e o sistema de telemetria embarcada para envio à estação de telemetria em solo. As informações advindas das monitorações de controle do

veículo e de sua seqüência de ativação de eventos também serão utilizadas pelo sistema de segurança em solo para auxiliar na decisão do operador de segurança em tele-destruir o veículo, no caso de descumprimento de qualquer uma das restrições observadas no plano de segurança de voo.

4.2 A MODELAGEM DO SISTEMA COMPUTACIONAL DO V-ALFA

Nesta primeira atividade do processo foi criado o modelo de Dependências Estratégicas (na sigla em inglês, SD) e o modelo de Razões Estratégicas (na sigla em inglês, SR) do ator Controlador Digital (CD), do sistema V-ALFA.

Os modelos gerados têm como objetivo manter uma visão simplificada do sistema, a fim de abstrair somente as relações de dependência mais significativas para a análise. O objetivo é de representar os elementos mais importantes e estratégicos para a condução do processo e, caso necessário, explorar com mais detalhes alguma particularidade do sistema.

Foram utilizadas como entrada, para esta fase do processo, as informações oriundas da definição preliminar dos objetivos de missão do sistema a ser construído. A partir dessas informações foram identificados os atores e os agentes do sistema que interagem com o ator CD.

Subsistemas do veículo foram representados como atores, tais como o sistema inercial de navegação, o sistema de controle dos atuadores, o sistema de apoio em solo para lançamento e monitoração do veículo, entre outros. Os operadores em solo foram representados com agentes, tais como o operador do banco de controle do veículo e o operador de segurança de voo. Logo após, metas foram derivadas dos objetivos do sistema proposto. Dependendo de como o sistema é concebido, as metas identificadas nesta fase podem ser muito próximas dos requisitos funcionais do sistema. Inicialmente a criação de um modelo SD baseado em metas auxiliou no entendimento do sistema, e serviu também como uma primeira aproximação para a identificação dos recursos e tarefas necessárias para atender às metas propostas.

Em seguida foram detalhados os recursos necessários para que os subsistemas se comuniquem e realizem suas metas. A visão do modelo SD de recursos é extremamente necessária para a visualização dos dispositivos físicos ou lógicos que o sistema necessita para realizar as metas. Finalmente, foram criados modelos SR focados principalmente na identificação das tarefas que cada ator deveria executar para atender às metas definidas, dispondo dos recursos apresentados.

4.2.1 Os modelos SD do veículo V-ALFA

A partir dos objetivos de missão do veículo espacial V-ALFA, apresentados na seção 4.1, são extraídas as principais metas e recursos que os atores identificados previamente devem realizar e dispor no sistema a ser construído, levando em consideração as relações de dependência entre eles.

O modelo SD foi apresentado sob duas óticas: da relação dos atores e das metas estabelecidas, e da relação dos atores e dos recursos disponíveis. O ator CD, representa o sistema computacional a ser investigado e suas relações de dependência com os atores externos (interfaces). Tanto metas que interagem com outros atores (metas externas) como as metas que não interagem com outros atores (metas internas) são modeladas, de modo a identificar os principais objetivos do sistema computacional. Para o modelo SD, os atores que interagem com o CD são apresentados sem a identificação de suas metas internas. O objetivo foi elaborar um modelo essencial do sistema, contendo apenas suas metas estratégicas. No modelo SD podem ser representadas as submetas do sistema, ou seja, metas derivadas de metas, que detalham os objetivos do sistema a ser construído.

Os principais relacionamentos entre o ator CD e os outros atores artificiais e agentes humanos são:

- AS (sistema de Apoio em Solo): se comunica com o CD na fase de pré-vôo para transmitir dados, realizar testes, receber as respostas do CD e comandar o início do vôo (NAV ON) até a desconexão dos cabos umbilicais;

- CD (Controlador Digital): além de todas as metas definidas anteriormente, tem como atribuição guiar o veículo o mais próximo da trajetória desejada, apontá-lo no ângulo possível e mais adequado, estabilizar sua rotação através de um movimento de basculamento, separar a coifa que envolve o satélite e emitir o comando de ejeção desta carga útil. Monitora os tempos de todos os eventos realizados durante o voo.
- CR (Sistema de Controle de Rolamento): recebe do CD os comandos de controle de rolamento do segundo estágio do V-ALFA;
- Op_AS (Operador do sistema de Apoio em Solo): envia comandos e recebe respostas, através do console do AS, durante a fase de pré-voo do V-ALFA;
- Op_SS (Operador do sistema de Segurança em Solo): envia comando de destruição do veículo no caso da violação de algum requisito de segurança da missão;
- SA (sistema de Segurança e Atuação): recebe comandos do CD relativos à seqüência de eventos que o veículo deve cumprir.
- SI (sistema Sensor Inercial): recebe informações do CD para iniciar sua preparação para voo e envia informações para que o CD possa realizar a navegação do veículo;
- SP (Sistema de Sensor de Pressão): envia dados para o CD de pressão dos motores dos estágios do veículo V-ALFA;
- SS (Sistema de Segurança em Solo): recebe do TE os dados de telemetria do veículo necessários para que o sistema de segurança de voo acompanhe a trajetória do V-ALFA; recebe do Op_SS o comando de destruição do veículo, caso seja violado algum dos requisitos de segurança de voo;
- TE (Sistema de Telemetria Embarcada): recebe do CD os dados de telemetria do voo e envia para o TS os dados necessários para análise pós voo e envia para o SS os dados necessários para a segurança de voo;
- TM1E (Sistema de Atuação de Tuberias Móveis do Primeiro Estágio): recebe do CD os comandos de controle de atitude e de rolamento do primeiro estágio do V-ALFA, e também recebe comandos de controle de atitude e rolamento do conjunto primeiro e segundo estágios; e
- TM2E (Sistema de Atuação de Tuberias Móveis do Segundo Estágio): recebe do CD os comandos de controle e atitude do conjunto primeiro e segundo

estágios e também recebe comandos de controle de atitude somente do segundo estágio do V-ALFA;

- TM3E (Sistema de Atuação de Tuberias Móveis Terceiro Estágio): recebe do CD os comandos de controle de atitude e de rolamento do terceiro estágio do V-ALFA;
- TS (Sistema de Telemetria em Solo): recebe do TE os dados necessários para análise pós-vôo do V-ALFA;

Como base na descrição anterior, foi desenvolvido o modelo SD de metas e o modelo SD de recursos do sistema computacional do V-ALFA, apresentados nas Figuras 4.2 e 4.3 respectivamente.

Figura 4.2 – Modelo SD do sistema CD relacionado à visão de metas.

Figura 4.3 – Modelo SD do sistema CD relacionado à visão de recursos.

4.2.2 Os modelos SR do veículo V-ALFA

Por uma questão pedagógica, cada modelo SR desenvolvido para este estudo foi apresentado em conjunto com a análise das palavras-guia e aplicação dos questionários dos exemplos do estudo de caso. A intenção foi detalhar somente as tarefas do sistema V-ALFA, utilizadas para ilustrar a aplicação das palavras-guias e dos questionários, de forma o mais simples e direta possível.

4.3 A PREPARAÇÃO PARA A APLICAÇÃO DAS PALAVRAS-GUIA

Para a aplicação do processo ELICERE nas próximas seções deste capítulo, foi escolhido um conjunto de atores, metas, recursos e tarefas diretamente relacionados ao Controlador Digital e seus modelos de Dependências Estratégicas (SD) e de Razões Estratégicas (SR).

Foram exemplificadas algumas das relações de dependência do ator CD com os meios de apoio em solo, de relações de dependência do sistema embarcado e ainda algumas relações de dependência entre atores externos ao CD, como o do sistema de telemetria embarcada, e atores humanos, como Operador de Apoio em Solo e Operador de Segurança em Solo.

A aplicação do processo ELICERE neste estudo teve a intenção de exemplificar alguns casos particulares para demonstrar quão factível é sua utilização. Não procura abranger necessariamente todos os possíveis componentes do sistema e nem representa um processo de validação do sistema.

Antes da aplicação das palavras-guia, representantes da engenharia de sistemas do veículo, ou os responsáveis pelo projeto, devem estabelecer um conjunto de atributos de dependabilidade a serem utilizados na definição e priorização de metas-soft para o sistema a ser construído, como sugerido no capítulo 3 desta tese.

Devem ainda serem definidas classificações de referência para os níveis de perigo e de severidade. Neste sentido, para este estudo de caso, os níveis de perigo adotados foram os seguintes:

- PERDA OU DANOS À VIDA HUMANA: perda da vida, ou dano a pessoas, ao pessoal operacional, ou à população sobrevoada;
- PERDA DA MISSÃO: não obtenção dos objetivos definidos para a missão proposta;
- PERDA DO SISTEMA: perda de um dos sistemas do veículo, podendo comprometer, ou não, os objetivos da missão;
- DANOS À PROPRIEDADE: danos as propriedades públicas e privadas devido à perda da missão ou do sistema;
- DANOS AO AMBIENTE: severos danos ao ambiente, como contaminação do ar por gases tóxicos, destruição de mata, poluição da água;
- PERDA DAS FACILIDADES DE LANÇAMENTO: perda do veículo lançador e/ou carga útil e/ou instalações no solo;
- DEGRADAÇÃO DA MISSÃO: perda de desempenho, ou de qualquer objetivo que não comprometa a missão como um todo; e
- QUALIDADE INSATISFATÓRIA: baixa qualidade do sistema ou incapacidade de obtenção das características de garantia de produto propostas para o sistema.

A escala de severidade, Tabela 4.1, foi baseada no modelo da ESA, o ECSS-Q-ST-30C (2009) e no modelo da AEB, o regulamento AEB-RTSE-1 (2007).

As faixas de severidade representam os níveis de integridade do sistema, ou de cada parte dele, em termos da magnitude que uma possível falha pode provocar no sistema espacial como um todo.

Tabela 4.1 – Classificação de severidades.

SN	Severidade	Nível de Perigo Associado (Consequência)
1	Catastrófica	Perda de vidas, danos permanentes à saúde, Perda de instalações de lançamento e/ou do sistema. Efeitos severos de detrimento do meio ambiente. Perda do Sistema. Perda das facilidades de lançamento.
2	Crítica	Perda completa da missão. Danos temporários à integridade física de pessoas Danos maiores às instalações, ao sistema de vôo ou de solo Danos maiores à propriedade pública e privada. Efeitos maiores de detrimento do meio ambiente. Dano maior das facilidades de apoio em solo.
3	Maior	Degradação do desempenho. Degradação dos objetivos da missão. Danos menores às pessoas e/ou instalações, sistemas e equipamentos. Atraso ou adiamento de marcos importantes do projeto.
4	Menor (ou Negligenciável)	Danos pequenos às instalações, sistemas e equipamentos. Atividades adicionais são necessárias. Atraso ou adiamento de marcos do projeto. Efeito marginal no desempenho. Impacto mínimo ou sem impacto no cronograma do projeto.

Fonte: (ECSS-Q-ST-30C, 2009) (AEB-RTSE-1, 2007).

4.4 CONSIDERAÇÕES PRELIMINARES SOBRE A DEFINIÇÃO DE METAS-SOFT

A definição de metas-soft no processo ELICERE leva primeiramente em consideração a definição de qual é o risco e sua probabilidade de ocorrência aceitáveis para o sistema. Com estas informações já é possível classificar e priorizar as metas que deverão ser efetivamente transformadas em requisitos.

A escala de risco aceitável, Tabela 4.2, foi baseada na interpretação de classes de risco do princípio ALARP (sigla do inglês, As Low As Reasonably Practicable)

(STOREY, 1996) e representa os níveis de tolerância dentro de uma escala, em que cada componente analisado pelo processo terá que se enquadrar.

Tabela 4.2 – Risco aceitável.

classe	nível	Interpretação
I	ALTO	O risco do não atendimento da palavra-guia do componente analisado é inaceitável em qualquer circunstância. O risco não pode ser justificado, exceto em circunstâncias extraordinárias.
II	SÉRIO	O risco do não atendimento da palavra-guia do componente analisado é tolerável somente se a redução do risco for impraticável ou seu custo for desproporcional ao benefício obtido.
III	MÉDIO	O risco do não atendimento da palavra-guia do componente analisado é tolerável se medidas de contenção do risco forem adotadas.
IV	BAIXO	O risco do não atendimento da palavra-guia do componente analisado é aceitável, porém necessita ser monitorado. É necessário garantir que o risco se mantenha neste nível.

Fonte: (STOREY, 1996)

Para este estudo de caso ficou estabelecido que na análise das palavras-guia do V-ALFA, se um componente estudado for classificado com um grau de severidade “catastrófica”, e se enquadrar, por exemplo, na graduação de risco “baixo”, classe IV, significará adotar medidas de contenção de risco, ou seja metas-soft, que mantenham o risco dentro de uma frequência “inacreditável” de ocorrência, como apresentado na Tabela 4.3. Metas qualitativas e quantitativas devem ser definidas para que o componente se mantenha dentro desta faixa de ocorrência, como por exemplo, utilização de recursos com qualidade espacial, recursos redundantes ou, no caso de tarefa, adoção de técnicas de programação defensiva.

Por outro lado, se um componente estudado for classificado com um grau de severidade “crítico”, e se enquadrar, por exemplo, na graduação de risco “médio”, classe III, significará adotar medidas de contenção de risco, ou seja, metas-soft, que mantenham o risco dentro de uma frequência “remota” de ocorrência, caso se opte pela frequência de maior tolerância da faixa, como apresentado na Tabela 4.3.

Metas qualitativas e quantitativas devem ser definidas para que o componente se mantenha dentro desta faixa de ocorrência, como por exemplo, adotar uma probabilidade de ocorrência de acidente da ordem de 10^{-6} falhas por ano. Se um componente estudado for classificado com um grau de severidade “menor”, e se enquadrar, por exemplo, na graduação de risco “médio”, classe III, significará criar metas-soft que mantenham o risco dentro de uma frequência “ocasional” ou “provável” de ocorrência, como apresentado na Tabela 4.3.

Tabela 4.3 – Probabilidade aceitável de ocorrência do risco.

		Probabilidade de Ocorrência					
Severidade		Inacreditável	Impro-vável	Remo-to	Ocasio-nal	Prová-vel	Fre-quente
	Catastrófica	IV	III	II	I	I	I
	Crítica	IV	IIII	III	II	I	I
	Maior	IV	IV	III	III	II	I
	Menor	IV	IV	IV	III	III	II
Falhas por ano		10^{-7}	10^{-6}	10^{-3}	10^{-2}	10^{-1}	

A seguir são apresentados os resultados da aplicação do processo ELICERE em cada componente e sua interpretação das palavras-guia para o estudo de caso V-ALFA. Vale comentar que não foram aplicadas palavras-guia de meta-soft para os elementos do sistema V-ALFA.

Foi dada ênfase às relações de dependência envolvendo meta, tarefa e recurso, levando em consideração que no sistema analisado as metas-soft ainda não foram definidas.

4.5 A APLICAÇÃO DAS PALAVRAS-GUIA NO MODELO DO V-ALFA

Neste capítulo, para cada caso do V-ALFA onde foi aplicado o processo ELICERE, foi ilustrado com exemplo extraídos dos modelos SD de metas e recursos

apresentados anteriormente (Figuras 4.2 e 4.3). Foram desenvolvidos sete modelos SR específicos do sistema V-ALFA e aplicado palavras-guia para sete metas, seis recursos e sete tarefas.

As metas-soft escolhidas para exemplificar a aplicação do processo ELICERE, através dos modelos SR, foram baseadas nas seguintes metas do sistema V-ALFA:

- Fornecer dados inerciais;
- Fornecer telemedidas;
- Preparar para vôo;
- Controlar atitude e rolamento 1E;
- Controlar rolamento 2E;
- Garantir segurança de vôo;
- Habilitar modo de operação.

As palavras-guia selecionadas para exemplificação foram destacadas pela cor cinza nos diagramas de árvores de palavra-guia.

4.5.1 A aplicação de palavras-guia na meta “fornecer dados inerciais”

A Figura 4.4, sintetizada das Figuras 4.2, 4.3 e acrescida do modelo SR correspondente, apresenta as relações de dependência entre o ator CD e o ator SI a respeito da meta “fornecer dados inerciais” e da meta “realizar navegação”. O não atendimento a meta “realizar navegação” significa que a meta “fornecer dados inerciais” não foi atingida. Este fato pode ter sido provocado porque os recursos disponibilizados pelo ator SI não foram fornecidos ao ator CD, ou algum problema relacionado às tarefas atribuídas a esta funcionalidade do sistema, de responsabilidade do SI ou do CD.

Figura 4.4 – Modelo SR da meta “fornecer dados inerciais”.

Uma das condições necessárias para se alcançar a meta “realizar navegação”, é que o recurso contendo os incrementos dos ângulos de atitude dos três eixos do sistema do corpo do veículo “deltaAlfa, deltaBeta, deltaEpsilon”, fornecido pelo ator SI, esteja disponível para o ator CD. Se este recurso não for fornecido, a meta não será atingida e a missão não terá sucesso.

Para alcançar a meta “realizar navegação” e a meta “fornecer dados inerciais”, é necessário também que o ator CD execute diversas tarefas, entre elas a tarefa “adquirir incremento de ângulo de atitude”. Se esta tarefa não for executada, a meta não será atingida e a missão não terá sucesso.

A Figura 4.5 apresenta a árvore da palavra-guia “meta não atingida” correspondente à meta “fornecer dados inerciais”.

Figura 4.5 – Árvore da palavra-guia “meta não atingida” de “fornecer dados inerciais”.

Como exemplo de aplicação do processo ELICERE para a meta “fornecer dados inerciais”, é apresentado o resultado da aplicação do questionário relativo à palavra-guia de “meta não atingida”, Figura 4.6, e o questionário de dois de seus componentes: “recurso ausente”, Figura 4.7, e “omissão da tarefa”, Figura 4.8.

META: “fornecer dados inerciais”.

PALAVRA-GUIA: META NÃO ATINGIDA

Interpretação: A meta de fornecer os dados inerciais do voo do veículo não foi cumprida. Meta não alcançada.

Tipo da falha: Sistema.

Modo de falha:

SI1: Recurso ausente.

SI2: Recurso incorreto.

SI3: Recurso adicional.

SI4: Recurso fora do tempo/ordem.

SI5: Omissão da tarefa.

SI6: Incorreção na tarefa.

SI7: Término anormal da tarefa.

SI8: Tarefa fora do tempo/ordem.

