



Ministério da
Ciência e Tecnologia



sid.inpe.br/mtc-m19/2011/04.04.12.08-TDI

AUXÍLIO COMPUTACIONAL A UM PROCESSO DE ENGENHARIA SIMULTÂNEA DE SISTEMAS ESPACIAIS

Jonas Bianchini Fulindi

Dissertação de Mestrado do Curso de Pós-Graduação em Engenharia e Tecnologia
Espaciais / Gerenciamento de Sistemas Espaciais, orientada pelo Dr. Geilson
Loureiro, aprovada em 31 de março de 2011.

URL do documento original:

<<http://urlib.net/8JMKD3MGP7W/39EQGDP>>

INPE
São José dos Campos
2011

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 - CEP 12.245-970

São José dos Campos - SP - Brasil

Tel.:(012) 3208-6923/6921

Fax: (012) 3208-6919

E-mail: pubtc@sid.inpe.br

CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO INTELLECTUAL DO INPE (RE/DIR-204):**Presidente:**

Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)

Membros:

Dr^a Inez Staciarini Batista - Coordenação Ciências Espaciais e Atmosféricas (CEA)

Dr^a Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação

Dr^a Regina Célia dos Santos Alvalá - Centro de Ciência do Sistema Terrestre (CST)

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Dr. Ralf Gielow - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

Dr. Wilson Yamaguti - Coordenação Engenharia e Tecnologia Espacial (ETE)

Dr. Horácio Hideki Yanasse - Centro de Tecnologias Especiais (CTE)

BIBLIOTECA DIGITAL:

Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Deicy Farabello - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

EDITORAÇÃO ELETRÔNICA:

Vivéca Sant´Ana Lemos - Serviço de Informação e Documentação (SID)



Ministério da
Ciência e Tecnologia



sid.inpe.br/mtc-m19/2011/04.04.12.08-TDI

AUXÍLIO COMPUTACIONAL A UM PROCESSO DE ENGENHARIA SIMULTÂNEA DE SISTEMAS ESPACIAIS

Jonas Bianchini Fulindi

Dissertação de Mestrado do Curso de Pós-Graduação em Engenharia e Tecnologia
Espaciais / Gerenciamento de Sistemas Espaciais, orientada pelo Dr. Geilson
Loureiro, aprovada em 31 de março de 2011.

URL do documento original:

<<http://urlib.net/8JMKD3MGP7W/39EQGDP>>

INPE
São José dos Campos
2011

Dados Internacionais de Catalogação na Publicação (CIP)

Fulindi, Jonas Bianchini.

F957a Auxílio computacional a um processo de engenharia simultânea de sistemas espaciais / Jonas Bianchini Fulindi. – São José dos Campos : INPE, 2011.

xxviii+245 p. ; (sid.inpe.br/mtc-m19/2011/04.04.12.08-TDI)

Dissertação (Mestrado em Engenharia e Tecnologia Espaciais/ Gerenciamento de Sistemas Espaciais) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2011.

Orientador : Dr. Geilson Loureiro.

1. Engenharia de sistemas. 2. Engenharia simultânea. 3. Desenvolvimento de produtos complexos. I. Título.

CDU 629.783

Copyright © 2011 do MCT/INPE. Nenhuma parte desta publicação pode ser reproduzida, armazenada em um sistema de recuperação, ou transmitida sob qualquer forma ou por qualquer meio, eletrônico, mecânico, fotográfico, reprográfico, de microfilmagem ou outros, sem a permissão escrita do INPE, com exceção de qualquer material fornecido especificamente com o propósito de ser entrado e executado num sistema computacional, para o uso exclusivo do leitor da obra.

Copyright © 2011 by MCT/INPE. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, microfilming, or otherwise, without written permission from INPE, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use of the reader of the work.

Aprovado (a) pela Banca Examinadora
em cumprimento ao requisito exigido para
obtenção do Título de **Mestre** em

**Engenharia e Tecnologia
Espaciais/Gerenciamento de Sistemas
Espaciais**

Dr. Walter Abrahão dos Santos

Presidente / INPE / São José dos Campos - SP

Dr. Geilson Loureiro

Orientador(a) / INPE / São José dos Campos - SP

Dr. Germano Kienbaum

Membro da Banca / INPE / São José dos Campos - SP

Dr. Luis Gonzaga Trabasso

Convidado(a) / ITA / São José dos Campos - SP

Este trabalho foi aprovado por:

maioria simples

unanimidade

Aluno (a): **Jonas Bianchini Fulindi**

São José dos Campos, 31 de março de 2011

“Viste a um homem diligente na sua obra? perante reis será posto: não será posto perante os de baixa sorte”.