Pior nível de perigo esperado: PERDA DA MISSÃO. A impossibilidade de serem fornecidos os dados inerciais do voo, em um dado instante, não permite a estabilização e condução do veículo durante o voo.

Severidade da falha: CRÍTICA.

Metas-soft

- **Risco Aceitável:** MÉDIO. Estabelecer na definição da meta, de forma qualitativa/ quantitativa, quais as condições/valores necessários para se atingir um risco aceitável médio.

- **Probabilidade Aceitável de Ocorrência:** REMOTA. Estabelecer na definição da meta, de forma qualitativa/ quantitativa, quais as condições/valores necessárias para se atingir uma probabilidade remota de ocorrência do risco.

- **Fator de Qualidade 1:** Avaliar se a meta, por meio da técnica GQM (Goal-Question-Metric), atende às expectativas do cliente e do usuário, dentro de critérios pré-estabelecidos (por exemplo, escala de 0 a 5 para a avaliação do número de versões do documento de definição da meta, sua complexidade, esforço para a meta ser alcançada, entendimento por parte dos envolvidos).

- **Fator de Qualidade 2:** Selecionar um conjunto correto de tarefas e recursos para que a meta seja alcançada (correção).

- **Fator de Qualidade 3:** Consistir a meta durante todo o desenvolvimento do sistema para que se possam evitar inconsistências (consistência).

Figura 4.6 – Questionário de “meta não atingida” de “fornecer dados inerciais”.

RECURSO: “deltaAlfa, deltaBeta, deltaEpsilon”.

PALAVRA-GUIA: RECURSO AUSENTE

Interpretação: Recurso incremento dos ângulos de atitude “deltaAlfa, deltaBeta, deltaEpsilon” não está disponível.

Tipo da falha: Hardware/Software.

Modo de falha

HW1: O hardware do SI não está disponível/funcionando.

HW2: O meio de transmissão do recurso não está disponível/funcionando.

HW3: O hardware do CD não está disponível/ funcionando.

SW1: O protocolo de transmissão do recurso não envia os valores de medidas dos incrementos dos ângulos de atitude.

Pior nível de perigo esperado: PERDA DA MISSÃO. A falta dos valores dos incrementos dos ângulos de atitude não permite que seja calculada corretamente a atitude inercial do veículo, levando-o fora de sua trajetória nominal.

Severidade da falha: CRÍTICA.

Metas-soft

- **Risco aceitável:** MÉDIO. Estabelecer na definição do recurso, de forma qualitativa/ quantitativa, quais as condições/valores necessários para se atingir um risco aceitável médio.

- **Probabilidade aceitável de ocorrência:** REMOTA. Estabelecer na definição do recurso, de forma qualitativa/ quantitativa, quais as condições/valores necessários para se atingir uma probabilidade remota de ocorrência do risco.

- **Fator de Qualidade 1:** Garantir que o hardware do SI e do CD fiquem disponíveis para fornecer as medidas de incremento dos ângulos de atitude durante o Tempo de Missão (TM) de 600 seg. dentro de uma probabilidade de 10^{-7} falhas por ano, considerando o modo de falha ausente (confiabilidade e disponibilidade do SI e do CD).

- **Fator de Qualidade 2:** Garantir que a Taxa de Ocorrência de Falha (ROCOF – *Rate of Failure Occurrence*) do meio de transmissão (cabo) seja de 2/100 (disponibilidade do meio de transmissão).

- **Fator de Qualidade 3:** Garantir que o protocolo do meio de transmissão recupere o recurso que ficou ausente durante o envio da informação (robustez).

- **Fator de Qualidade 4:** Garantir que a comunicação entre o SI e o CD não sofra interferências ambientais acima do previsto para o perfil da missão, considerando o modo de falha ausente (capacidade de sobrevivência).

Figura 4.7 – Questionário de “recurso ausente” de “deltaAlfa, deltaBeta, deltaEpsilon”.

TAREFA: “adquirir incrementos de ângulos de atitude”.

PALAVRA-GUIA: OMISSÃO DA TAREFA

Interpretação: A tarefa de adquirir os incrementos de ângulos de atitude do veículo não foi realizada.

Tipo da falha: Software

Modo de falha

SW1: A tarefa que adquire os valores dos incrementos dos ângulos de atitude do veículo não foi executada.

Pior nível de perigo esperado: PERDA DA MISSÃO. Os dados relativos aos incrementos de ângulos de atitude do veículo não são adquiridos, o que acarretará na impossibilidade de realizar o cálculo de posicionamento e controle do veículo durante seu voo.

Severidade da falha: CRÍTICA.

Metas-soft

- **Risco Aceitável:** MÉDIO. Estabelecer na definição da tarefa, de forma qualitativa/ quantitativa, quais as condições/valores necessárias para se atingir um risco aceitável médio.
- **Probabilidade Aceitável de Ocorrência:** REMOTA. Estabelecer na definição da tarefa, de forma qualitativa/ quantitativa, quais as condições/valores necessárias para se atingir uma probabilidade remota de ocorrência do risco.
- **Fator de Qualidade 1:** Garantir que a tarefa de aquisição dos dados dos ângulos de atitude seja executada durante a navegação do veículo, dentro de uma probabilidade aceitável de 1 falha a cada 300 segundos (confiabilidade, disponibilidade).
- **Fator de Qualidade 2:** Garantir a execução da tarefa no tempo máximo de 500 milisegundos antes do início da navegação e com a prioridade esperada (previsibilidade).
- **Fator de Qualidade 3:** Prover a execução da tarefa a despeito de condições adversas enfrentadas pelo sistema (capacidade de sobrevivência e tolerância à falha).

Figura 4.8 – Questionário de “omissão da tarefa” de “adquirir incrementos de ângulos de atitude”.

4.5.2 A aplicação de palavras-guia na meta “fornecer telemédidas”

A Figura 4.9 apresenta o modelo SR das relações de dependência entre o ator CD e o ator TE para atender a meta “fornecer telemédidas”. O atendimento incorreto desta meta implica também no atendimento incorreto da meta “fornecer dados segurança de voo” do relacionamento entre o ator TE e o ator SS, e da meta “fornecer dados análise pós-voo” do relacionamento entre o ator TE e o ator TS.

A meta “fornecer telemédidas” não será atingida se o recurso disponibilizado pelo ator CD não foi fornecido ao ator TE, ou algum problema relacionado às tarefas atribuídas para esta funcionalidade do sistema, de responsabilidade do CD.

Figura 4.9 – Modelo SR da meta “fornecer telemédidas”.

Para alcançar a meta “fornecer telemedidas”, é necessário que o recurso “dados voo”, contendo as informações extraídas dos sensores, dos cálculos executados e os comandos fornecidos pelo ator CD para os atuadores, esteja correto para ser enviado ao ator TE. Se este recurso não for enviado corretamente, a meta não será atingida e a análise pós-voo, e a segurança da missão ficará comprometida.

Para alcançar a meta “fornecer telemedidas” é necessário também que o ator CD execute diversas tarefas, entre elas a tarefa “adquirir e condicionar dados”. Se esta tarefa for executada com incorreções, a meta não será atingida com sucesso.

A Figura 4.10 apresenta a árvore da palavra-guia “meta incorreta” correspondente à meta “fornecer telemedidas”.

Figura 4.10 – Árvore da palavra-guia “meta incorreta” de “fornecer telemedidas”.

Como exemplo, é apresentado o resultado da aplicação do questionário relativo à palavra-guia de “meta incorreta”, Figura 4.11, e o questionário da palavra-guia “recurso incorreto”, Figura 4.12, e da palavra guia “incorreção na tarefa”, Figura 4.13.

META: “fornecer telemedidas”.

PALAVRA-GUIA: META INCORRETA

Interpretação: A meta de fornecer as telemedidas geradas durante o voo do veículo não foi atingida corretamente.

Tipo da falha: Sistema.

Modo de falha:

SI1: Recurso ausente.

SI2: Recurso incorreto.

SI3: Recurso adicional.

SI4: Recurso fora do tempo/ordem.

SI5: Omissão da tarefa.

SI6: Incorreção na tarefa.

SI7: Término anormal da tarefa.

SI8: Tarefa fora do tempo/ordem.

Pior nível de perigo esperado: DEGRADAÇÃO DA MISSÃO. O monitoramento realizado de forma incorreta não permite que se obtenham as informações do comportamento do voo a serem processados em terra.

Severidade da falha: MAIOR.

Metas-soft

- **Risco Aceitável:** MÉDIO. Estabelecer na definição da meta, de forma qualitativa/ quantitativa, quais as condições/valores necessários para se atingir um risco aceitável médio.

- **Probabilidade Aceitável de Ocorrência:** OCASIONAL. Estabelecer na definição da meta, de forma qualitativa/ quantitativa, quais as condições/valores necessárias para se atingir uma probabilidade ocasional de ocorrência.

- **Fator de Qualidade 1:** Avaliar se a meta, por meio da técnica GQM (Goal-Question-Metric), atende às expectativas do cliente e do usuário, dentro de critérios pré-estabelecidos (por exemplo, escala de 0 a 5 para a avaliação do número de incorreções observadas, erros encontrados nos testes, esforço para corrigir a meta, entendimento por parte dos envolvidos).

- **Fator de Qualidade 2:** Selecionar um conjunto correto de tarefas e recursos para que a meta seja alcançada corretamente (correção).

- **Fator de Qualidade 3:** Consistir a meta durante o desenvolvimento do sistema (definir revisões de modo que possam se evitar incorreções ou erros de interpretação) (consistência).

Figura 4.11 – Questionário de “meta incorreta” de “fornecer telemedidas”.

RECURSO: “dados voo”.

PALAVRA-GUIA: RECURSO INCORRETO

Interpretação: Fornecido recurso “dados voo” incorreto (valores maiores, menores, opostos, “parte de”, outros). O recurso fornecido está incorreto.

Tipo da falha: Software/ Hardware.

Modo de falha

HW1: O hardware do CD não está funcionando corretamente.

HW2: O meio de transmissão do recurso (cabo) não está funcionando corretamente.

SW1: O protocolo de transmissão do recurso dos dados de voo do CD para o TE não está correto.

Pior nível de perigo esperado: DEGRADAÇÃO DA MISSÃO. Os dados incorretos relativos às monitorações dos eventos de voo, dos tempos de acionamento dos eventos, entre outras informações fornecidas, não permitem a análise pós-voo do comportamento do foguete, não atendendo a meta “fornecer telemedidas”.

Severidade da falha: MAIOR.

Metas-soft

- **Risco aceitável:** MÉDIO. Estabelecer na definição do recurso, de forma qualitativa/quantitativa, quais as condições/valores necessários para se atingir um risco médio.

- **Probabilidade aceitável de ocorrência:** OCASIONAL. Estabelecer na definição do recurso, de forma qualitativa/ quantitativa, quais as condições necessárias para se atingir uma probabilidade ocasional de ocorrência.

- **Fator de Qualidade 1:** Garantir que o hardware do CD forneça corretamente os dados de voo durante o Tempo de Missão (TM) de 600 seg., dentro de uma probabilidade aceitável (ocasional) de falhas por demanda (confiabilidade).

- **Fator de Qualidade 2:** Garantir que o meio de transmissão (cabo) envie corretamente os valores dos dados de voo durante o Tempo de Missão (TM) dentro de uma probabilidade aceitável (ocasional) de falhas por demanda (confiabilidade).

- **Fator de Qualidade 3:** Garantir que o recurso fique integro durante o Tempo de Missão (TM) (correção).

- **Fator de Qualidade 4:** Verificar ao final de cada transmissão se o recurso foi entregue corretamente (consistência).

- **Fator de Qualidade 5:** Garantir que o protocolo do meio de transmissão entregue os dados de forma correta no Tempo de Missão (TM) de 600 segundos. (confiabilidade).

- **Fator de Qualidade 6:** Garantir que a comunicação entre o TE e o CD não sofra interferências ambientais acima do previsto para o perfil da missão (capacidade de sobrevivência).

Figura 4.12 – Questionário de “recurso incorreto” de “dados voo”.

TAREFA: “adquirir e condicionar dados”.

PALAVRA-GUIA: INCORREÇÃO NA TAREFA

Interpretação: A tarefa de adquirir e condicionar os dados do voo do veículo foi provida incorretamente.

Tipo da falha: Software

Modo de falha

SW1: A tarefa de software criada para adquirir e condicionar os dados de voo do veículo não está correta.

Pior nível de perigo esperado: DEGRADAÇÃO DA MISSÃO. A aquisição e o condicionamento incorreto dos dados não permitem a análise pós-voo do comportamento do foguete, não atendendo a meta “fornecer telemetrias”.

Severidade da falha: MAIOR.

Metas-soft

- **Risco aceitável:** MÉDIO. Estabelecer na definição da tarefa, de forma qualitativa/ quantitativa, quais as condições/ valores necessários para se atingir um risco aceitável médio.

- **Probabilidade aceitável de ocorrência:** OCASIONAL. Estabelecer na definição da tarefa, de forma qualitativa/ quantitativa, quais as condições necessárias para se atingir uma probabilidade ocasional de ocorrência.

- **Fator de Qualidade 1:** Garantir que a tarefa de adquirir e condicionar os dados do voo esteja livre de defeitos e seja executada corretamente (correção) dentro de uma probabilidade aceitável (ocasional) de falhas por demanda (POFOD – *Probability of Failure on Demand*)

- **Fator de Qualidade 2:** Verificar se a tarefa adquiriu os dados a uma taxa de 1000 bps e se foram corretamente convertidos para as devidas unidades de medida (precisão).

Figura 4.13. – Questionário de “incorreção na tarefa” de “adquirir e condicionar dados”.

4.5.3 A aplicação de palavras-guia na meta “preparar para vôo”

A Figura 4.14 apresenta o modelo SR das relações de dependência entre o ator CD e o ator AS para atender a meta “preparar para vôo”. O atendimento de forma incorreta desta meta significa que algum recurso disponibilizado ou tarefa realizada antes do lançamento do veículo para vôo não foram executadas corretamente, provocando a interrupção dos preparativos para o vôo.

O recurso “perfil da trajetória”, contendo os parâmetros relativos ao perfil do vôo do V-ALFA, deve ser carregado corretamente através do meio de comunicação entre o AS e o CD, na fase de pré-vôo. Se este recurso for fornecido de forma incorreta pelo AS, a meta não será atingida, pois o CD não carregará os dados com os valores especificados. Haverá a necessidade de se carregar este recurso novamente com os valores corrigidos.

Figura 4.14 – Modelo SR da meta “preparar para vôo”.

Já a tarefa “carregar perfil da trajetória”, função que recebe os dados relativos ao perfil da trajetória do voo proposto não for capaz de consistir e tratar os valores enviados pelo AS, a meta não será atingida e criará uma tabela incorreta do perfil da trajetória provocando a perda da missão.

A Figura 4.15 apresenta a árvore da palavra-guia “meta incorreta” correspondente à meta “preparar para voo”.

Figura 4.15 – Árvore da palavra-guia “meta incorreta” de “preparar para voo”.

Como exemplo, será apresentado o resultado da aplicação do questionário relativo à palavra-guia “meta incorreta”, Figura 4.16, e o questionário da palavra-guia “recurso incorreto”, Figura 4.17, e o da palavra-guia “incorreção na tarefa”, Figura 4.18.

META: “preparar para voo”.

PALAVRA-GUIA: META INCORRETA

Interpretação: A meta de executar a preparação para voo do veículo não foi cumprida corretamente. Meta não alcançada corretamente.

Tipo da falha: Sistema.

Modo de falha:

SI1: Recurso ausente.

SI2: Recurso incorreto.

SI3: Recurso adicional.

SI4: Recurso fora do tempo/ordem.

SI5: Omissão da tarefa.

SI6: Correção na tarefa.

SI7: Término anormal da tarefa.

SI8: Tarefa fora do tempo/ordem.

Pior nível de perigo esperado: PERDA DA MISSÃO. A execução incorreta da preparação para voo, ocasionará o carregamento de um perfil de trajetória anômalo que provocará um voo incorreto.

Severidade da falha: CRÍTICA.

Metas-soft

- **Risco Aceitável:** BAIXO. Estabelecer na definição da meta, de forma qualitativa/quantitativa, quais as condições/valores necessários para atingir um risco aceitável baixo.

- **Probabilidade Aceitável de Ocorrência:** INACREDITÁVEL. Estabelecer na definição da meta, de forma qualitativa/ quantitativa quais as condições/ valores necessários para se atingir uma probabilidade inacreditável de ocorrência do risco.

- **Fator de Qualidade 1:** Avaliar se a meta, por meio da técnica GQM (Goal-Question-Metric), atende às expectativas do cliente e do usuário, dentro de critérios pré-estabelecidos (por exemplo, escala de 0 a 5 para a avaliação do número de correções observadas, erros encontrados nos testes, esforço para corrigir a meta, entendimento por parte dos envolvidos).

- **Fator de Qualidade 2:** Selecionar um conjunto correto de tarefas e recursos para que a meta seja alcançada corretamente (correção).

- **Fator de Qualidade 3:** Consistir a meta durante o desenvolvimento do sistema para que possa se evitar correções (consistência).

Figura 4.16 – Questionário de “meta incorreta” de “preparar para voo”.

RECURSO: “perfil da trajetória”

PALAVRA-GUIA: RECURSO INCORRETO

Interpretação: RECURSO que contém os parâmetros de voo do V-ALFA foi criado de forma incorreta. Recurso oferecido em formato incorreto. Recurso com conteúdo não compatível com o correto.

Tipo da falha: Hardware/Software

Modo de falha

HW1: O hardware do AS não está funcionando corretamente.

HW2: O hardware do CD não está funcionando corretamente.

HW3: O meio de transmissão do recurso (cabo) não está funcionando corretamente.

SW1: O protocolo de comunicação entre o CD e o AS não está funcionando corretamente.

SW1: O recurso não possui o formato correto.

Pior nível de perigo esperado: QUALIDADE INSATISFATORIA.

A falta de valores corretos do perfil da trajetória do veículo, a serem fornecidos pelo AS, não permite o carregamento correto no CD.

Severidade da falha: MAIOR.

Metas-soft

- **Risco Aceitável:** MÉDIO. Estabelecer na definição do recurso, de forma qualitativa/ quantitativa, quais as condições/valores necessários para se atingir um risco aceitável médio.

- **Probabilidade Aceitável de Ocorrência:** OCASIONAL. Estabelecer na definição de recurso, de forma qualitativa/ quantitativa quais as condições/valores necessários para se atingir uma probabilidade ocasional de ocorrência do risco.