Provérbios 22.29

A Rafael Bianchini Fulindi.

AGRADECIMENTOS

Agradeço a Deus por me conceder saúde, vida e inspiração, para ver cores onde não houvesse cores, por ouvir música onde não houvesse música e por apreciar a cada manhã, em que eu me levantava e seguia em frente.

Ao amor e dedicação de meu pai Edivaldo e da minha mãe Marisa, e de uma grande fonte de inspiração, meu irmão Rafael. E a toda a minha família.

A meus tios Walter e Cida, por terem me acolhido tão afetivamente em sua casa em Jacareí, antes que eu me mudasse para São José dos Campos.

À CAPES pela bolsa de estudos, pois sem ela não teria condições de realizar este trabalho.
À Dra. Maria do Carmo e a todos os Professores do curso CSE.

A meu orientador, Dr. Geilson Loureiro pela orientação neste trabalho, pelas dúvidas esclarecidas, e pela oportunidade de trabalhar com ele. Muito obrigado Dr. Geilson.

A todos os amigos que conheci no INPE, pessoas especiais e que eu muito as admiro e respeito.

Aos meus amigos de São José dos Campos.

À Sonia Sousa por toda a ajuda e apoio, sempre me fez bem falar com ela.

Aos colegas do Instituto de Aeronáutica e Espaço, e da divisão de sistemas espaciais ASE-IAE.

Aos amigos Artur, Denis e Kátia.

Ao colega do INPE Adalberto Coelho, ao Márcio Branco e ao Lino pelo auxílio e comentários sobre o trabalho. Aos colegas com quem trabalhei próximo no prédio Satélite, e sempre estavam prontos a me auxiliar e tirar minhas dúvidas: o Mário Baruel e o Renato Magalhães.

Ao Renato Calado pela avaliação e comentários sobre este trabalho, e pelas discussões muito proveitosas, seja engenharia, ou seja música.

A um grande amigo que fiz, e uma grande personalidade, admirável o seu jeito de encarar a vida, dessa forma muito me ajudou a superar diversas dificuldades. Muito obrigado Fábio Dall Cortivo.

Ao amigo Leandro Casagrande pelas discussões diversas e edificadoras.

Aos meus amigos da CAP, Luís, Sóstenes, Rudinei, Laurita, Leonardo, Marlon, Dalila e todos os demais.

A Simone Del Ducca pela amizade e ajuda.

E por último, meu agradecimento ao Dr. Loures, uma pessoa que admiro. Muito obrigado pela confiança, pelas oportunidades e pelas palavras sempre positivas e de apoio. Muito obrigado Dr. Loures.

RESUMO

Esta dissertação versa sobre o auxílio computacional a um processo de engenharia simultânea de sistemas espaciais. O processo a ser auxiliado por computador é baseado no *framework* e método desenvolvido pelo Professor Geilson Loureiro em sua tese de doutorado e evoluído desde então. O método auxiliado por computador aqui proposto foi demonstrado para produtos espaciais e aeroespaciais, mas da mesma forma que o processo não se restringe a esses produtos, o processo auxiliado por computador por sua vez também não. Sendo assim, o auxílio computacional proposto nessa dissertação também pode ser aplicado a produtos complexos em geral. O processo é composto pela análise de *stakeholders*, a análise de requisitos, a análise funcional e a análise de implementação, dos quais são destacados os elementos a serem capturados por um software ambiente de engenharia de sistemas comercial para a descrição de produto e organização, o que propõe a engenharia simultânea de sistemas, capturar os elementos que tornem o desenvolvimento de um sistema composto de elementos de produto e organização de maneira integrada e simultânea. Ferramentas computacionais de engenharia de sistemas auxiliam em partes das etapas do desenvolvimento da engenharia de sistemas. No entanto este trabalho ilustra uma ferramenta computacional que cobre o processo de engenharia simultânea de sistemas. Este trabalho contribui com a modelagem de organização, pois adapta notações para capturar os elementos de organização e os elementos que trocam energia, material e informação com ela, e suas interconexões. Também contribui ao assumir a análise de *stakeholders* como parte da solução sistema e ao capturar os *stakeholders* e suas declarações, adaptando uma notação para isso. O processo de engenharia de sistemas já considera diversas notações de modelagem para descrever o produto em operação e a organização de desenvolvimento. O processo de engenharia simultânea de sistemas, no entanto, ao considerar todos os processos do ciclo de vida do produto necessita de muitas notações de modelagem para a descrição da solução sistema, ainda manter a consistência entre seus diversos níveis de abstração, o que não seria viável sem o auxílio computacional.

COMPUTER AID TO A SPACE SYSTEMS CONCURRENT ENGINEERING PROCESS

This dissertation aims to present a computer aided process for concurrent engineering of space systems. The computer aided process is based on a framework and a method developed by Professor Geilson Loureiro in his doctorate thesis. This method and framework has evolved over the years. The method aided by computer, proposed in this work, was applied to space and aerospace products. The process is not restricted to these products and the same can be said of the computer aided process. So, the proposed computer aided process in this work also can be applied to other complex products. The process is carried out by the stakeholders analysis, requirements analysis, functional analysis and implementation analysis. The elements of this process are captured by a commercial systems engineering software to describe the product and the organization. This is the approach of the systems concurrent engineering: model product and organization elements in order to develop concurrently an integrated system. Systems engineering tools provide aids in parts of the systems engineering development. However this work shows a computational tool that cover the systems concurrent engineering process. This work contributes with organization modeling, adapting notations to capture the organization elements and the elements that exchange energy, material and information and its interconnections. This work also contributes to stakeholders analysis as part of the system solution capturing the stakeholders and their concerns. A notation is adapted for it. Systems engineering already considers many diagrams to describe the product in operation and the development organization. Systems concurrent engineering considers all life cycle processes and its scenarios. Therefore it is necessary many more diagrams to describe the system solution and keep the consistency in all layers of the abstraction. It would not be possible without the computer aided process.

LISTA DE FIGURAS

	<u>Pág.</u>
Figura 2.1 - Processo de engenharia de sistemas da NASA.....	12
Figura 2.2 - Processo de engenharia de sistemas da ECSS.....	13
Figura 2.3 - Processo de engenharia de sistemas do IEEE.....	14
Figura 2.4 - Processos para fazer a engenharia de um sistema.....	15
Figura 2.5 - Tabela comparativa das ferramentas disponibilizadas no site do INCOSE.....	25
Figura 2.7 - Framework de visão total.....	29
Figura 2.8 - Método de análise simultânea estruturado.....	31
Figura 3.1 - Framework de visão total a partir da evolução de Loureiro (1999).....	34
Figura 3.2 - Método dentro do <i>framework</i> de visão total.....	35
Figura 3.3 - Elementos do processo de engenharia simultânea de sistemas.....	37
Figura 3.5 - Exemplo estilizado de uma sequência de processos.....	39
Figura 3.7 - Cenários que serão realizados pela organização.....	41
Figura 3.8 - <i>Stakeholders</i> de produto para o produto em operação, e para o produto em outro processo do ciclo de vida ou cenário.....	42
Figura 3.9 - <i>Stakeholders</i> de organização.....	43
Figura 3.10 - Interesses de <i>stakeholders</i> de produto.....	44
Figura 3.11 - Interesses de <i>stakeholders</i> de organização.....	46
Figura 3.12 - Medidas de efetividade para produto.....	48
Figura 3.13 - Medidas de efetividade para organização.....	48
Figura 3.14 - Lista de eventos para produto.....	49
Figura 3.15 - Lista de eventos para organização.....	49
Figura 3.16 - Requisitos de sistema para produto e para organização.....	50
Figura 3.17 - Contexto funcional do sistema para produto.....	51
Figura 3.18 - Circunstâncias nos cenários de produto.....	52
Figura 3.19 - Contexto funcional de organização.....	53
Figura 3.20 - Circunstâncias nos cenários de organização.....	53
Figura 3.21 - Lista de eventos para identificar funções de produto e organização.....	55
Figura 3.22 - Diagrama de estrutura funcional de produto.....	56
Figura 3.23 - Diagrama de transição de estados de produto.....	57
Figura 3.24 - Tabela FMECA.....	58
Figura 3.25 - Diagrama de estrutura funcional de organização.....	60
Figura 3.26 - Diagrama de comportamento funcional de organização.....	61
Figura 3.27 - Contexto de implementação de produto.....	62
Figura 3.28 - Contexto de implementação de organização.....	63
Figura 3.29 - Interconexão de arquitetura de implementação de produto.....	64
Figura 3.30 - Fluxos de arquitetura de implementação de produto.....	65
Figura 3.31 - Interconexão de arquitetura de implementação de organização.....	66
Figura 3.32 - Fluxos de arquitetura de implementação de organização.....	67
Figura 4.1 - Modelo de rastreabilidade do projeto.....	75
Figura 4.2 - Item de informação gerado no <i>Cradle</i>	76
Figura 4.3 - Associação entre itens de informação por <i>links</i>	76