- **Fator de Qualidade 1:** Garantir que o hardware do AS fique disponível durante a transmissão do recurso (confiabilidade).

- **Fator de Qualidade 2:** Garantir que o hardware do CD fique disponível durante a recepção do recurso (confiabilidade).

- **Fator de Qualidade 3:** Garantir que o meio de transmissão (cabo) envie corretamente os dados do perfil da trajetória (correção) utilizando código verificador CRC (Cyclic Redundancy Check) (confiabilidade do meio de transmissão).

- **Fator de Qualidade 5:** Garantir que o recurso seja gerado de forma correta (correção).

Figura 4.17 – Questionário de “recurso incorreto” de “perfil de trajetória”.

TAREFA: “carregar perfil da trajetória”.

PALAVRA-GUIA: INCORREÇÃO NA TAREFA

Interpretação: A TAREFA responsável por carregar o perfil de trajetória no CD não está correta.

Tipo da falha: Software.

Modo de falha

SW1: A tarefa de software criada para receber comando do perfil da trajetória não está correta.

SW2: A tarefa de software criada para consistir comando do perfil da trajetória não está correta.

SW3: A tarefa de software criada para tratar comando do perfil da trajetória não está correta.

SW4: O perfil da trajetória não foi enviado corretamente pelo AS.

Pior nível de perigo esperado: PERDA DA MISSÃO. Os dados carregados de forma incorreta farão com o veículo seja alimentado com os parâmetros indevidos, provocando a perda da missão.

Severidade da falha: CRÍTICA.

Metas-soft

- **Risco Aceitável:** BAIXO. Estabelecer na definição da tarefa, de forma qualitativa/ quantitativa, quais as condições/valores necessários para se atingir um risco baixo.

- **Probabilidade Aceitável de Ocorrência:** INACREDITÁVEL. Estabelecer na definição da tarefa, de forma qualitativa/quantitativa, quais as condições necessárias para se atingir uma probabilidade inacreditável de ocorrência do risco.

- **Fator de Qualidade 1:** Garantir que a tarefa de carregar o perfil de trajetória do voo do veículo esteja livre de defeitos e seja executada corretamente dentro de uma probabilidade aceitável (inacreditável) de falhas por demanda (POFOD – *Probability of Failure on Demand*) de 0.001 (disponibilidade da tarefa).

- **Fator de Qualidade 2:** Garantir que a tarefa seja capaz de prever como são os dados que compõem o perfil de trajetória do veículo (previsibilidade), tamanho, intervalo, ordem de grandeza e consistir estas informações.

- **Fator de Qualidade 3:** Garantir que a tarefas de receber, consistir e tratar comando, bem como a de enviar resposta, carregue corretamente o perfil da trajetória (correção).

- **Fator de Qualidade 4:** Verificar ao final de cada transmissão se o recurso foi entregue corretamente (consistência).

Figura 4.18 – Questionário de “incorreção na tarefa” de “carregar perfil da trajetória”.

4.5.4 A aplicação de palavras-guia na meta “controlar atitude e rolamento 1E”

A Figura 4.19 apresenta o modelo SR das relações de dependência entre o ator CD e o ator TM1E a respeito da meta “controlar atitude e rolamento 1E”. O atendimento fora do tempo desta meta poderá provocar a perda da estabilidade do veículo ou o desvio de sua trajetória.

Esta meta não será atingida se os recursos disponibilizados pelo ator CD não forem fornecidos ao ator TM1E, ou ocorrer algum problema relacionado às tarefas atribuídas para esta funcionalidade do sistema, de responsabilidade do CD.

Figura 4.19 – Modelo SR da meta “controlar atitude e rolamento 1E”.

Para alcançar a meta “controlar atitude e rolamento 1E”, é necessário que o recurso que contém os ângulos de deflexão das tubeiras móveis esteja disponível no tempo correto. Estes valores são necessários para a devida correção da trajetória e da estabilização do veículo, através do sistema de tubeiras móveis, representado pelo ator TM1E. Se este recurso não for fornecido no tempo correto, a meta poderá não ser atingida e comprometer o desempenho da missão.

Para alcançar a meta “controlar atitude e rolamento 1E” é necessário também que a tarefa “selecionar sistema de atuação” esteja disponível para fornecer os ângulos de deflexão no tempo correto para o ator TM1E.

Esta tarefa pertence ao conjunto de tarefas da “controlar tubeira TM1E” executada pelo CD. Se esta tarefa não fornecer o recurso no tempo correto, a meta poderá não ser atingida e comprometer o desempenho da missão.

A Figura 4.20 apresenta a árvore da palavra-guia “meta fora do tempo/ordem”, correspondente a meta “controlar rolamento e atitude 1E”.

Como exemplo da aplicação do processo ELICERE, é apresentado o resultado da aplicação do questionário relativo à palavra-guia “meta fora do tempo”, Figura 4.21, o questionário da palavra-guia “recurso fora do tempo”, Figura 4.22, e o questionário da palavra-guia “tarefa fora do tempo/ordem”, como detalhado na Figura 4.23.

Figura 4.20 – Árvore da palavra-guia “meta fora do tempo/ordem” de “controlar atitude e rolamento 1E”.

META: “controlar atitude e rolamento 1E”.

PALAVRA-GUIA: META FORA DO TEMPO/ORDEM

Interpretação: A meta de controlar a atitude e rolamento do primeiro estágio foi obtida fora do tempo estabelecido.

Tipo da falha: Sistema.

Modo de falha:

SI1: Recurso fora do tempo/ordem.

SI2: Incorreção na tarefa.

SI3: Tarefa fora do tempo/ordem

Pior nível de perigo esperado: PERDA DO SISTEMA. O controle foi realizado no tempo incorreto, não permitindo que se obtenha e processe as informações para deflexão dos ângulos de tubeira do primeiro estágio no tempo correto. Não foi possível realizar o controle de veículo no tempo correto para sua condução e estabilização na trajetória nominal em um dado instante.

Severidade da falha: MAIOR.

Metas-soft

Risco Aceitável: MÉDIO. Estabelecer na definição da meta, de forma qualitativa/ quantitativa, quais as condições/valores necessários para se atingir um risco aceitável médio.

Probabilidade Aceitável de Ocorrência: OCASIONAL. Estabelecer na definição da meta, de forma qualitativa/quantitativa, quais as condições/valores necessárias para se atingir uma probabilidade ocasional de ocorrência do risco.

- **Fator de Qualidade 1:** Avaliar se a meta, por meio da técnica GQM (Goal-Question_Metric), atende às expectativas do cliente e do usuário, dentro de critérios pré-estabelecidos (por exemplo, valores -1, 0 ou +1 para a avaliação se atende a restrição do tempo, respectivamente com atraso, atende e adiantado, e valores de 0 a 5 para avaliação do número de incorreções observadas, erros encontrados nos testes, esforço para corrigir a meta, entendimento por parte dos envolvidos).

- **Fator de Qualidade 2:** Selecionar um conjunto correto de tarefas e recursos para que a meta seja alcançada (correção).

- **Fator de Qualidade 3:** Consistir a meta durante o desenvolvimento do sistema para que possam se evitar incorreções (consistência).

- **Fator de Qualidade 4:** Selecionar um conjunto correto de tarefas e recursos para que a meta seja alcançada sem resultados adicionais (correção).

- **Fator de Qualidade 5:** Garantir que a meta seja alcançada ciclicamente em um frequência máxima de 100Hz (desempenho).

Figura 4.21 – Questionário de “meta fora do tempo/ordem” de “controlar atitude e rolamento 1E”.

RECURSO: “TM1Ea, TM1Eb, TM1Ec, TM1Ed”.

PALAVRA-GUIA: RECURSO FORA DO TEMPO/ ORDEM

Interpretação: Os valores dos ângulos de deflexão do primeiro estágio do veículo não estão disponíveis no tempo estabelecido.

Tipo da falha: Hardware/ Software.

Modo de falha

HW1: O hardware do CD atrasou ao enviar os valores dos ângulos de deflexão da tubeira do primeiro estágio.

HW2: O hardware do TM1E não está funcionando corretamente.

SW1: O protocolo de comunicação entre o CD e o TM1E está entregando a informação fora do tempo previsto.

Pior nível de perigo esperado: PERDA DO SISTEMA. Os valores de ângulo de deflexão que foram fornecidos fora do tempo estabelecido não permitirão a sua estabilização e a correção da trajetória do veículo, em um dado instante.

Severidade da falha: MAIOR.

Metas-soft

Risco Aceitável: MÉDIO. Estabelecer na definição do recurso, de forma qualitativa/ quantitativa, quais as condições/valores necessários para se atingir um risco médio.

Probabilidade Aceitável de Ocorrência: OCASIONAL. Estabelecer na definição do recurso, de forma qualitativa/quantitativa, quais as condições/valores necessárias para se atingir uma probabilidade ocasional de ocorrência do risco.

- **Fator de Qualidade 1:** Garantir que o hardware do CD esteja disponível e possa fornecer o recurso dentro do tempo e da ordem prevista para o Tempo de Missão (TM) (confiabilidade e disponibilidade).

- **Fator de Qualidade 2:** Garantir que o hardware do TM1E esteja disponível e possa receber o recurso dentro do tempo e da ordem prevista para o Tempo de Missão (TM) de 600 segundos (confiabilidade e disponibilidade).

- **Fator de Qualidade 3:** Garantir que o protocolo de comunicação entre o CD e o TM1E mantenha uma taxa de atraso máximo (confiabilidade e disponibilidade).

Figura 4.22 – Questionário de “recurso fora do tempo/ordem” de “TM1Ea, TM1Eb, TM1Ec, TM1Ed”.

TAREFA: “selecionar sistema de atuação”.

PALAVRA-GUIA: TAREFA FORA DO TEMPO/ORDEM

Interpretação: A TAREFA de seleção do sistema de atuação das tubeiras foi executada fora do tempo previsto.

Tipo da falha: Software.

Modo de falha:

SW1: O software do CD que seleciona o sistema de atuação não fornece, no tempo correto, os ângulos de atuação das tubeiras do primeiro estágio.

Pior nível de perigo esperado: PERDA DO SISTEMA. A seleção dos dados do sistema de atuação é feita fora do tempo, não permitindo a correção da trajetória do veículo em um dado instante de tempo.

Severidade da falha: MAIOR.

Metas-soft

Risco Aceitável: MÉDIO. Estabelecer na definição da tarefa, de forma qualitativa/ quantitativa, quais as condições/valores necessários para se atingir um risco aceitável médio.

Probabilidade Aceitável de Ocorrência: OCASIONAL. Estabelecer na definição do recurso, de forma qualitativa/ quantitativa, quais as condições/valores necessárias para se atingir uma probabilidade ocasional de ocorrência do risco.

- **Fator de Qualidade 1:** Garantir que a tarefa de seleção do sistema de atuação execute no tempo correto (desempenho e tempo de resposta).
- **Fator de Qualidade 2:** Garantir o comportamento temporal da tarefa (previsibilidade).
- **Fator de Qualidade 3:** Garantir que a tarefa de seleção do sistema de atuação mantenha um comportamento temporal conforme o especificado para o Tempo de Missão (TM) de 600 segundos (desempenho e tempo de resposta).
- **Fator de Qualidade 4:** Garantir que a tarefa de seleção do sistema de atuação mantenha a capacidade de temporização e ordem para a correta seleção dos canais de acionamento das tubeiras (previsibilidade e eficiência de execução).

Figura 4.23 – Questionário de “tarefa fora do tempo/ordem” de “selecionar sistema de atuação”.

4.5.5 A aplicação de palavras-guia na meta “controlar rolamento 2E”

A Figura 4.24 apresenta a relação de dependência da meta “controlar rolamento 2E” entre o ator CD e o ator CR. O atendimento adicional desta meta implicará na perda do sistema, visto que a possível ativação das eletro-válvulas além do número de vezes necessário poderá fazer com que os fluidos responsáveis pelo acionamento do CR se esgotem e parem de controlar o veículo, em rolamento, precocemente.

Figura 4.24 – Modelo SR da meta “controlar rolamento 2E”.

Para alcançar a meta “controlar rolamento 2E” é necessário também que o recurso de acionamento dos motores (em inglês, *motor driver*) “MD1, MD2, MD3, MD4”, contendo os valores de ativação combinada das eletro-válvulas dos motores esteja correto. Se este recurso for fornecido de forma intempestiva (adicional), poderá provocar uma perda excessiva do conteúdo do reservatório do fluido, comprometendo irreversivelmente o controle do segundo estágio em rolamento.

Para alcançar a meta “controlar rolamento 2E” é necessário executar a tarefa “comandar atuadores”, que é a função que envia os valores combinados de ativação das tubeiras, Se esta tarefa for realizada mais vezes do que o necessário, a meta não será atingida.

A Figura 4.25 apresenta a árvore da palavra-guia “meta adicional” correspondente à meta “controlar rolamento 2E”.

Figura 4.25 – Árvore da palavra-guia “meta adicional” de “controlar rolamento 2E”.

Como exemplo, é apresentado o resultado da aplicação do questionário relativo à palavra-guia “meta adicional”, Figura 4.26, o questionário da palavra-guia “recurso adicional”, Figura 4.27, e “incorreção da tarefa”, Figura 4.28.

META: “controlar rolamento 2E”

PALAVRA-GUIA: META ADICIONAL

Interpretação: A meta de controlar rolamento do segundo estágio alcançou um resultado adicional (além do proposto).

Tipo da falha: Sistema.

Modo de falha:

SI1: Recurso incorreto.

SI2: Recurso adicional.

SI3: Incorreção na tarefa.

Pior nível de perigo esperado: DEGRADAÇÃO DA MISSÃO. O acionamento do controle de rolamento de forma adicional provocará um consumo excessivo do fluido, vindo a interromper seu funcionamento prematuramente, prejudicando o sistema.

Severidade da falha: MAIOR.

Metas-soft

Risco Aceitável: BAIXO. Estabelecer na definição da meta, de forma qualitativa/ quantitativa, quais as condições/valores necessários para se atingir um risco baixo.

Probabilidade Aceitável de Ocorrência: IMPROVÁVEL. Estabelecer na definição da meta, de forma qualitativa/ quantitativa, quais as condições/valores necessárias para se atingir uma probabilidade improvável de ocorrência do risco.

- **Fator de Qualidade 1:** Avaliar se a meta, por meio da técnica GQM (Goal-Question-Metric), atende às expectativas do cliente e do usuário, dentro de critérios pré-estabelecidos (por exemplo, valores de 0 a 5 para avaliação do número de incorreções observadas, erros encontrados nos testes, esforço para corrigir a meta, entendimento por parte dos envolvidos).

- **Fator de Qualidade 2:** Selecionar um conjunto correto de tarefas e recursos para que a meta seja alcançada (correção).

- **Fator de Qualidade 3:** Consistir a meta durante o desenvolvimento do sistema para que possam se evitar incorreções (consistência).

- **Fator de Qualidade 4:** Selecionar um conjunto correto de tarefas e recursos para que a meta seja alcançada sem resultados adicionais (correção).

Figura 4.26 – Questionário de “meta adicional” de “controlar rolamento 2E”.

RECURSO: “MD1, MD2, MD3, MD4”

PALAVRA-GUIA: RECURSO ADICIONAL

Interpretação: RECURSO fornecido além do estabelecido para o controle do 2E. Dados adicionais enviados.

Tipo da falha: Hardware/Software

Modo de falha

HW1: o hardware do CR não está funcionando corretamente, ativando as válvulas além do necessário.

SW1: o protocolo de comunicação entre o CD e o CR fornece valores adicionais aos necessários para o acionamento do sistema de controle de rolamento.

Pior nível de perigo esperado: DEGRADAÇÃO DA MISSÃO. Valores adicionais comandados para as válvulas não permitem que seja feita a correta estabilização em rolamento, provocando a perda do sistema.

Severidade da falha: MAIOR.

Metas-soft

Risco Aceitável: BAIXO. Estabelecer na definição do recurso de forma qualitativa/ quantitativa, quais as condições/valores necessários para se atingir um risco baixo.

Probabilidade Aceitável de Ocorrência: IMPROVÁVEL. Estabelecer na definição do recurso, de forma qualitativa/quantitativa, quais as condições/valores necessárias para se atingir uma probabilidade improvável de ocorrência do risco.

- **Fator de Qualidade 1:** Garantir que o recurso contenha valores corretos, não duplicados, nada mais do que o necessário, e dentro de um intervalo de tempo especificado (correção).

- **Fator de Qualidade 2:** Melhorar a capacidade de verificar, ao final de cada transmissão, se o recurso foi entregue corretamente (consistência).

Figura 4.27 – Questionário de “recurso adicional” de “MD1, MD2, MD3, MD4”.

TAREFA: “comandar atuadores”

PALAVRA-GUIA: INCORREÇÃO NA TAREFA

Interpretação: A TAREFA que comanda os atuadores do segundo estágio apresenta uma incorreção que faz com que seja fornecido um recurso adicional ao ator CR.

Tipo da falha: Software.

Modo de falha:

SW1: a tarefa de comandar atuadores está gerando recursos adicionais para movimentação no sistema de controle de rolamento do 2E.

Pior nível de perigo esperado: DEGRADAÇÃO DA MISSÃO. A rotina fornece dados adicionais que comprometem o funcionamento do sistema de controle de rolamento, desestabilizando o veículo no segundo estágio, ou provocando um consumo excessivo do reservatório de gás.

Severidade da falha: MAIOR.

Metas-soft

Risco Aceitável: BAIXO. Estabelecer na definição da tarefa de forma qualitativa/ quantitativa, quais as condições/valores necessários para se atingir um risco baixo.

Probabilidade Aceitável de Ocorrência: IMPROVÁVEL. Estabelecer na definição da tarefa de forma qualitativa/ quantitativa, quais as condições/valores necessárias para se atingir uma probabilidade improvável de ocorrência do risco.

- Fator de Qualidade 1: Garantir que a tarefa de comandar os atuadores do sistema execute sem enviar informações adicionais ou duplicadas (segurança).

Figura 4.28 – Questionário de “incorreção na tarefa” de “comandar atuadores”.

4.5.6 Aplicação de palavras-guia na meta “garantir segurança de vôo”

A Figura 4.29 apresenta o modelo SR das relações de dependência entre o ator SS e o ator Op_SS para atender a meta “garantir segurança de vôo”. O atendimento desta meta fora do tempo solicitado significa que a meta não foi atingida por problemas com o recurso “telecomando”, com o recurso “dados de segurança de vôo” ou por problemas com as tarefas atribuídas para esta atividade do sistema, como as relacionadas com “manter segurança de vôo”.