Figura 4.5 - Declaração da missão do sistema.....	79
Figura 4.7 - Desdobramento de processo em cenários.....	80
Figura 4.9 - Elementos da análise de requisitos associados por itens de informação. ...	82
Figura 4.12 - Captura dos estados e circunstâncias.....	85
Figura 4.13 - Elementos físicos da análise de implementação.....	86
Figura 5.1 - Processos do ciclo de vida do produto.....	89
Figura 5.2 - Cenário de desenvolvimento de projeto e construção.	89
Figura 5.3 - Cenário de montagem e teste.....	90
Figura 5.4 - Cenário de operação para produto.	90
Figura 5.5 - Escopo do esforço de desenvolvimento.....	91
Figura 5.6 - <i>Stakeholders</i> de produto e seus interesses para o contexto de teste.....	92
Figura 5.7 - <i>Stakeholders</i> de produto e seus interesses para o cenário de operação.	93
Figura 5.8 - <i>Stakeholders</i> de organização e seus interesses para o cenário de desenvolvimento do projeto e construção.	93
Figura 5.9 - <i>Stakeholders</i> de organização e seus interesses para o cenário de montagem.	94
Figura 5.10 - Item de informação gerado para <i>stakeholder</i>	95
Figura 5.11 - Item de informação gerado para interesse de <i>stakeholder</i>	95
Figura 5.12 - <i>Link</i> entre elemento do diagrama e itens de informação.	96
Figura 5.13 - Descrição de informação do fluxo.....	97
Figura 5.14 - Item de informação que representa as MoEs.....	98
Figura 5.15 - Visualização dos itens de informação associados por <i>links</i>	99
Figura 5.17 - Atributos dos <i>links</i> que associam os itens de informação.....	100
Figura 5.18 - Itens de informação da análise de <i>stakeholders</i> listados em uma tabela.	101
Figura 5.19 - Documento gerado a partir da tabela.	101
Figura 5.20 - Contexto funcional de produto para o cenário de operação.....	103
Figura 5.21 - Desdobramento do contexto funcional de produto para o cenário de operação.....	104
Figura 5.22 - Contexto funcional de produto para o cenário de teste.....	105
Figura 5.23 - Desdobramento do contexto funcional do produto para o cenário de teste.	106
Figura 5.24 - Contexto funcional de organização para o contexto de desenvolvimento do projeto e construção.....	107
Figura 5.25 - Desdobramento do contexto funcional de organização para o contexto de desenvolvimento do projeto e construção.	108
Figura 5.26 - Desdobramento de ‘definir modelos’.	108
Figura 5.27 - Desdobramento de ‘definir requisitos de teste’.	109
Figura 5.28 - Contexto funcional de organização para o cenário de montagem.	109
Figura 5.29 - Desdobramento do contexto funcional de organização para o cenário de montagem.	110
Figura 5.30 - Funções que devem ser desempenhadas pelo produto em operação.	110
Figura 5.31 - Campo de informação da função de produto.....	111
Figura 5.32 - Item de informação de risco para produto.	112
Figura 5.33 - Item de informação de risco associado à função ‘abrir painel solar’.....	113
Figura 5.34 - Campos de informação da função de organização.....	113
Figura 5.35 - Item de informação de risco para organização.	114

Figura 5.36 - Item de informação de risco associado à função da organização.	115
Figura 5.37 - Arquitetura de implementação de produto.....	116
Figura 5.38 - Arquitetura de implementação de produto e interface.....	117
Figura 5.39 - Arquitetura de implementação de organização e interfaces.	117
Figura 5.40 - Arquitetura de implementação de organização.....	118
Figura 5.41 - Diagrama de arquitetura do produto em operação.....	121
Figura 5.42 - Diagrama de arquitetura do produto em teste.....	122
Figura 5.43 - Diagrama de arquitetura da organização de desenvolvimento.	122
Figura 5.44 - Diagrama de arquitetura da organização de montagem.....	123
Figura 6.1 - Processos do ciclo de vida do produto.....	129
Figura 6.2 - Cenário de desenvolvimento do projeto.	129
Figura 6.3 - Cenário ‘integrar modelo de voo’.....	130
Figura 6.4 - Cenários de preparar para o lançamento e operar.....	130
Figura 6.5 - Escopo do esforço de desenvolvimento.....	131
Figura 6.7 - <i>Stakeholders</i> de produto e seus interesses para o cenário de operação do VLM.	133
Figura 6.10 - Itens de informação de endereçamento de stakeholders.....	135
Figura 6.11 - Itens de informação de interesses de stakeholders.....	136
Figura 6.12 - Medidas de efetividade.	136
Figura 6.13 - Referências cruzadas entre itens de informação de MoE’s.	137
Figura 6.14 - Exemplo de detalhes de link.	138
Figura 6.15 - Exibição de stakeholder, interesse e requisito de stakeholder.....	139
Figura 6.16 - Exemplo de uma tabela gerada em extensão RTF.....	140
Figura 6.18 - Desdobramento do contexto do produto no cenário ‘integrar VLM’.....	142
Figura 6.19 - Contexto funcional do produto em operação.....	143
Figura 6.20 - Desdobramento do contexto do produto em operação.....	144
Figura 6.21 - Contexto funcional de organização de desenvolvimento.	145
Figura 6.22 - Contexto funcional de organização de ‘preparar para o lançamento’... ..	146
Figura 6.23 - Desdobramento do contexto funcional de organização de ‘preparar para lançar’.....	147
Figura 6.25 - Risco associado a uma função.	148
Figura 6.26 - Descrição da função modelo do produto.	149
Figura 6.28 - Link de risco à função.....	151
Figura 6.29 - Risco organização de preparar para lançar, transportar estágios.....	151
Figura 6.30 - Link à função de organização.	152
Figura 6.31 - Interconexão de arquitetura do produto.....	153
Figura 6.32 - Fluxo de arquitetura de produto.....	154
Figura 6.33 - Contexto de implementação de organização de engenharia de sistemas.....	155
Figura 6.34 - Interconexão de arquitetura da organização de preparar para o lançamento.	156
Figura 6.35 - Arquitetura de produto em operação.	158
Figura 6.36 - Arquitetura do produto em integração.....	158
Figura 6.37 - Arquitetura da organização de desenvolvimento do projeto e construção.	159
Figura 6.38 - Arquitetura da organização de preparar para lançar.	159
Figura A.1 - Cobertura do ciclo de vida pelo <i>Cradle</i>	179

Figura A.2 - Módulos do <i>Cradle</i>	181
Figura A.3 - Itens associados por <i>links</i>	182
Figura A.4 - Exemplo de árvore, diagramas de hierarquia e de tabelas.	183
Figura A.5 - Captura de requisitos usando <i>Cradle</i>	184
Figura A.6 - Rastreabilidade desde requisitos ao modelo de implementação	185
Figura A.7 - Modelagem de análise em diferentes níveis de abstração.	186
Figura A.8 - Modelagem do contexto do projeto.	187

LISTA DE TABELAS

	<u>Pág.</u>
Tabela 5.1 - Atributos de produto e organização.	119
Tabela 6.1 - Atributos de produto e organização.	157