Figura 4.29 – Modelo SR da meta “garantir segurança de vôo”.

Um dos recursos necessários para alcançar a meta “destruir veículo” é o “telecomando”, que é diretamente ligado à meta “garantir segurança de vôo”. Este recurso contém o comando de radiofrequência de tele-destruição do veículo, que

deverá ser enviado no tempo correto para a interrupção do voo, em caso da violação de seu plano de segurança. O envio do recurso fora do tempo especificado não garantirá a segurança do voo, permitindo que o veículo possa se deslocar para uma área habitada, pondo em risco a vida de pessoas e bens materiais.

Para alcançar a meta “garantir segurança de voo” é necessário que o ator Op_SS execute um conjunto de tarefas, entre elas a “interromper voo”. Se esta tarefa não for executada no tempo correto, o envio do “telecomando” do ator SS para o ator CD não será feito no tempo correto, podendo conduzir a missão a um acidente catastrófico.

A Figura 4.30 apresenta a árvore da palavra-guia “meta fora do tempo/ordem”, correspondente à meta “garantir segurança de voo”.

Figura 4.30 – Árvore da palavra-guia “meta fora do tempo/ordem de “garantir segurança de voo”.

Como exemplo, é apresentado o resultado da aplicação do questionário relativo a palavra-guia “meta fora do tempo/ordem”, Figura 4.31, e o questionário da palavra-guia “recurso fora do tempo/ordem”, Figura 4.32, e o da palavra-guia “tarefa fora do tempo/ordem”, Figura 4.33.

META: “garantir segurança de voo”.

PALAVRA-GUIA: META FORA DO TEMPO/ORDEM

Interpretação: A META garantir segurança de voo do veículo foi atendida fora do tempo/ordem estabelecido.

Tipo da falha: Sistema.

Modo de falha:

SI1: Recurso fora do tempo/ordem.

SI2: Incorreção na tarefa.

SI3: Tarefa fora do tempo/ordem

Pior nível de perigo esperado: PERDA OU DANOS A VIDA HUMANA. No caso de violação do plano de segurança, o atraso na obtenção desta meta poderá provocar a saída do veículo da área de segurança e a queda de partes do foguete em área protegida.

Severidade da falha: CATASTRÓFICA.

Meta-soft

Risco Aceitável: BAIXO. Estabelecer na definição da meta, de forma qualitativa/ quantitativa, quais as condições/valores necessários para se atingir um risco aceitável baixo.

Probabilidade Aceitável de Ocorrência: INACREDITÁVEL. Estabelecer na definição da meta, de forma qualitativa/ quantitativa, quais as condições/valores necessárias para se atingir uma probabilidade inacreditável de ocorrência do risco.

- **Fator de Qualidade 1:** Avaliar se a meta, por meio da técnica GQM (Goal-Question-Metric), atende às expectativas do cliente e do usuário, dentro de critérios pré-estabelecidos (por exemplo, valores -1, 0 ou +1 para a avaliação se atende a restrição do tempo, respectivamente com atraso, atende e adiantado, e valores de 0 a 5 para avaliação do número de incorreções observadas, erros encontrados nos testes, o esforço para atender o tempo de atendimento da meta, e os procedimentos operacionais para sua execução atende as normas de segurança).

- **Fator de Qualidade 2:** Selecionar um conjunto correto de tarefas e recursos para que a meta seja alcançada (correção).

- **Fator de Qualidade 3:** Consistir a meta durante o desenvolvimento do sistema para de modo que possam se evitar incorreções (consistência).

- **Fator de Qualidade 4:** Idealizar uma IHM (Interface Homem-Máquina) adequada para a comunicação com o operador (ergonomia).

- **Fator de Qualidade 5:** Garantir que a operação de abortar a missão seja realizada dentro de critérios de segurança da interface.

- **Fator de Qualidade 6:** Garantir que as informações dos dados de voo auxiliem o operador na tomada de decisão correta e no tempo previsto (compreensibilidade).

Figura 4.31 – Questionário de “meta fora do tempo/ordem” de “garantir segurança de voo”.

RECURSO: “telecomando”

PALAVRA-GUIA: RECURSO FORA DO TEMPO/ ORDEM

Interpretação: O recurso telecomando não estava disponível no tempo correto.

Tipo da falha: Hardware.

Modo de falha

HW1: O meio de transmissão do telecomando do SS para o CD não enviou a informação a tempo para que seja acionada a destruição do veículo.

HW2: O hardware do SS falhou ao tentar enviar o recurso no tempo correto.

HW3: O hardware do CD falhou ao adquirir o recurso no tempo correto.

Pior nível de perigo esperado: PERDA OU DANOS A VIDA HUMANA. O atraso no envio do comando de destruição possibilitará o encaminhamento do veículo para uma área protegida, provocando sua destruição tardia e a possibilidade de queda de partes do foguete em território habitado.

Severidade da falha: CATASTRÓFICA.

Meta-soft

Risco Aceitável: BAIXO. Estabelecer na definição do recurso, de forma qualitativa/ quantitativa, quais as condições/valores necessários para se atingir um risco baixo.

Probabilidade Aceitável de Ocorrência: INACREDITÁVEL. Estabelecer na definição do recurso, de forma qualitativa/quantitativa, quais as condições/valores necessárias para se atingir uma probabilidade inacreditável de ocorrência do risco.

- **Fator de Qualidade 1:** Garantir que o hardware do SS esteja disponível e possa enviar o recurso dentro do tempo/ordem prevista para o Tempo de Missão de 600 segundos.(TM) (disponibilidade).

- **Fator de Qualidade 2:** Garantir que o hardware do CD esteja disponível e possa receber o recurso dentro do tempo/ordem prevista para o Tempo de Missão (TM) de 600 segundos (disponibilidade).

- **Fator de Qualidade 3:** Garantir que o meio de transmissão (cabo) envie o telecomando durante o tempo de Missão (TM) dentro de uma probabilidade de Falha Sob Demanda (POFOD – *Probability of Failure on Demand*) de 0.001 (disponibilidade da tarefa) de 0.000001.

Figura 4.32 – Questionário de “recurso fora do tempo/ordem” de “telecomando”.

TAREFA: “interromper voo”

PALAVRA-GUIA: TAREFA FORA DO TEMPO/ORDEM

Interpretação: A tarefa responsável por interromper o voo do foguete, no caso da violação do plano de segurança não foi acionada no tempo correto.

Tipo da falha: Humana.

Modo de falha:

HU1: Não foi acionada no tempo correto pelo Op_SS a ordem de destruição do veículo.

Pior nível de perigo esperado: PERDA OU DANOS A VIDA HUMANA. O atraso no envio da ordem de destruição do veículo possibilitará o encaminhamento do veículo para uma área protegida, provocando sua destruição tardia e a possibilidade de queda de partes do foguete em território habitado.

Severidade da falha: CATASTRÓFICA.

Meta-soft

Risco Aceitável: BAIXO. Estabelecer na definição da tarefa, de forma qualitativa/ quantitativa, quais as condições/valores necessários para se atingir um risco aceitável baixo.

Probabilidade Aceitável de Ocorrência: INACREDITÁVEL. Estabelecer na definição da tarefa, de forma qualitativa/quantitativa, quais as condições/valores necessárias para se atingir uma probabilidade inacreditável de ocorrência do risco.

- **Fator de Qualidade 1:** Avaliar e interpretar a meta dentro de critérios pré-estabelecidos
- **Fator de Qualidade 2:** Selecionar um conjunto correto de tarefas e recursos para que a meta seja alcançada (correção).
- **Fator de Qualidade 3:** Consistir a meta durante o desenvolvimento do sistema para de modo que possam se evitar incorreções (consistência).
- **Fator de Qualidade 4:** Garantir que o procedimento operacional de interromper o voo seja executado de forma a garantir o envio do telecomando em tempo de destruir o foguete (desempenho do Operador).
- **Fator de Qualidade 5:** Garantir que o procedimento de interromper o voo seja executado com segurança dentro de um projeto que facilite o operador a tomar decisões críticas, utilizando critérios rigorosos de usabilidade (usabilidade).

Figura 4.33 – Questionário de “tarefa fora do tempo/ordem” de “interromper voo”.

4.5.7 Aplicação de palavras-guia na meta “habilitar modo de operação”

A Figura 4.34 apresenta o modelo SR das relações de dependência entre o ator Op_AS e o ator AS para atender a submeta “habilitar modo de operação”, da meta “enviar comandos Op_AS”.

Uma habilitação incorreta do modo de operação significa que os recursos disponibilizados pelo ator Op_AS não estão corretos ou algum problema relacionado às tarefas do ator AS atribuídas a esta funcionalidade não permitiram que a meta fosse atingida corretamente.

Figura 4.34 – Modelo SR da meta “habilitar modo de operação”.

Para alcançar a meta “habilitar modo de operação” é necessário que o ator AS execute uma série de tarefas, entre elas a tarefa “tratar interrupção da sequência automática”. Se esta tarefa tiver incorreções, a meta não será atingida e as tarefas

de preparação para o voo poderão atrasar ou até mesmo provocar danos a equipamentos a bordo do veículo.

A Figura 4.35 apresenta a árvore da palavra-guia “meta não atingida” correspondente à meta “habilitar modo de operação”.

Figura 4.35 – Árvore da palavra-guia “meta não atingida” de “habilitar modo de operação”.

Como exemplo, é apresentado o resultado da aplicação do questionário relativo a palavra-guia “Meta Não Atingida”, Figura 4.36, e o questionário da palavra-guia “incorreção na tarefa”, Figura 4.37.

META: “habilitar modo de operação”.

PALAVRA-GUIA: META NÃO ATINGIDA

Interpretação: A META de habilitar o modo de operação do veículo durante a fase de pré-vôo não foi cumprida. Meta não alcançada.

Tipo da falha: Sistema.

Modo de falha:

SI1: Recurso ausente.

SI2: Recurso incorreto.

SI3: Recurso adicional.

SI4: Recurso fora do tempo/ordem.

SI5: Omissão da tarefa.

SI6: Incorreção na tarefa.

SI7: Término anormal da tarefa.

SI8: Tarefa fora do tempo/ordem.

Pior nível de perigo esperado: QUALIDADE INSATISFATÓRIA. A impossibilidade de habilitar corretamente o modo de operação provocará a interrupção da missão durante a fase de pré-vôo.

Severidade da falha: MENOR.

Metas-soft

- **Risco Aceitável:** BAIXO. Estabelecer na definição da meta, de forma qualitativa/ quantitativa, quais as condições/valores necessários para se atingir um risco aceitável baixo.

- **Probabilidade Aceitável de Ocorrência:** REMOTA. Estabelecer na definição da meta, de forma qualitativa/ quantitativa, quais as condições/valores necessárias para se atingir uma probabilidade remota de ocorrência do risco.

- **Fator de Qualidade 1:** Avaliar se a meta, por meio da técnica GQM (Goal-Question-Metric), atende às expectativas do cliente e do usuário, dentro de critérios pré-estabelecidos (por exemplo, valores de 0 a 5 para avaliação do número de incorreções observadas, erros encontrados nos testes, esforço para corrigir a meta, entendimento por parte dos envolvidos)..

- **Fator de Qualidade 2:** Selecionar um conjunto correto de tarefas e recursos para que a meta seja alcançada (correção).

- **Fator de Qualidade 3:** Consistir a meta durante o desenvolvimento do sistema para de modo que possam se evitar incorreções (consistência).

continua

continuação

- **Fator de Qualidade 4:** Idealizar uma IHC (Interface-Homem-Computador) padrão.
- **Fator de Qualidade 5:** Garantir que a operação de abortar a missão seja realizada dentro de critérios de segurança da interface.
- **Fator de Qualidade 6:** Manter a interação homem-máquina amigável e eficiente (ergonomia).
- **Fator de Qualidade 7:** Garantir que o comando enviado para habilitar um determinado modo de operação seja suficiente e não permita ao operador comandar modos diferentes.
- **Fator de Qualidade 8:** Prover ao operador *feedback* de cada operação efetuada na transferência do modo de operação
- **Fator de Qualidade 9:** Garantir que a meta seja eficiente para alcançar seus objetivos (eficiência).

Figura 4.36 – Questionário de “meta não atingida” de “habilitar modo de operação”.

TAREFA: “tratar interrupção da sequência automática”.

PALAVRA-GUIA: INCORREÇÃO NA TAREFA

Interpretação:

Interpretação: A TAREFA que trata a interrupção da sequência automática de preparação para voo pelo AS não funcionou corretamente.

Tipo da falha: Software.

Modo de falha:

SW1: a incorreção na tarefa de tratamento da interrupção da sequência automática não permitiu que fossem executados os procedimentos previstos para o encerramento das rotinas de pré-voo.

Pior nível de perigo esperado: QUALIDADE INSATISFATÓRIA. Um erro na rotina de interrupção da sequência automática provocou a não interrupção da sequência automática, obrigando ao operador desligar o sistema manualmente.

Severidade da falha: MENOR.

Metas-soft

Risco Aceitável: BAIXO. Estabelecer na definição da tarefa de forma qualitativa/ quantitativa, quais as condições/valores necessários para se atingir um risco baixo.

Probabilidade Aceitável de Ocorrência: REMOTA. Estabelecer na definição da tarefa de forma qualitativa/ quantitativa, quais as condições/valores necessárias para se atingir uma probabilidade remota de ocorrência do risco.

- **Fator de Qualidade 1:** Garantir que a tarefa de tratamento da interrupção da sequência automática possa ser executada sem danificar equipamentos de voo (segurança).

- **Fator de Qualidade 1:** Garantir que a tarefa de tratamento da interrupção da sequência automática desabilite os equipamentos de solo de modo a permitir sua reativação de modo rápido (recuperabilidade).

Figura 4.37 – Questionário de “incorreção na tarefa” de “tratar interrupção da sequência automática”.

4.6 CONSIDERAÇÕES

Através do estudo de caso pode-se acompanhar a aplicação do processo ELICERE desde a representação do sistema computacional embarcado do veículo V-ALFA e de suas interfaces com os sistemas em solo, passando pela aplicação de palavras-guias e o preenchimento dos questionários - até a definição de metas-soft para a melhoria da dependabilidade do sistema.

Os casos ilustrativos escolhidos para exemplificar a aplicação do processo abrangeram atores relacionados ao controle do veículo (Sensor Inercial, Controlar Digital, Tubeira Móvel e Controle de Rolamento), como outros atores embarcados que se comunicam com o Controlador Digital (Segurança e Atuação, Sensor de Pressão, Telemetria Embarcada) e ainda atores de solo (Telemetria em Solo, Apoio em Solo e Segurança em Solo). Foram incluídos também casos envolvendo atores humanos e suas relações de dependência com o sistema (Operador de Segurança e Operador de Apoio em Solo).

Primeiramente, foram desenvolvidos os diagramas SD de metas e SD de recursos do V-ALFA, cujos modelos foram extraídos da descrição textual dos objetivos de missão apresentados na Seção 4.1 deste capítulo. Durante a confecção destes dois modelos foram identificadas diversas relações de dependência entre metas, recursos e tarefas que não tinham sido detalhados na descrição dos objetivos do sistema e que são necessários para satisfazer a relação entre os atores. Mais especificamente, foi identificada a relação entre o ator CD e o ator SS, através das metas “fornecer dados segurança de vôo” e “destruir veículo”, bem como dos recursos “dados segurança vôo” e “telecomando”. Estas metas implicam em uma série de informações que precisam ser discutidas com mais detalhe no projeto do V-ALFA, mas já é possível observar sua importância com relação à dependabilidade na construção do sistema.

Procurou-se descrever neste capítulo casos de palavras-guias associadas aos componentes do sistema que apresentassem níveis de perigo e severidades variadas, a fim de exemplificar melhor o processo ELICERE.

Por exemplo, o componente meta “fornecer dados inerciais” aborda um dos objetivos mais sensíveis em termos de controle do veículo, pois se esta meta não for atingida, o CD não poderá processar as informações inerciais e conduzir o veículo dentro da trajetória planejada. Este exemplo foi escolhido por apresentar um perigo que acarreta a “perda da missão” e, portanto, de severidade “crítica” para o sistema. Foi atribuída uma meta-soft que estabelece um risco aceitável “médio” e uma probabilidade de ocorrência “remota”. A atribuição de “remota” pode estar associada a uma probabilidade de ocorrência de acidente envolvendo hardware da ordem de 10^{-3} falhas por ano, valor que deve ser definido pela equipe de engenharia de sistemas do projeto.

Mais severo ainda seria se o componente recurso “telecomando” fosse fornecido fora do tempo estipulado pelo sistema, provocando, por exemplo, um acionamento tardio da destruição do veículo, no caso do mesmo se encaminhar para uma região protegida do voo, conforme definido pelo sistema de Segurança em Solo. Este tipo de perigo pode acarretar a queda de partes do foguete em região habitada, podendo ocasionar até a “perda ou danos à vida humana”, portanto de severidade “catastrófica” para o sistema. Foi atribuída uma meta-soft que estabelece um risco aceitável “baixo” e uma probabilidade de ocorrência “inacreditável”. A atribuição de “inacreditável” pode estar associada a uma probabilidade de ocorrência de acidente envolvendo hardware da ordem de 10^{-7} falhas por ano, valor que deve ser definido pela equipe de engenharia de sistemas do projeto.

Também foi utilizado um exemplo de aplicação de palavras-guia a uma incorreção na tarefa de “tratar interrupção da seqüência automática”, que produziria um programa de software incorreto para a interrupção dos passos referentes aos preparativos para lançamento do veículo. Como a interrupção da seqüência automática pode ser feita também manualmente pelo Operador de Apoio em Solo (Op_AS), a severidade é “menor” e produziria uma tarefa de “qualidade insatisfatória” para o sistema. Foi atribuída uma meta-soft que estabeleça um risco aceitável “baixo” e uma probabilidade de ocorrência “remota”.

Outra utilidade para a combinação da resposta do item “pior nível de perigo esperado”, que descreve a conseqüência do não atendimento da palavra-guia do

componente analisado, com o item “severidade” do questionário, é ajudar na identificação e na definição das metas-soft de “risco aceitável” e de “probabilidade aceitável de ocorrência do risco”. Com esta combinação já é possível classificar e priorizar os componentes do sistema que devem ser cuidadosamente analisados e atendidos com relação à melhoria da dependabilidade, e que possuir um risco tolerável para o componente do sistema, seja meta, recurso ou tarefa. Níveis de perigo relacionados à perda ou danos à vida humana, bem como perda de missão, à propriedade e ao ambiente são naturalmente fortes candidatos à priorização, principalmente se possuírem severidades catastróficas ou críticas.

Para cada severidade e risco associado foi definida uma meta-soft de probabilidade aceitável de ocorrência do risco, baseada na Tabela 4.3, onde para o V-ALFA se estabeleceu adotar o valor mais tolerante da faixa de ocorrência aceitável. Isto significa que se a severidade for “menor” e o risco aceitável for “médio”, a probabilidade aceitável de ocorrência do risco será “provável” e não “ocasional”.

A definição de metas-soft para o componente “meta” pode estar associada a algum tipo de métrica como proposta pela técnica de Meta- Questão- Métrica (GQM) (da sigla em inglês, *Goal-Question-Metric*) descrita por Basili (2005). Através da identificação de alguma medida para se acompanhar, avaliar e melhorar metas são objetivos que devem ser alcançados em projetos de sistemas críticos. As metas (ou objetivos) são definidas em termos de seu propósito, atributos de qualidade e ambiente, e cada objetivo é refinado em um conjunto de perguntas que representam uma definição operacional do objetivo. Para cada pergunta, as métricas relevantes são atribuídas a fim de avaliar se estes objetivos foram alcançados.

Já as metas-soft identificadas como do conjunto de fatores capacidade de defesa, de adequação e de qualidade devem ser simples e objetivas, o quanto for possível, a fim de auxiliar na futura definição de requisitos não funcionais do sistema.

Como exemplo de definição de meta-soft qualitativa e quantitativa, observa-se no exemplo da Figura 4.9, o recurso “dados segurança voo”, que é o conjunto de informações necessárias para que o sistema de Segurança em Solo (SS) acompanhe a trajetória do veículo durante seu voo. Se este recurso, por algum

motivo, não estiver presente ciclicamente (palavra-guia “recurso ausente”) para que o SS apresente a evolução do voo ao Operador do sistema de Segurança em Solo (Op_SS), a meta “garantir segurança de voo” não será atingida, colocando em risco a segurança da operação de lançamento do veículo.

Neste caso será necessário definir metas-soft qualitativas endereçadas ao recurso “dados segurança voo” como “garantir robustez contra perturbações ambientais” (fator robustez) ou “disponibilizar informação alternativa da trajetória real do voo” (fator capacidade de sobrevivência).

Paralelamente ao trabalho de se mapear metas em requisitos não-funcionais qualitativos, é desejável quantificar estas metas, de modo a se dimensionar um valor a ser definido na especificação. Esta quantificação poderá facilitar a definição prática e objetiva de um requisito, para aqueles que forem especificar, projetar e desenvolver um sistema computacional crítico.

Em Loucopoulos e Karakostas (1995) é citada a abordagem de derivação de requisitos não funcionais, mostrando um exemplo relacionado ao controle de tráfego aéreo, onde metas abstratas devem motivar requisitos de sistema que serão realizadas em tempo real. Requisitos não funcionais como desempenho e capacidade devem ser detalhados até que se consiga obter uma afirmação específica do que se espera do seu atendimento. No caso do exemplo proposto, a meta “os dados do radar devem ser apresentados em tempo real” (meta-soft qualitativa), motiva um requisito não funcional mais detalhado como “a posição da aeronave deve ser mostrada em menos do que 0.165 segundos” (meta-soft quantitativa).

Este grau de objetividade de um requisito deve ser perseguido nos sistemas críticos, de forma a se conseguir, em última análise, sua verificação, por exemplo, através de testes. Uma forma de melhorar a objetividade dos requisitos é através da identificação de suas propriedades mensuráveis. Exemplos de propriedades mensuráveis são mostrados na Tabela 4.4.

Tabela 4.4 – Propriedades mensuráveis relacionadas a requisitos não funcionais.

Propriedade	Métrica
Desempenho	Transações processadas por segundo
	Tempo de resposta por entrada de usuário
Confiabilidade	Taxa de ocorrência da falha
	Tempo entre falhas
Disponibilidade	Probabilidade de falha sob demanda
Tamanho	Kbytes
Usabilidade	Tempo para se aprender 80% das facilidades
	Número de erros cometido pelo usuário em um dado período de tempo
Robustez	Tempo para reiniciar após uma falha do sistema
Portabilidade	Número de sistemas alvo

Fonte: (KOTONYA; SOMMERVILLE, 2002).

.Apesar de discutível até que ponto é possível ser objetivo em relação a metas e requisitos, já foram apresentadas propostas, como a de Sommerville (2001), que sugere que este problema só poderá ser ultrapassado se um requisito não funcional satisfizer dois atributos: objetividade e capacidade de ser testado.

No caso do componente recurso “dados segurança vôo”, um exemplo de meta-soft quantitativa que poderia ser definida é a de “garantir que a Probabilidade de Falha Sob Demanda seja de 0.001”, ou seja, só poderá ocorrer uma perda de pacote de dados de vôo a cada 1000 pacotes enviados. (fator confiabilidade – POFOD) (sigla do inglês, *Probability of Failure On Demand*).

5 CONSIDERAÇÕES FINAIS

Esta tese de doutorado intitulada “ELICERE: o Processo de Elicitação de Metas de Dependabilidade para Sistemas Computacionais Críticos” propõe um conjunto mínimo de atividades e recursos necessários para a identificação precoce de metas de dependabilidade em sistemas computacionais de natureza crítica.

A utilização do processo auxilia de forma significativa o gerenciamento da qualidade em projetos de desenvolvimento de sistemas intensivos de software, principalmente no que tange à identificação de requisitos não funcionais do sistema. O processo ELICERE propõe uma nova forma de conduzir a identificação e o tratamento de questões como segurança, confiabilidade e disponibilidade em sistemas críticos. Ainda, o processo estimula a comunicação entre engenheiros de sistemas e de software através da utilização de elementos representativos do sistema que são de fácil entendimento e palavras-guia que conduzem à análise de perigos de forma intuitiva e compreensível.

Para se alcançar estes resultados foram necessários o planejamento, a definição e a modelagem das atividades do processo seguido das caracterizações dos recursos necessários à sua condução. As caracterizações de meta, recurso, tarefa, meta-soft, palavra-guia e fatores de qualidade são subprodutos de valor considerável da pesquisa que justificam todo o trabalho desenvolvido e, conseqüentemente, a relevância desta tese.

Para construção do processo foi necessário o estudo das práticas de engenharia de requisitos, das atividades de análise de perigos, o estudo de fatores da qualidade voltado para sistemas críticos, bem como a escolha de uma linguagem de modelagem que representasse adequadamente o sistema, e uma nova forma de abordar as questões de dependabilidade.

Em virtude do caráter específico do processo, foi necessário aprimorar o conhecimento sobre sistemas críticos, em particular os de um sistema espacial, a fim

de exemplificar, de maneira simples e didática, a condução da aplicação do ELICERE em um estudo de caso. Mais do que validar o sistema ou o projeto como um todo, o estudo de caso tem o propósito de mostrar, passo a passo, a aplicação do processo ELICERE, seus benefícios, suas eventuais fragilidades e os desafios para os trabalhos futuros.

Na seção 5.1 discute-se o resultado do trabalho sob uma perspectiva conceitual, mostrando-se como foram utilizadas as técnicas i^* e as palavras-guia para o ELICERE. Na seção 5.2 são apresentadas algumas considerações do trabalho sob a perspectiva prática, isto é, os desdobramentos do estudo de caso V-ALFA. Na seção 5.3 são introduzidas algumas contribuições da tese sob a ótica de processo, tecnologia e pessoas. Na seção 5.4 são abordados os trabalhos futuros que poderão ser conduzidos para a melhoria do processo e sua implantação prática nas organizações, e na seção 5.5 são tecidas as conclusões finais.

5.1 ABORDAGEM CONCEITUAL DO PROCESSO ELICERE

Quando as pesquisas da disciplina de engenharia de requisitos se voltaram para as técnicas orientadas a metas, foi estabelecido um campo distinto de investigação e prática. Tomando como base os trabalhos de levantamento, avaliação e discussão das abordagens GORE apresentados por diversos autores (KAVAKLI; LOUCOPOULOS, 1999) (VAN LAMSWEERDE, 2001) (LAPOUCHNIAN, 2005), classifica-se uma abordagem orientada a metas segundo sua visão de representação de como essas metas podem ser expressas. Elas podem ser expressas em uma variedade de formatos, usando mais ou menos notações formalísticas, sendo assim classificadas como não formal, semi-formal e formal.

Segundo Kavakli e Loucopoulos (2003), a abordagem informal geralmente usa uma linguagem de texto para expressar metas; a semi-formal usa, em sua maioria, diagrama de blocos e setas; e a abordagem formal expressa metas através de uma lógica assertiva utilizando alguma linguagem de especificação formal.

A falta de um conhecimento de técnicas formais, adotadas pelas técnicas apresentadas no capítulo 1 como o estado da arte nesta área - metodologia KAOS, estrutura NFR e a abordagem i^* do RESCUE - pode ser um obstáculo para a utilização, se quem exerce este papel na organização não possui uma formação em métodos formais. Os modelos formais resultantes do uso destas metodologia não são, muitas vezes, de representação simples e intuitiva, dificultando o diálogo entre engenheiros de sistemas e de software, quando da negociação das metas do sistema.

O processo ELICERE se posiciona então, de forma particular para a identificação de metas-soft, possuindo como principal diferencial o fato de ser usado para o entendimento e identificação de fatores que promovam a melhoria da dependabilidade em um sistema crítico. Estes fatores podem variar desde aqueles de mais alto nível de um sistema, até aqueles que afetam diretamente as metas de cada um de seus componentes, quer seja de software ou hardware. O ELICERE pode ser classificado como um processo de representação não formal dos modelos de um sistema crítico, com especificação qualitativa de suas metas, e passos bem definidos para sua realização. O processo propõe-se a investigar os perigos relacionados aos recursos, tarefas e metas do sistema e suas possíveis conseqüências, por meio de questionários utilizando uma linguagem de fácil interpretação por parte do engenheiro de requisitos, não necessitando de conhecimento prévio de modelagem de sistemas.

Na atividade onde se aplica análise de perigos, a finalidade dos questionários é de agrupar as principais informações que conduzam à identificação de metas-soft, de forma a se obter evidência estatística das situações de perigo do sistema e assim, estabelecerem-se prioridades para incorporação das metas-soft mais urgentes. É importante destacar que a aplicação dos questionários para a análise dos modelos de dependência de metas e recursos do sistema computacional pode ser feita de forma *top-down*, enquanto a análise dos modelos de dependência de tarefas pode ser feito de forma *bottom-up*, conforme a visão que se pretende ter do sistema. Conseqüentemente, as metas-soft extraídas diretamente de metas serão definidas com um nível de abstração mais alto do que as metas-soft extraídas de recursos e tarefas. As metas-soft de recursos, por exemplo, terão uma definição de metas de

dependabilidade no nível de dados ou de componentes eletrônicos, as tarefas, metas de dependabilidade no nível de programas de software ou funções de hardware a serem cumpridas.

Uma das questões surgidas durante o desenvolvimento da tese foi qual seria a diferença entre identificar meta e identificar requisito. Uma meta é geralmente alguma propriedade do sistema em um alto nível de abstração e um requisito deve possuir características objetivas e testáveis (KOTONYA; SOMMERVILLE, 2000). Acredita-se que na fase inicial da construção do sistema, quando ainda está sendo definido seu desenho, muitas vezes não é possível propor requisitos, e sim um padrão de objetivos (metas) entre os componentes do sistema e as necessidades às quais estes objetivos devem atender (requisitos). Nesta fase do desenvolvimento do projeto não se deve desprezar a utilidade de uma meta, dentro de uma abordagem sistemática, para descobrir e estruturar requisitos. No caso específico da meta-soft, em uma abordagem mais voltada para o cliente, não há ainda compromisso técnico de solução. Já o requisito não funcional tem mais a conotação de característica técnica de projeto, apresentado de forma a facilitar a implantação da solução.

As metas-soft devem atuar de forma a minimizar possíveis riscos associados ao componente do sistema analisado, isto é, devem definir medidas que garantam que os riscos e perigos associados àquele componente não interfiram na confiança no funcionamento do sistema.

5.2 ABORDAGEM PRÁTICA DO PROCESSO ELICERE

No estudo de caso proposto como exemplo da aplicação do processo ELICERE, a modelagem do sistema V-ALFA em um alto nível de abstração foi feita empregando-se o diagrama de Dependências Estratégicas (SD). Este diagrama representou tanto as metas como os recursos necessários para se atingir tais metas. O diagrama de Razões Estratégicas (SR) foi adotado para representar as tarefas já possíveis de serem identificadas nesta fase do projeto, e necessárias para atingir as metas com os respectivos recursos definidos por meio dos modelos SD. Com este tipo de

organização dos diagramas i* é possível criar visões dos atores e suas relações de dependência com os elementos de representação: visão inter-atores representando o diagrama SD de atores-metas e o de atores-recursos, e visão intra-atores representando o diagrama SR de atores-tarefas. A visão distinta de cada elemento da representação i* em modelos separados tem a intenção de auxiliar a identificação dos elementos mais críticos do modelo a serem estudados e, com isso, priorizá-los para aplicação do questionário e das palavras-guia pertinentes para a análise.

Não obstante a facilidade de transcrição das metas textuais do sistema V-ALFA para os diagramas SD e SR, pode-se observar que este processo foi utilizado de modo recorrente. Na atividade de criar os modelos do sistema foi necessário definir novos componentes (tarefa, por exemplo) de modo a se atender as metas com mais completude. Foi o caso de todos os diagramas SR do V-ALFA, cujas tarefas tiveram que ser definidas após um levantamento mais detalhado das possíveis (e prováveis) funcionalidades dos atores envolvidos. Cada tarefa criada para atender uma meta do sistema precisou ser cuidadosamente pesquisada e pormenorizada de forma a se criar, o mais cedo possível, uma visão detalhada de como o ator deveria se comportar. Utilizando informações de memória do projeto VLS, foi possível identificar estas tarefas, que foram incorporadas nos exemplos dos modelos SR, multiplicando as possibilidades do processo ELICERE de analisar uma gama maior desvios de intenção de projeto.

Para a execução do processo ELICERE no projeto V-ALFA foi necessário definir algumas premissas básicas para a condução do questionário. Premissas como, por exemplo, os tipos de falhas consideradas na análise, uma categorização para os possíveis níveis de perigo e uma classificação para a severidade das falhas. Também foi necessário definir uma matriz associando esses níveis de severidade a uma probabilidade de ocorrência, a fim de auxiliar na definição de metas-soft do risco considerado aceitável para este tipo de perigo, e qual seria a probabilidade de ocorrência aceitável para este tipo de risco. Essas tabelas foram baseadas em referências de severidade e risco adotados pela ESA e pela NASA, e foram apresentadas escalas ilustrativas baseadas nas atuais definições adotadas pela AEB.

Nos exemplos apresentados no estudo de caso do sistema V-ALFA foram feitas as análises de palavras-guia de sete metas, seis recursos e sete tarefas, através de seus respectivos questionários, a fim de mostrar como pode ser conduzido o processo ELICERE.

Foram escolhidas metas relacionadas tanto a funcionalidades críticas de voo do sistema, que são tratadas diretamente pelo computador, como a aquisição de dados inerciais e controle do veículo, quanto às relativas à funcionalidade do sistema em solo, como suas atividades de preparação para voo. Foi também escolhida uma meta relativa à transmissão de dados de voo e outra relativa ao sistema de segurança de voo, que interagem com o sistema embarcado através de um operador humano.

A intenção de se identificar, na análise de palavras-guia, uma meta-soft associada a risco aceitável de um determinado componente, seja ele meta, recurso ou tarefa, é se estabelecer de forma qualitativa/ quantitativa, quais as condições ou valores necessários para se manter um patamar de risco tolerável pelo sistema a ser construído. Este risco aceitável associado com a severidade atribuída ao componente analisado identificará a probabilidade aceitável de ocorrência deste risco, em termos da frequência admitida para sua ocorrência. Isto significa criar metas-soft que mantenham o componente do sistema dentro de faixas de probabilidades aceitáveis.

No caso de se tratar de recurso de hardware, metas-soft quantitativas podem ser definidas de modo a garantir que um componente do sistema V-ALFA como, por exemplo, o computador de bordo, tenha um grau de confiabilidade compatível com sua missão (lançar um satélite em órbita). A aquisição de componentes de hardware com qualidade espacial, muitas vezes disponível somente no mercado internacional, pode ser necessária, ou a manufatura e qualificação deste tipo de recurso dentro de padrões e normas espaciais, ou ainda o desenvolvimento de arquiteturas tolerantes à falha. No caso de recursos lógicos ou tarefas do sistema, metas-soft qualitativas devem ser definidas de modo a garantir a qualidade do componente em questão. Isto significa, no caso de tarefas relacionadas a software, por exemplo, definir metas

de processo e produto que permitam melhorar a confiança no funcionamento do sistema analisado.

Metas de processo devem ser definidas como metas que estipulem a verificação e validação do componente em questão, como as práticas de análise da dependabilidade de sistema propostas pela ECSS-Q-ST-30C (2009) e ECSS-Q-ST-40C (2009). Estes padrões da ESA apresentam planos onde são indicadas técnicas e métodos para garantir a dependabilidade e a segurança de um projeto espacial. Na fase de “Análise da Missão/Identificação das Necessidades” do projeto são identificados os requisitos relevantes de dependabilidade, assim como na fase de “Definição Preliminar” são atribuídas severidades a cada componente e definidos os requisitos não funcionais em todos os níveis do sistema. Nesta fase, podem ainda ser sugeridas metas que estipulem níveis de estresse para os testes, critérios de aceitação e rejeição de componentes críticos, margens de desempenho, fatores humanos que possam influenciar uma operação que ponha em risco o sistema, a identificação de fatores externos, internos e de instalações que possam influenciar negativamente no desenvolvimento do sistema, entre outras metas-soft.

Como metas de produto, podem ser definidas metas que garantam a integridade da tarefa ou do recurso através da identificação do grau de tolerância do hardware a falhas ou ao mau funcionamento do software. A identificação e implementação de programas de controle nos itens críticos do sistema, a inclusão de funções de detecção, isolamento, diagnóstico e recuperação de componentes contra falhas, programação defensiva, entre outras metas, devem ser descritas em detalhe na análise das palavras-guia de tarefa e recurso.

A união da severidade com o risco aceitável para um desvio analisado (recurso ausente, por exemplo) é feita como objetivo de se classificar os desvios de intenção de projeto, de forma a identificar aqueles que deverão receber metas prioritariamente.

No estudo de caso do projeto V-ALFA foi exemplificado um conjunto adequado de metas-soft relacionadas a fatores de qualidade - que promoverão requisitos não funcionais - para sistemas computacionais críticos, subdividido em três grupos: capacidade de defesa, adequação e qualidade. O conjunto de fatores proposto pelo

processo ELICERE visa principalmente guiar o engenheiro de requisitos na escolha da meta de dependabilidade mais adequada para amenizar ou solucionar o desvio de intenção de projeto analisado.

No grupo de capacidade de defesa foram agrupados os fatores diretamente ligados às questões de segurança e tolerância a falhas de um sistema crítico. Para os componentes analisados que necessitem de algum desses fatores, devem ser identificadas metas que atendam às restrições de dependabilidade identificadas previamente, e que permitam ações do sistema como a detecção, reação, proteção, e até mesmo adaptação, contra ameaças, incidentes, acidentes, perigos e riscos ao sistema. As metas propostas devem identificar também procedimentos, como alarmes ao operador em caso de perigos, manutenção do sistema em funcionamento mesmo em condições degradadas, ou ainda a implantação ou melhoria de salvaguardas no sistema contra ataques maliciosos, entre outros procedimentos ligados a robustez, segurança, capacidade de sobrevivência, tolerância à falhas e recuperabilidade.

Nos grupos dos fatores de adequação e de qualidade foi apresentado um conjunto de atributos extraídos de modelos de diversos autores, como o da qualidade proposto por Firesmith (2004, 2006), e o de fatores da qualidade apresentado por Camargo Júnior; Almeida Júnior e Melnikoff (1997), sob o ponto de vista da segurança. Este último foi baseado na avaliação dos modelos de qualidade de McCall (KITCHENHAM, PFLEEGER, 1996) e do padrão da ABNT-NBR-ISO/IEC-9126 (1994) e sugere um conjunto de atributos que representam o conceito de segurança utilizado em sistemas críticos.

O conjunto de fatores proposto pelo processo ELICERE, e aplicado no estudo do V-ALFA, ajuda a garantir a integridade das metas, tarefas e dos recursos do sistema, através de metas-soft qualitativas e quantitativas. Metas-soft devem ser definidas para as metas do sistema para garantir que os objetivos do sistema a ser desenvolvido estão alinhados com os objetivos do cliente. O uso de técnicas como GQM (no inglês, *Goal, Question, Metric*), como sugeridas por Kavakli e Loucopoulos, (2003) pode ajudar a identificar se as metas foram atingidas parcial ou integralmente, levantando questões que possam obter respostas mensuráveis. É

importante definir uma escala de valores (0 a 5, por exemplo) onde se identifique a satisfação da meta com relação às necessidades do cliente, o entendimento do que precisa ser feito, o número de versões produzidas para que a meta alcance uma descrição satisfatória, entre outras questões que devem ser descritas e avaliadas principalmente para os objetivos mais críticos do sistema.

Com relação a metas-soft de recursos e tarefas, podem sugerir a criação de rotinas que verifiquem, por exemplo, se uma entrada ou saída foi devidamente executada, verifique a consistência de um dado em uma possível divisão por zero, ou o desempenho de um determinado trecho de programação em uma região crítica do sistema.

As metas podem também sugerir uma descrição detalhada da cada tarefa do sistema através de comentários, ou regras para a construção das tarefas de modo a facilitar a sua manutenção. Uma limitação do número de processos que disputam os recursos do sistema e uma definição clara do tempo de resposta esperado para cada tarefa crítica do sistema são outras metas que podem ser definidas a partir da análise do V-ALFA. Devem ser ainda levados em conta uma série de fatores como confiabilidade, previsibilidade, disponibilidade operacional, testabilidade, desempenho (e seus subfatores), precisão, rastreabilidade, facilidade de manutenção, precisão, entre outros na elaboração de metas-soft.

É importante enfatizar que o processo foi aplicado ao estudo de caso de forma ilustrativa. Não representa a avaliação de um projeto real. A intenção foi mostrar a viabilidade do processo em uma aplicação, o mais próxima possível do real. Os resultados obtidos com o estudo de caso da tese não são suficientes e completos ao ponto de avaliar situações factíveis e complexas de um projeto espacial real, que demandariam um envolvimento não compatível com a condução do projeto acadêmico.

5.3 CONTRIBUIÇÕES DO PROCESSO ELICERE

Uma das importantes contribuições do processo ELICERE é a melhoria da cultura de segurança em organizações que desenvolvem sistemas críticos. O estudo independente apresentado por Almeida e Johnson (2005) sobre o acidente do VLS-1 para o voo 3 cita a pouca atenção ao desenvolvimento de uma “cultura de segurança” apropriada ao projeto. Levando em consideração também outros estudos sobre projetos de sistemas críticos da área espacial (LEVESON, 2002) (JOHNSON, 2003) (HJORTNAES, 2003) (NASA-MCO, 1999) (NASA-JPL, 2000), mesmo de agências com um alto nível de maturidade em termos de profissionais e infra-estrutura, conclui-se que existem ainda problemas de ordem técnica (tecnologia), operacional (processo) e humana (pessoas) (LAHOZ et al., 2006). Estes obstáculos devem ser ultrapassados, baseando-se principalmente em uma mudança cultural das pessoas e da organização envolvida no processo. Em qualquer que seja a fase do projeto ou a disciplina envolvida, esta mudança deve ser calcada na qualidade e, mais especificamente, na definição de metas que ajudem a aumentar o grau de confiança no funcionamento do sistema.

Na visão de tecnologia, verifica-se que em muitas organizações desenvolvedoras de sistemas computacionais críticos o uso de técnicas de elicitação e de representação de requisitos ainda são inadequados, ou mesmo inexistentes. Entrevistas, prototipação ou mesmo o estudo de documentação, muitas vezes são mal empregadas, apresentando resultados não muito confiáveis. Também, o uso inapropriado de modelos de representação dos componentes do sistema e suas interações podem não destacar adequadamente as dependências comportamentais advindas da especificação destes componentes, gerando problemas que podem ser descobertos (quando descobertos) somente em fases muito avançadas do desenvolvimento de um sistema. Caso típico de perdas financeiras, de atraso no cronograma de projeto e perda total do crédito da equipe. A falta de uma abordagem sistêmica de segurança em sistemas intensivos de software, principalmente em ambientes de pouca maturidade em cultura de segurança, pode não ser percebido, em um primeiro momento, nas atividades de elicitação, mas fatalmente afetará os

estágios mais avançados do desenvolvimento, ou, pior ainda, somente quando o sistema estiver em operação.

O processo ELICERE, através da condução das atividades de eliciação de requisitos de dependabilidade ainda nas fases iniciais de um projeto pode ajudar a ultrapassar esta barreira da tecnologia. Desenhar o sistema em termos de metas, tarefas e recursos, e a partir daí realizar uma análise de perigos baseada em palavras-guia aplicadas a estes componentes do sistema, permite avaliar se os riscos associados ao sistema são aceitáveis. A criação de um processo que permita a indicação de metas de dependabilidade promove soluções capazes de atender a uma especificação mais completa e confiável. Outro grande benefício é o uso de uma linguagem comum entre times heterogêneos, como engenheiros de sistema e de software, para desenvolverem soluções de hardware e de software que melhorem a segurança e a confiabilidade do sistema.

Na visão de processo, a indefinição ou pobreza de modelos de processo, assim como uma abordagem precária da dependabilidade, está fortemente relacionada a pouca maturidade das equipes e das organizações. Muitas vezes, por razões externas ao projeto, como pessoal sem a qualificação necessária, estrutura organizacional ineficiente ou mesmo pressões de tempo e de recursos, as atividades do processo de eliciação de requisitos são menosprezadas em detrimento de outras atividades, como por exemplo, as relacionadas diretamente ao desenvolvimento de código. Outro obstáculo presente em projetos de sistemas críticos é a comunicação inadequada entre os envolvidos. Sem o auxílio de algum tipo de mecanismo ou de procedimento, o processo de comunicação será ineficiente, não permitindo que durante o desenvolvimento do projeto os requisitos extraídos sejam avaliados de forma abrangente, de maneira a permanecerem estáveis durante a vida do projeto.

O processo ELICERE se propõe a facilitar a condução das atividades de eliciação através da formalização de um conjunto de passos, que inclui a criação de modelos do sistema e em seguida a aplicação de um questionário baseado em palavras-guia. Estas atividades, dependendo do nível de detalhe pretendido, podem durar um tempo relativamente pequeno no cronograma de um projeto, permitindo assim sua realização. A representação do sistema baseado na técnica i^* é de fácil

entendimento para engenheiros de sistema e de software, bem como o questionário sugerido. Esses fatores são um forte estímulo para que o processo ELICERE seja incluído no desenvolvimento de um sistema crítico, além do fato de permitir a identificação e priorização dos recursos e tarefas a serem trabalhados para a melhoria constante da dependabilidade.

Na visão de pessoas, diferentes enfoques de ordem cultural entre os participantes, e mesmo entre engenheiros de sistema e engenheiros de software, podem criar um obstáculo quase intransponível. Software é uma disciplina relativamente nova, segundo a visão dos engenheiros de sistema, e muitas vezes, a participação da equipe de software nas atividades de engenharia de requisitos de sistema não é considerada tão importante (GONZALES, 2005). Outro ponto crítico é o entendimento limitado do domínio do problema que, aliado a uma visão parcial dos envolvidos, prejudica qualquer técnica de elicitação. Sem o entendimento claro e preciso do domínio do problema, é difícil interagir com os participantes e ser capaz de ir além da visão particular de como cada envolvido enxerga o sistema. Ainda mais grave é um gerenciamento de projeto que não considera a importância das atividades de elicitação. A gerência de projeto tem que equilibrar o interesse dos participantes no processo de elicitação, definir as prioridades dos usuários e promover o consenso entre os envolvidos. Sem este equilíbrio, forças antagônicas de interesse aparentemente comuns, podem gerar, como uma consequência extrema, a morte lenta e gradual de um projeto.

Na visão do processo ELICERE o engenheiro de requisitos tem de ser integrado tanto à equipe de sistema como à de software, bem como ao gerenciamento de projeto. A intenção é poder obter uma visão mais abrangente e completa do problema. Esta integração deve permitir compartilhar as especialidades e os valores de cada perfil de engenharia e ainda criar uma cultura de segurança única. Deve-se alcançar um meio termo entre a aversão ao risco da engenharia de sistema versus a tendência a modismos da engenharia de software. A implantação da cultura de segurança ainda nas atividades de análise da missão deve propiciar um reconhecimento claro de seu valor pelos participantes, estar integrada a todas as atividades de engenharia, capaz de ser contabilizada e ter uma situação de liderança (relativamente) independente.

O que se espera, tanto no nível organizacional, quanto gerencial e de engenharia, é que o processo ELICERE traga uma nova mentalidade, fortemente direcionada à dependabilidade, obrigando as equipes a se posicionarem de forma pró-ativa no atendimento de requisitos de confiabilidade e segurança. É necessário que qualquer recurso computacional, embutido ou em solo, seja submetido a algum tipo de análise ou avaliação de sua contribuição, ou não, ao sucesso da missão. Todas as fases de desenvolvimento de um produto devem ser observadas sob o ponto de vista da dependabilidade, e caso seja detectado algum tipo de comportamento não seguro, metas de dependabilidade devem ser definidas no sentido de se garantir que um comportamento seguro será mantido pelo sistema.

5.4 TRABALHOS FUTUROS

Um dos próximos desafios do processo ELICERE será o de criar uma estrutura computacional para manter o controle e a rastreabilidade das metas-soft com os recursos, tarefas e metas do sistema. Uma quantidade significativa de metas-soft deverá ser armazenada, classificada e manipulada de forma inteligente, de maneira a priorizar e viabilizar a incorporação das sugestões apresentadas. Só assim será possível uma análise completa de um sistema de grande porte e de alta complexidade, baseada em palavras-guia, abrangendo todas as particularidades de um projeto computacional e, assim, serem sugeridas metas de dependabilidade.

Outra vertente de trabalho está relacionada à formulação de objetivos ou metas-soft mais objetivas. Metas muito subjetivas, que somente expressam um determinado interesse ou visão de um participante no sistema, dificultam o entendimento do conjunto como um todo. Que pese o fato da adoção da técnica de elicitação proposta pelo processo ELICERE contribuir significativamente para o campo da engenharia de requisitos, entretanto, pode não ser imediata a extração dos requisitos (não-funcionais), tão necessários para a garantia da dependabilidade em um sistema computacional crítico. É preciso aprimorar a semântica para a derivação de metas em requisitos, até se atingir um nível objetivo e claro do requisito não

funcional. Será necessário até derivar metas em submetas, até se alcançar um nível de granularidade que possa ser mapeado em um requisito, ou conjunto de requisitos detalhados e atômicos.

Paralelamente ao trabalho de se mapear metas em requisitos não-funcionais é desejável atribuir um valor quantitativo para cada meta, de modo a se dimensionar valores mais objetivos e mensuráveis para cada meta-sof proposta. Esta quantificação facilitará a verificação e a validação do requisito, para aqueles que forem participar tanto da especificação como da aceitação de um sistema computacional crítico.

Outro trabalho que pode ser explorado é o de viabilizar o uso do processo ELICERE de forma recursiva. As metas-soft extraídas a partir do processo de elicitação, e conseqüentemente os requisitos não funcionais definidos, poderiam estar sujeitas a uma nova avaliação por parte do processo. O propósito desta avaliação é de refinar o modelo e garantir que estes novos requisitos atendam aos objetivos de missão e de segurança previamente estabelecidos. Isto significa aplicar o processo quantas vezes forem necessárias, até que algum critério de parada prévio seja atingido ou, por exemplo, que os responsáveis pelo projeto concordem com o grau de refinamento do modelo.

A atividade de validação das metas-soft identificadas pelo processo ELICERE, com relação aos requisitos do projeto e depois à arquitetura, não foi abordado neste trabalho. Este tipo de processo deve ser um dos próximos desafios do ELICERE no intuito de garantir que as metas de dependabilidade estão adequadas e pertinentes para serem adotadas pelo sistema durante todo seu ciclo de desenvolvimento.

Expandir a aplicação do processo ELICERE para os sistemas que interagem com o sistema computacional crítico (sistema de contexto sócio-técnológico). Com isso seria possível viabilizar a indicação dos fatores de dependabilidade para o gerenciamento desse tipo de projeto, para recursos humanos, para procedimentos e processos em sistemas de tecnologia de informação, sob o prisma da confiança no funcionamento.

Finalmente, uma iniciativa a ser tomada é idealização de um projeto de pesquisa que viabilize agregar ainda mais as habilidades e competências de engenheiros de requisitos e de engenheiros de segurança de sistemas, tanto no sentido de consolidar o processo ELICERE, como seus possíveis desdobramentos para sistemas computacionais críticos.

Principalmente no que se refere à cooperação de centros de pesquisas, de universidades e da indústria especializada no fomento ao desenvolvimento tecnológico nesta área da engenharia de sistemas e de software. Parcerias em projetos que promovam a melhoria da dependabilidade em sistemas computacionais embarcados através de cooperações, como por exemplo, entre o Instituto de Aeronáutica e Espaço (IAE), o Instituto Nacional de Ciência e Tecnologia – Sistemas Embarcados Críticos (INCT-SEC) e a Associação das Indústrias Aeroespaciais do Brasil (AIAB), mostra-se um caminho a ser trilhado, a fim de diminuir os obstáculos técnicos, operacionais e culturais atualmente encontrados em projetos de sistemas críticos da área espacial.

5.5 CONCLUSÕES FINAIS

Esta tese propõe um processo de elicitação a ser aplicado a projetos críticos, como os da área espacial, que minimize problemas relacionados à dependabilidade, como segurança e confiabilidade, entre outros fatores de qualidade, ainda nas fases iniciais de um projeto.

O fato do processo não exigir um conhecimento de técnicas sofisticadas de engenharia de requisitos, nem de análise de perigos é um forte indicativo de que o ELICERE pode ser aplicado em qualquer ambiente de desenvolvimento de sistemas críticos. Não exige o uso de ferramentas automáticas de apoio à execução e consome um tempo parcial de dedicação por parte de quem aplica o processo no projeto. Apresenta um custo baixo em relação ao benefício que pode proporcionar, principalmente levando-se em consideração que na maioria dos projetos de sistemas críticos, a análise de perigos pode chegar a consumir um percentual de recursos

significativo no orçamento do projeto. Segundo Hecht e Buettner (2006), o custo da engenharia de confiabilidade de software, pode variar de 0.1 a 3.0 por cento dos custos totais de desenvolvimento .

Ao final acredita-se que o processo ELICERE possa contribuir com a engenharia de software em diversos aspectos. Primeiro, na aplicação da técnica de modelagem de requisitos organizacionais em projetos de sistemas críticos e estratégicos, como os da área espacial no Brasil, fato que até a presente data não tinha sido experimentado. Segundo, na criação de um conjunto de palavras-guia aplicadas aos principais componentes de um sistema computacional para elicitación de metas de dependabilidade. Terceiro, no levantamento de um conjunto mínimo de fatores de qualidade para utilização na análise de sistemas que possuem restrições rigorosas quanto à segurança, disponibilidade entre outros atributos. Finalmente, na melhoria da comunicação entre os participantes do desenvolvimento de um sistema crítico, através de uma abordagem integrada entre as engenharias de sistema e de software para identificação de metas de dependabilidade.

A apresentação do processo ELICERE através de um estudo de caso mostrou sua viabilidade prática de uso em sistemas críticos, voltada para a área de trabalho de atuação do autor. Foi possível também submeter o trabalho de tese a avaliação de pesquisadores da área de engenharia de requisitos e de engenharia de segurança, através de artigos técnicos e apresentações em diversos eventos nacionais e internacionais, corroborando com a proposta do processo ELICERE, e na sua aplicação prática.

O uso dos modelos i^* e de palavras-guia, adotados para o ELICERE, são pertinentes e adequados para a investigação do sistema com relação à dependabilidade. Eles proporcionam um estímulo à descoberta de requisitos não funcionais e ao intercâmbio de conhecimento entre os envolvidos (engenharia de sistemas, requisitos e segurança). Além de facilitar a descoberta de requisitos não funcionais, promovem sua organização dentro de uma estrutura de fácil acesso e priorização.

O tipo de abordagem proposta pode auxiliar no encaminhamento de soluções de novos projetos de sistemas espaciais, como os do programa Cruzeiro do Sul,

incorporando na visão dos sistemas computacionais dos futuros veículos uma nova ótica de segurança. Isto significa estimular a discussão de questões como a da escolha dos fatores de qualidade mais adequados para sistemas computacionais, visto que somente a incorporação do fator confiabilidade não ser, por si só, suficiente para garantir a confiança no funcionamento do sistema. Sistemas computacionais intensivos de software exigem que engenheiros de sistemas adotem uma nova visão para manipular os novos perigos e riscos advindos deste tipo de sistema, que atinge níveis de complexidade e de controle muitas vezes fora da capacidade da compreensão humana.

O processo ELICERE pretende proporcionar uma forma de investigação metodológica e objetiva. Ele é um processo complementar e não substitutivo, devendo ainda ser aplicados nos projetos de sistemas computacionais críticos técnicas convencionais de engenharia de requisitos e de engenharia de segurança para a especificação dos requisitos. O que se destaca são os ganhos que o processo traz ao levantar a discussão sobre dependabilidade o mais cedo possível em um projeto, sinalizando, durante a análise das palavras-guia, os riscos e perigos de um projeto desta natureza, e promovendo assim um entendimento comum entre engenheiros de sistemas e de software sobre as questões da dependabilidade. Permite também ser utilizado em fases posteriores de desenvolvimento, inclusive na manutenção do sistema, quando novas funcionalidades devem ser modeladas e avaliadas quanto à introdução de novos riscos ao sistema em operação.

REFERÊNCIAS

ABNT.ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT-NBR-ISO/IEC 9126**: Tecnologia de informação – avaliação de produto de software – características de qualidade e diretrizes para o seu uso. Rio de Janeiro, 1994.

_____.**ABNT-NBR-14857-2**: Sistemas espaciais - gerenciamento do programa - parte 2: garantia do produto. Rio de Janeiro, 2002. 7p.

AEB.AGÊNCIA ESPACIAL BRASILEIRA. **AEB-RTSE-1**: Regulamento Técnico Geral da Segurança Espacial - Parte 1, Resolução nº 71. Brasília, 2007. 63p.

ALENCAR, F. M. R. **Mapeando a modelagem organizacional em especificações precisas**. 1999. 319 p. Tese (Doutorado) – Ciência da Computação do Centro de Ciências Exatas e da Natureza da Universidade Federal de Pernambuco, Recife, 1999.

ALENCAR F.; SILVA, C.; MOREIRA, A.; ARAÚJO, J.; CASTRO, J. Identifying candidate aspects with i-star approach, WORKSHOP ON EARLY ASPECTS, In: INTERNATIONAL CONFERENCE ON ASPECT-ORIENTED SOFTWARE DEVELOPMENT, 5. Bonn, 2006. **AOSD: Proceedings**. Bonn, 2006.

ALENCAR, F.; CASTRO, J.; MONTEIRO, C.; RAMOS, R.; SANTOS, E. Towards aspectual i*.In: INTERNATIONAL i* WORKSHOP, 3. Recife, 2008. **Proceedings**. p. 1-4. Recife, 2008.

ALMEIDA, I. M; JOHNSON C. W: Extending the borders of accident investigation: applying novel analysis techniques to the loss of the Brazilian space programmer's launch vehicle VLS-1 v03. **Safety Science**, v. 46, n. 1, p. 38-53, Elsevier Science, 2005.

ATKINS C., SAMPSON J. Critical appraisal guidelines for single case study research. In: EUROPEAN CONFERENCE ON INFORMATION SYSTEMS, 2002, Poland. **ECIS: Proceedings**. Poland: 2002.

BARBACCI, M.; KLEIN, M. H.; LONGSTAFF, T. A.; WEINSTOCK, C. B. **Quality attributes**. Pittsburgh: Software Engineering Institute, Carnegie Mellon University, 1995. (Technical Report CMU/SEI-95-TR-021).

BAR-YAM, Y. **Unifying Principles in Complex Systems**, In: Converging Technology for Improving Human Performance - Nanotechnology, Biotechnology, Information Technology and Cognitive Science (NBIC). ROCO, M. C.; BAINBRIDGE, W. S. Cambridge: Kluwer Academic Publisher, 2003. 467 p.

BASIL V. R. **Bridging the gap between business strategy and software development**. University of Maryland, *Fraunhofer* Institute Experimental Software Engineering (IESE), 2007. /Relatório técnico/

BLETOS, N. A. Launch vehicle guidance, navigation, and control. **Crosslink**, v. 5, n. 1 Winter 2004. Disponível em: <<http://www.aerospace.org/publications/crosslink/winter2004/06.html>> Acesso em: 15 jun. 2009.

BUBENKO, J. Enterprise modelling. **Ingenierie des systems d' information**, v. 2, n. 6, 1994.

BURNS D.; PITLADO R. A Modified (HAZOP) methodology for safety critical system assessment. In: SAFETY-CRITICAL SYSTEM SYMPOSIUM. 1993, Bristol. **Proceedings**. Bristol: 1993.

BUSH D. Modelling support for early identification of safety requirements: a preliminary investigation. In: INTERNATIONAL WORKSHOP ON REQUIREMENTS FOR HIGH ASSURANCE SYSTEMS, 4. IEEE INTERNATIONAL REQUIREMENTS ENGINEERING CONFERENCE, 13., 2005, Paris. **RHAS05: Proceedings**. Paris: 2005.

CAMARGO JÚNIOR, J. B. **Metodologia de Análise de Risco em Sistemas Computacionais de Aplicação Crítica**. 225 p. Tese de Livre Decência - Escola Politécnica da USP, 2002.

CAMARGO JÚNIOR, J. B.; ALMEIDA JÚNIOR, J. R.; MELNIKOFF, S. S. S. O uso de fatores de qualidade na avaliação da segurança de software em sistemas críticos. In: CONFERÊNCIA INTERNACIONAL DE TECNOLOGIA DE SOFTWARE: QUALIDADE DE SOFTWARE, 8. 1997, Curitiba. **CITS: Anais**. Curitiba: 1997. p. 181-195.

CARD, S. K.; MORAN T. P.; NEWEL A. **The psychology of human-computer interaction**. Hillsdale: Lawrence Erlbaum Associates, 1983.

CASTRO, J.; MYLOPOULOS. J.; KOLP, M. Towards requirements-driven information systems engineering: the Tropos project. **Information Systems**, v. 27, n.6, Sept. 2002.

CASTRO LEITE, W. F. Control system of Brazilian launcher. In: ESA INTERNATIONAL CONFERENCE ON SPACECRAFT GUIDANCE, NAVIGATION AND CONTROL SYSTEM, 4., 1999, Noordwijk. **Proceedings**. Noordwijk: 1999.

_____. Estrutura do sistema de controle do VLS, **Controle e instrumentação**, São Paulo, n. 72, p. 71-77, Ago. 2002.

CHUNG, L.; NIXON, B. A.; YU, E., MYLOPOULOS, J. A. **Non-functional requirements in software engineering**. International Series in Software Engineering, v.5. Kluwer Academic Publishers, 1999. 476 p.

CONMY P. M. **Safety analysis of computer resource management software**. 2005. 245 p. Thesis (Phd) - Department of Computer Science, University of York, York, 2005.

DARDENNE A.; VAN LAMSWEERDE A.; FICKAS S. Goal-directed requirements acquisition. **Science of Computer Programming**, v. 20, p. 3-50, 1993.

DE LEMOS R. **Análise de segurança de software contra faltas acidentais**. In: SIMPÓSIO DE COMPUTADORES TOLERANTES A FALHA, 7. Campina Grande: 1997. /Apostila/

DE LEMOS R.; VERÍSSIMO P. **Confiança no Funcionamento, Proposta para uma Terminologia em Português**, Publicação conjunta INESC e LCM/UFSC. 1989. Disponível em: <<http://www.cs.kent.ac.uk/people/staff/rdl/CoF/inesc.html>> Acesso em: 12 out. 2006.

DIAS LOPES, P. S. N. **Uma taxonomia da pesquisa na área de engenharia de requisitos**. 2002. 92 p. Dissertação (Mestrado) - Instituto de Matemática e Estatística da Universidade de São Paulo, São Paulo, 2002.

DOD.DEPARTMENT OF DEFENSE, Military Standard **DOD-MIL-STD-1629A**: Procedures for performing a failure mode, effects and criticality analysis. Washington, D.C., 1980. /Cancelada em 1998/

_____. **DOD-MIL-STD-2167A**: Defense system software development. Washington, D.C., 1988. /Cancelada/

_____. **DOD-MIL-STD-498**: Software development and documentation. Washington, D.C., 1994.

_____. **DOD-MIL-STD-882D**: Standard practice for system safety. Washington, D.C., 2000.

ECSS.EUROPEAN COOPERATION for SPACE STANDARDIZATION **ECSS-E-ST-40C**: Space engineering: software. Noordwijk, 2009. 206 p.

_____. **ECSS-M-ST-10C**: Space project: management – project planning and implementation - rev.1. Noordwijk, 2009. 50 p.

_____. **ECSS-P-001B**: Glossary of terms. Noordwijk, 2004. 36 p.

_____. **ECSS-Q-80-02**: Software process assessment and improvement – part 1: framework. Noordwijk, 2005. 115 p.

_____. **ECSS-Q-ST-30C**: Space product assurance: dependability. Noordwijk, 2009. 54 p.

_____. **ECSS-Q-ST-30-02C**: Space product assurance: failure modes, effects (and criticality) analysis (FMEA/FMECA). Noordwijk, 2009. 74 p.

_____. **ECSS-Q-ST-40C**: Space product assurance: safety. Noordwijk, 2009. 75 p.

_____. **ECSS-Q-ST-80C**: Space product assurance: software product assurance. Noordwijk, 2009. 113 p.

ELLIS, C. A.; WAINER, J. Goal-based models of collaboration. **Collaborative Computing**, v. 1, n. 1. p. 61-86, 1994.

ESTRADA, H.; MARTINEZ, A.; PASTOR, O.; MYLOPOULOS, J.; GIORGINI, P. A service-oriented approach for the i* framework. In: INTERNATIONAL i* WORKSHOP, 3. Recife, 2008. **Proceedings**. p. 21-24. Recife, 2008.

FARIA, E. **Dicionário escolar latino-português**, 3rd. ed. Brasília, D. F.: Ministério da Educação e Cultura, 1962.

FEATHER, M. Language support for the specification and development of composite systems. **ACM Transactions on Programming Languages and Systems**, v. 9, n. 2, p. 198-234, 1987.

FIRESMITH, D. G. Engineering safety requirements, safety constraints, and safety-critical requirements. **Journal of Object Technology**, Zurich, March/April, 2004. p. 27-42.

_____. Engineering safety-related requirements for software-intensive systems. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 28., 2006, Shanghai. **ICSE: Proceedings**. Shanghai: 2006. /Tutorial/

_____. **Open process framework (OPF), quality factors**. Disponível em: <<http://www.opfro.org/index.html?Components/WorkProducts/ModelSet/QualityModel/QualityFactors/QualityFactors.html~Contents>>, Acesso em 21 maio 2007.

GALETTI, I.; SPINOLA, M. Um processo de engenharia de requisitos para assessorar na seleção das técnicas de eliciação de requisitos de software. In: ENCONTRO NAC. DE ENG. DE PRODUÇÃO, 25., 2005, Porto Alegre. **Anais**. Porto Alegre: 2005.

GONZALES, R. Developing the requirements discipline: software vs. systems. **IEEE Software**, March/April 2005, p. 59-61.

GREENSPAN, S.; MYLOPOULOS, J., BORGIDA, A. On formal requirements modeling languages: RML revisited. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 16. , 1984, Sorrento. **ICSE: Proceedings**. Sorrento: 1984. p. 135-147.

HAMMER, W. **Handbook of system and product safety**. Englewood: Prentice-Hall, 1972.

HECHT, M.; BUETTNER, D. Software testing in space programs. **Crosslink**, v. 6, n. 3 (Fall2005). Disponível em: <<http://www.aero.org/publications/crosslink/fall2005/06.html>> Acesso em: 29 Jan. 2006.

HJORTNAES, K. **ESA software initiative**. ESA Report, May, 2003 Disponível em http://klabs.org/richcontent/software_content/presentations/esa_software_initiative_final_may_7.pdf Acesso em 20/05/2006.

IEC.INTERNATIONAL ELECTROTECHNICAL COMISSION: **IEC-61508-1**. Functional safety of electrical/eletronic/programmable eletronic safety-related systems – part1: general requirements. 1998.

IEEE.INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS: **IEEE-830**. IEEE guide to software requirements specifications. New York, 1984.

INGHAM M. D.; RASMUSSEN R. D.; BENNETT M. B.; MONCADA A. C. Generating requirements for complex embedded systems using state analysis. In: INTERNATIONAL ASTRONAUTICAL CONGRESS, 55., 2004, Vancouver. **IAC: Proceedings**. Vancouver: 2004.

ISAKSEN, U.; BOWEN, J.; NISSANKE, N. **System and software safety in critical systems**. Reading: Department of Computer Science, The University of Reading, 1997. (Technical Report RUCS/97/TR/062/A).

ISO/IEC.INTERNATIONAL ORGANIZATION FOR STANDARDIZATION/ INTERNATIONAL ELECTROTECHNICAL COMMISSION: **ISO/IEC-12207** Information technology - software life cycle processes. Geneva, 2001. (Amendment).

_____.**ISO/IEC 15504-2**: Information technology — process assessment: part 2: - performing an assessment. Geneva: International Standard, 2003.

JOHNSON, C. W. **A handbook of accident and incident reporting**. Glasgow University Press, Glasgow, 2003. Disponível em: <<http://www.dcs.gla.ac.uk/~johnson/book/>> Acesso em 25 jun 2009.

_____.What are emergent properties and how do they affect the engineering of complex systems? **Reliability Engineering and System Safety**, v. 91, n. 12, Dec. 2006. p. 1475-1481. Disponível em <<http://www.dcs.gla.ac.uk/~johnson/papers/emergence.pdf>> Acesso em: 1 jun. 2009.

JONES, M.; GOMEZ, E.; MANTINEO, A.; MORTENSEN, U.K.; **Introducing ECSS software- engineering standards within ESA – practical approaches for space- and ground-segment software**, ESA, Aug. 2002. (Bulletin 111).

KAVAKLI, V.; LOUCOPOULOS, P. Goal driven requirements engineering: evaluation of current methods .In: EVALUATION OF MODELING METHODS IN SYSTEMS ANALYSIS AND DESIGN, 2003, Velden. **EMMSAD: Proceedings**. Velden: 2003.

_____.Modelling of organisational change using the EKD framework. **Communications of the Association for Information Systems**, v. 2, July 1999.

KITCHENHAM, B.; PFLEEGER, BS. L. **Software** quality: the elusive target. **IEEE Software**, p.12-21, Jan. 1996.

KLETZ, T. A. The Origins and history of loss prevention, process safety and environmental protection. **Developments in Process Safety**. v. 77, May 1999, p. 109-116.

KOTONYA, G.; SOMMERVILLE, I. **Requirements engineering**. West Sussex: John Wiley & Son Ltd, 2000. 282 p.

LAHOZ, C. H. N. **Uma abordagem para a gerência da qualidade em um ambiente de engenharia de software centrado em processo**. 2004. 307 p. Dissertação (Mestrado) - Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2004.

_____.**Defining dependability goals in computer systems for space projects**. Trabalho apresentado no IEEE International Requirements Engineering Conference, 16., Doctoral Symposium, 2008. Barcelona.

_____.Processos de desenvolvimento de software para a área espacial. In: SIMPÓSIO INTERNACIONAL DE MELHORIA DE PROCESSO DE SOFTWARE, 9. 2007, São Paulo. **SIMPROS: Tutorial**. São Paulo: 2007.

LAHOZ, C. H. N.; ABDALA M. A. D.; MOURA C. A. T.; DE LEMOS R. Avaliação de ações cooperativas na análise da segurança do sistema de tele-destruição de um veículo de sondagem In: SIMPÓSIO DE SEGURANÇA EM INFORMÁTICA, 2000, São José dos Campos. **SSI: Anais**. Instituto Tecnológico de Aeronáutica, S. J. Campos: 2000.

LAHOZ, C. H. N.; ALVES, M. C. B.; ABDALA, M. A. D. **Definição e melhoria de processos de software para o desenvolvimento de sistemas críticos**. In: SIMPÓSIO INTERNACIONAL DE MELHORIA DE PROCESSO DE SOFTWARE, 7., 2005, São Paulo. **SIMPROS Tutorial**. São Paulo: 2005.

LAHOZ, C. H. N.; BURGARELI, L. A.; CAMARGO JÚNIOR, J. B. Definition and modeling of a software safety process for space applications. In: INTERNATIONAL SYSTEM SAFETY CONFERENCE, 23., 2005, San Diego. **ISSC: Proceedings**. San Diego: 2005. 1 CD-ROM.

LAHOZ, C. H. N.; BURGARELI, L. A.; MOURA, C. A. T. Melhoria e confiabilidade em desenvolvimento de software para projetos espaciais. In: SAFETY WORKSHOP, 2. **Anais**. São Paulo: EPUSP, 2004. p.73-92.

LAHOZ, C. H. N.; CAMARGO JÚNIOR, J. B. An approach for dependability requirements elicitation in computer systems for space projects, In: INTERNATIONAL SYSTEM SAFETY CONFERENCE, 25. 2007, Baltimore: **ISSC: Proceedings**. Baltimore: 2007. 1 CD-ROM.

_____.ELICERE: A process for defining dependability goals for critical computer systems, In: ANNUAL OF INTERNATIONAL SYMPOSIUM OF INCOSE, 2009, Singapore.**INCOSE09: Proceedings**. Singapore, 2009.

_____.Um Estudo sobre a atividade de elicitação de requisitos em projetos de software da área espacial. **Cadernos IME, Série Informática**, n. 20, Jul. 2006. Edição especial. p.116-123/Apresentado no Workshop on Requirements Engineering, 9. Rio de Janeiro, 2006.

LAHOZ, C. H. N.; CAMARGO JÚNIOR, J. B.; ABDALA, M. A. D.; BURGARELI, L. A. A software safety requirements elicitationn study on critical computer systems. In: INTERNATIONAL CONFERENCE ON SYSTEM SAFETY, 1., 2006, London. **IET: Proceedings**. London: 2006.

LAHOZ, C. H. N.; MOURA, C. A. T. Defining quality processes models in software engineering environments: a step towards dependable software and manageable projects In: INTERNATIONAL ASTRONAUTICAL CONGRESS, 55. 2004, Vancouver: **IAC: Proceedings**. Vancouver: 2004.1 CD-ROM.

LAPLANTE, P. A. **Real time systems design and analysis**. Piscataway: IEEE Press Wiley- Interscience John Wiley & Sons, 2004.

LAPOUCHNIAN, A. **Goal-oriented requirements engineering: an overview of the current research**. Toronto: Department of Computer Science, University of Toronto, 2005. /Technical Report/

LAUESEN, S. **Software requirements, styles and techniques**. Harlow: Addison-Wesley, 2002. 591 p.

LEITE, J. C. S. P.; CAPPELLI, C. Exploring i* characteristics that support software Transparency In: INTERNATIONAL i* WORKSHOP, 3. Recife, 2008. **Proceedings**. p. 51-54. Recife, 2008.

LEITE, J. C. S. P.; WERNECK, V. M. B.; OLIVEIRA, A. P. A.; CAPPELLI, C.; CERQUEIRA, A. L. A.; CUNHA, H. S.; GONZALES-BAIXAULI, B. Understanding the strategic actor diagram: an exercise of meta modeling. In: WORKSHOP EM ENGENHARIA DE REQUISITOS, 7. Toronto, 2007. **WER07: Proceedings**. p. 2-12. Toronto, 2007.

LEVESON, N. **Evaluating accident models using real aerospace accidents**. MIT SERL Technical Report, June, 2001. ESD-WP-2002-06, Disponível em: <<http://sunnyday.mit.edu/accidents/>> Acesso em 20 jun. 2007.

_____. The role of software in spacecraft accidents, **AIAA Journal of Spacecraft and Rockets**, v. 41, n. 4, July 2004.

_____. **Safeware – system safety and computers**. USA: Addison Wesley, 1995. 680 p.

_____. **System safety engineering: back to the future**, Aeronautics and Astronautics - Massachusetts Institute of Technology, 2002. Disponível em: <<http://sunnyday.mit.edu/book2.pdf>> Acesso em: 09 mar. 2007. [Draft book]

_____. **White paper on approaches to safety engineering**. April, 2003, Disponível em: <<http://sunnyday.mit.edu/caib/concepts.pdf>> Acesso em 15 dez. 2006.

LOUCOPOULOS, P; KARAKOSTAS, V. **System requirements engineering**. London: McGraw Hill, 1995.

LUNDEBERG, M. The ISAC approach to specification of information systems and its application to the organisation of an IFIP working conference. In: OLLE, T. W.; SOL, H. G.; VERRIJN-STUART, A. **Information systems design methodologies: a comparative review**. New York: Elsevier Science, 1982.

LUTZ, R. R. Analyzing software requirements errors in safety-critical, embedded systems. In: IEEE INTERNATIONAL SYMPOSIUM ON REQUIREMENTS ENGINEERING, 1993. **ICRE: Proceedings**. San Diego: IEEE Computer Society Press, Jan. 1993. p. 53-65.

LUTZ, R. R.; MIKULSKI I. C. Ongoing requirements discovery in high-integrity systems. **IEEE Software**, March / April 2004, v. 21, n. 2, March 2004. p. 19-25.

LUTZ, R. R.; WOODHOUSE R. M. Experience report: contributions of SFMEA to requirements analysis. In: IEEE INTERNATIONAL CONFERENCE ON REQUIREMENTS ENGINEERING, 2. 1996, Colorado Springs. **ICRE: Proceedings**. Colorado Springs: 1996.

_____.Requirements analysis forward and backward search. **Software Engineering**, Special Volume on Requirements Engineering, 1997.

LYONS, J. L. **Report of the inquiry board into the failure of flight 501 of the ariane 5 rocket**. European Space Agency, Paris, 1996. /Technical report/

MAGEE, C. L.; DE WECK, O. L. Complex system classification. In: ANNUAL INTERNATIONAL SYMPOSIUM OF THE INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING, 14., 2004, Toulouse. **INCOSE: Proceedings**. Toulouse: 2004.

MAIDEN, N. A. M.; JONES, S. V. Dependability in RESCUE: a concurrent engineering approach to the specification of requirements for air traffic management. In: INTERNATIONAL CONFERENCE ON DEPENDABLE SYSTEMS AND NETWORKS, WORKSHOP ON INTERDISCIPLINARY APPROACHES TO ACHIEVING AND ANALYSING SYSTEM DEPENDABILITY, 2004, Florence. **DSN: Proceedings**. Florence: 2004.

MAIDEN, N. A. M.; JONES, S.V.; FLYNN M. Innovative requirements engineering applied to ATM. In: AIR TRAFFIC MANAGEMENT, 2003, Budapest. **ATM: Proceedings**. Budapest: 2003.

MAIDEN, N. A. M; KAMDAR, N.; BUSH, D. Analysing i* system models for dependability properties: the uberlingen accident. In: WORKING CONFERENCE ON REQUIREMENTS ENGINEERING: FOUNDATION FOR SOFTWARE QUALITY, 12., Luxembourg, 2006. **Proceedings**. Luxembourg: 2006.

MOD.MINISTRY OF DEFENCE **MOD-DEF-STAN-00-40**: Reliability and maintainability (R&M) – part 7 (ARMP-7) NATO R&M terminology applicable to ARMP's, Glasgow, 2003.

_____.**MOD-DEF-STAN-00-58**. HAZOP studies on systems containing programmable electronics, part 1 requirements. draft Issue 2, Glasgow, 2000.

MORAES, JR., P.; CARRIJO, D. S.; GARCIA, A. ; COSTA, L. E. L ; OLIVEIRA, U. C.; SANTANA JR., A.; VILLAS BOAS, D. J. F.; YAMAMOTO, M. K. An overview of the Brazilian vehicle program cruzeiro do sul. In: INTERNATIONAL ASTRONAUTICAL CONGRESS, 57., 2006, Valencia. **IAC: Proceedings**. Valencia: 2006.

MOURA, C. A. T.; DOMICIANO, A. J. V.; FERNANDES, M. O. S. Incorporação de software em veículos espaciais brasileiros. In: CONFERÊNCIA INTERNACIONAL DE TECNOLOGIA DE SOFTWARE, 9., 1998, Curitiba. **CITS: Anais**. Curitiba: 1998.

MOURA, C. A. T., LAHOZ, C. H. N., ABDALA, M. A. D. A practical approach to quality assurance in critical system. In: LATIN-AMERICAN DEPENDABLE COMPUTING SYMPOSIUM, 1., 2003, São Paulo. **LADC: Tutorial**. São Paulo: 2003.

MYLOPOULOS, J. A research agenda for the days of software agents. In: INTERNATIONAL I* WORKSHOP, 3. **.Invited talk**. Recife, 2008.

MYLOPOULOS, J.; CHUNG, L.; NIXON, B. Representing and using nonfunctional requirements: a process-oriented approach. **IEEE Transactions on Software Engineering**, v. 18, n. 6, p. 483-497, 1992.

NASA.National Aeronautics and Space Administration, **NASA-GB-8719.13**: Technical standard, software safety guidebook. Washington, 2005. Disponível em: <<http://www.hq.nasa.gov/office/codeq/doctree/871913.pdf>>. Acesso em: 19 out. 2008.

_____.**NASA-JPL**: Report on the loss of the mars polar lander and deep space 2 missions. Technical Report JPL D-18709, Jet Propulsion Laboratory, California Institute of Technology, 2000.

_____.**NASA-MCO**: Mars climate orbiter: mishap investigation board, phase I report. Technical report, NASA Headquarters, Washington, 1999.

NIMMO, I.; NUNNS, S. R.; EDDERSHAW, B. W. Lessons learned from the failure of a computer system controlling a nylon polymer plant. In: SAFETY AND RELIABILITY SOCIETY SYMPOSIUM, 1987, Altrincham. **Proceedings**. Altrincham: 1987.

NIXON, B. Dealing with performance requirements during the development of information systems. In: INTERNATIONAL SYMPOSIUM ON REQUIREMENTS ENGINEERING, 1. 1993, San Diego. **RE: Proceedings**. San Diego: 1993. p. 42-49.

OMG.Object management group. Disponível em: <<http://www.omg.org/>> Acesso em: 26 nov. 2008.

PERINI, A.; SUSI, A. Modeling information technology needs in the agriculture domain. In: TROPOS WORKSHOP, 2001, Trento. **Proceedings**. Trento:Trento University, 2001.

PETIT, M.; ROUSSEAU, A. Using i* for early requirements of a computer system for hospital emergency departments. In: TROPOS WORKSHOP, 2001, Trento. **Proceedings**. Trento:Trento University, 2001.

PISACANE, V. L.; MOORE, R. C. **Fundamentals of space systems**. The Johns Hopkins University/Applied Physics Laboratory. Series in Science and Engineering, New York: Oxford University Press, 1994.

POHL, K. **Process-centered requirements engineering**. Taunton: Research Studies Press Ltd, 1996.

PUMFREY, D.J. **The Principled design of computer system safety analyses**. 2000. 224p. Thesis (Phd) - Department of Computer Science, University of York, York, 2000.

RAUSAND, M. **HAZOP -hazard and operability study, department of production and quality engineering**. In: System Reliability Theory; Models, Statistical Methods and Applications, 2nd ed. Norwegian: Norwegian University of Science and Technology, Wiley, 2004.

REDMILL, F.; CHUDLEIGH, M.; CATMUR, J. **System safety: HAZOP e software HAZOP**. Sussex: John WILEY, 1999. 248 p.

RIBEIRO, T. S. Veículos lançadores de satélites - cenário atual e futuro. **Revista Parcerias Estratégicas**, v. 1, n. 7, outubro, 1999. Disponível em: <<http://ftp.mct.gov.br/CEE/revista/Parcerias7/revista19.PDF>> Acesso em: 20 dez. 2006.

ROBINSON, W. N. Integrating multiple specifications using domain goals. In: INTERNATIONAL WORKSHOP ON SOFTWARE SPECIFICATION AND DESIGN, 5., 1989, Pittsburgh. **IWSSD: Proceedings**. Pittsburgh: 1989, p. 219-226.

ROMANI, M. A. S. **Processo de análise de requisitos de dependabilidade para software espacial**. 2007. 195 p. Dissertação (Mestrado) - Instituto Tecnológico da Aeronáutica, ITA, São José dos Campos, 2007.

ROSS, D.; SCHOMAN, K. Structured analysis for requirements definition. **IEEE Transactions on Software Engineering**, v. 3, n.1, p.6-15, January 1977.

RUS, I.; KOMI-SIRVIO, S.; COSTA, P. **Software dependability properties: a survey of definitions, measures and techniques**. Maryland: High Dependability Computing Program (HDCP). Fraunhofer Center for Experimental Software Engineering, 2003. (Technical Report 03-110).

SAE.Society of Automotive Engineers. **SAE-J-1739**: Potential failure mode and effects analysis in design (design FMEA) and potential failure mode and effects analysis in manufacturing and assembly processes (process FMEA), 2000. (Reference Manual).

SERVIÇO PÚBLICO FEDERAL.Ministério da Defesa. Comando da Aeronáutica. Departamento de Pesquisas e Desenvolvimento. **Relatório da investigação do acidente ocorrido com o VLS1-v03, em 22 de agosto de 2003, em Alcântara, Maranhão**. São José dos Campos, 2004.

SOMMERVILLE, I. **Software engineering**. 6th. ed. Essex: Addison Wesley, 2001. 693 p.

SOUZA, E. A. **O treinamento industrial e a gerência de riscos - uma proposta de instrução programada**. 1995. 38 p. Dissertação (Mestrado) - Universidade Federal de Santa Catarina, 1995.

SPEM, **Software process engineering metamodel specification**. version 1.0, Needham: Object Management Group, formal 02-11-14, Nov. 2002.

SPICE.Software process improvement and capability determination. software process assessment – part 2, a model for process management, **SPICE** Version 1.00, 1995.

STARK, J.; SWINERD, G.; TATNALL, A. **Introduction**. In: FORTESCUE P.; STARK J.; SWINERD G. In: Spacecraft systems engineering. 3rd ed., Wiley England, 2004. p. 2-8./ cap. 1/

STOREY, N. **Safety-critical computer systems**. Essex: Addison Wesley, 1996. 453 p.

TATNALL, A. R. L.; FARROW J. B; FRANCIS R. Cap. 19, Spacecraft system engineering. In: FORTESCUE P.; STARK J.; SWINERD G. **Spacecraft systems engineering**. Third Edition, Wiley England: 2004. p. 613-659.

VAN LAMSWEERDE, A. Goal-oriented requirements engineering: a guided tour. In: IEEE INTERNATIONAL SYMPOSIUM ON REQUIREMENTS ENGINEERING, 5., 2001, Toronto. **ISRE: Proceedings**. Toronto: 2001.

_____. **Goal-oriented requirements engineering: from system objectives to UML models to precise software specifications**. University of Louvain, Louvain-la-Neuve, 2003. /Relatório/

_____. Requirements engineering in the year 00: a research perspective. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 2000, Limerick. **ICSE: Proceedings**. Limerick: 2000. p. 5-19.

VAN LAMSWEERDE, A.; DARIMONT, R; MASSONET, P. Goal-directed elaboration of requirements for a meeting scheduler: problems and lessons learnt. In: INTERNATIONAL SYMPOSIUM ON REQUIREMENTS ENGINEERING, 2. 1995, York. **RE: Proceedings**. York: 1995. p. 194-203.

VILLAS BOAS, D. J. F. O Contexto histórico das atividades espaciais e a tecnologia dos foguetes, salto para o futuro. **Boletim Eletrônico TVE Brasil**, 2006, ISSN 1982-0283, 2006. Disponível em: <<http://www.tvebrasil.com.br/salto/boletins2006/te/tetxt2.htm>> Acesso em: 18 Maio 2008.

WEBER, T. S. **Um roteiro para exploração dos conceitos básicos de tolerância a falhas**. 2002. 62 p. Material didático do Instituto de Informática – Universidade Federal do Rio Grande do Sul - UFRGS, 2002.

WEISS, G. Agent orientation in software engineering. **Knowledge Engineering Review**, v. 16, n. 4, p. 349-373, 2002

YELLOWBOOK3, **Engineering safety management, issue 3 – fundamentals and guidance**. London: Railtrack House, 2000.

YIN, R. K. **Case study research: design and methods**. Beverly Hills: Calif: Sage Publications, 1984.

YU, E. **Modelling strategic relationship for process reengineering**. Toronto. Thesis (Phd) - Computer Science Department, University of Toronto, Toronto, 1995.

_____. Towards modeling and reasoning support for early-phase requirements engineering. In: IEEE INTERNATIONAL SYMPOSIUM ON REQUIREMENTS ENGINEERING, 3. 1997, Washington. **RE: Proceedings**. Washington: 1997. p. 226-235.

YU, E.; MYLOPOULOS, J. M. Understanding “why” in software process modelling, analysis and design. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 16. 1994. Sorrento. ICSE: **Proceedings**. Sorrento: 1994. p.159-168.

YUE, K. What does it mean to say that a specification is complete? In: INTERNATIONAL WORKSHOP ON SOFTWARE SPECIFICATION AND DESIGN, 4. 1987, Monterrey. **IWSSD: Proceedings**. Monterrey: 1987.

ZAINAL, Z. Case study as a research method. **Jurnal Kemanusiaan** v.9, Jun 2007. Disponível em: <http://www.fppsm.utm.my/jurnal/JK9D06/JK9_WANKHAIRUZZAMANWANISMAIL.pdf> Acesso em: 20 jun 08.

ZAVE, P. Classification of research efforts in requirements engineering. **ACM Computing Surveys**, v. 29, n. 4, p.315-321, 1997.

REFERÊNCIAS BIBLIOGRÁFICAS COMPLEMENTARES

AXELSSON, L.; DURBIN, N.; TAEI, I.. **Safety culture**, In: INTERNATIONAL SYSTEM SAFETY CONFERENCE, 23., Technical Panel on Safety Culture. San Diego: 2005. /Arquivo eletrônico/

BEKEY, I. **Advanced space system concepts and technologies**. American Institute of Aeronautics & Astronautics, California, April 2003./Relatório técnico/

CAMARGO JÚNIOR, J. B. **Metodologia de análise de risco em sistemas computacionais de aplicação crítica**. 2002, 151 p. Tese (Livre Docência) - Escola Politécnica da Universidade de São Paulo, São Paulo, 2002.

CARLEIAL A. B. Uma breve história da conquista espacial. **Revista Parcerias Estratégicas**, v. 1, n. 7, Out. 1999. Disponível em: <<http://ftp.mct.gov.br/CEE/revista/rev07.htm>> Acesso em: 20 dez. 2006.

ECSS.EUROPEAN COOPERATION for SPACE STANDARDIZATION **ECSS-Q-80-03**: Space product assurance: methods and techniques to support the assessment of software dependability and safety. Noordwijk, 2006. /Draft/

KAUPPINEN, M.; KUJALA, S.; ALTIO, T.; LEHTOLA, L. Introducing requirements engineering: how to make a cultural change happen in practice. In: IEEE JOINT INTERNATIONAL CONFERENCE ON REQUIREMENTS ENGINEERING, Essen, 2002. **Proceedings**. Essen: 2002. p. 43-51.

KAVAKLI, E., LOUCOPOULOS, P., Goal Driven Requirements Engineering: Evaluation of Current Methods, **Requirements Engineering. Journal**. London, v. 6, n. 4, p. 237-251, 1999.

KRIGE, J.; RUSSO, A.; SEBESTA, L. **A history of the european space agency, the story of ESA, 1973 to 1987**, v. 2. European Space Agency, Noordwijk, 2000.

MCDERMID J.A.; PUMFREY, D.J. Safety analysis of hardware / software interactions in complex systems, In: INTERNATIONAL SYSTEM SAFETY CONFERENCE, 16., 1998, Seattle. **ISSC: Proceedings**. Seattle: 1998. p. 232-241

NASA. National Aeronautics and Space Administration Software Safety Guidebook, NASA TECHNICAL STANDARD, March 31, 2004. Disponível em <<http://www.hq.nasa.gov/office/codeq/doctree/871913.htm>> Acesso em: 06 nov. 2006.

YU E.; LIU L.; LI Y. Modelling strategic actor relationships to support intellectual property management. In: INTERNATIONAL CONFERENCE ON CONCEPTUAL MODELLING, 20., 2001, Yokohama. **Proceedings**. Yokohama: 2001.