LISTA DE SIGLAS E ABREVIATURAS

CBERS	<i>China-Brazil Earth Resource Satellite</i>
IAE	<i>Instituto de Aeronáutica e Espaço</i>
VLM	<i>Veículo Lançador de Microsatélites</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
NASA	<i>National Aeronautics and Space Administration</i>
ECSS	<i>European Cooperation for Space Standardization</i>
EIA	<i>Electronic Industries Alliance</i>
DFA	<i>Design for Assembly</i>
DFM	<i>Design for Manufacturing</i>
DFI	<i>Design for Inspectability</i>
DFR	<i>Design for Reliability</i>
DFQ	<i>Design for Quality</i>
CDF	<i>Concurrent Design Facility</i>
BPMN	<i>Business Process Model and Notation</i>
SysML	<i>Systems Modeling Language</i>
MoE's	<i>Measurment of Effectiveness</i>
DFD	<i>Data Flow Diagram</i>
ERD	<i>Entity Relationship Diagram</i>
STD	<i>State Transition Diagram</i>
DSD	<i>Data Structure Diagram</i>
IDEF0	<i>Integrated Definition for Functional Modeling</i>
eFFBD	<i>Enhanced Functional Flow Block Diagram</i>
PFD	<i>Process Flow Diagram</i>
UCD	<i>Use Case Diagram</i>
PD	<i>Package Diagram</i>
SQD	<i>Sequence Diagram</i>
COD	<i>Colaboration Diagram</i>
CD	<i>Class Diagram</i>
SCD	<i>Statechart Diagram</i>
ACD	<i>Activity Diagram</i>
AID	<i>Architecture Interconnect Diagram</i>
PAD	<i>Physical Architecture Diagram</i>
CPD	<i>Component Diagram</i>
DPD	<i>Deployment Diagram</i>
STC	<i>Structure Chart</i>
SAD	<i>Software Architecture</i>
MGSE	<i>Mechanical Ground Support Equipment</i>
GSE	<i>Ground Support Equipment</i>

SUMÁRIO

	<u>Pág.</u>
1 INTRODUÇÃO	1
1.1 Motivação	2
1.2 Objetivo	5
1.3 Metodologia	5
1.4 Estrutura do Trabalho.....	6
2 FUNDAMENTAÇÃO TEÓRICA	9
2.1 Desenvolvimento Integrado de Produto.....	9
2.2 Processos de Engenharia de Sistemas	10
2.2.1 NASA.....	10
2.2.2 ECSS.....	12
2.2.3 IEEE	13
2.2.4 EIA.....	14
2.3 Engenharia Simultânea.....	16
2.4 <i>TEAM X</i>	17
2.5 CDF	18
2.6 Modelagem de Produto	20
2.7 Modelagem de Organização	20
2.8 Modelagem Integrada de Produto e Organização.....	21
2.9 BPMN	22
2.10 SysML	23
2.11 Ferramentas disponíveis	23
2.12 <i>Framework</i> e Método para Desenvolvimento de Produto Complexo	27
2.12.1 <i>Framework</i> de Visão Total.....	28
2.12.2 Método de Análise Estruturada Simultânea	29
3 ENGENHARIA SIMULTÂNEA DE SISTEMAS	33
3.1 Framework	33
3.2 Método	34

3.3 Descrição do Processo de Engenharia Simultânea de Sistemas.....	37
3.3.1 Missão.....	38
3.3.2 Processos do Ciclo de Vida.....	38
3.3.3 Cenários.....	40
3.3.4 Cenários Dentro do Escopo do Esforço de Desenvolvimento.....	40
3.3.5 <i>Stakeholder</i> de Produto.....	41
3.3.6 <i>Stakeholder</i> de Organização.....	43
3.3.7 Interesses de <i>Stakeholder</i> de Produto.....	44
3.3.8 Interesses de <i>Stakeholder</i> de Organização.....	45
3.3.9 MoEs e Requisitos de <i>Stakeholder</i> de Produto e de Organização.....	46
3.3.10 Requisitos de Sistema.....	50
3.3.11 Contexto Funcional de Produto e Modos.....	51
3.3.12 Contexto Funcional de Organização e Modos.....	52
3.3.13 Estrutura Funcional do Produto.....	54
3.3.14 Comportamento Funcional do Produto.....	56
3.3.15 Análise de Perigo e Risco para Produto e Organização.....	57
3.3.16 Atributos Funcionais de Produto e Organização.....	59
3.3.17 Estrutura Funcional de Organização.....	59
3.3.18 Comportamento Funcional de Organização.....	60
3.3.19 Contexto de Implementação de Produto.....	61
3.3.20 Contexto de Implementação de Organização.....	62
3.3.21 Requisitos de Interface Externa de Produto.....	63
3.3.22 Requisitos de Interface Externa de Organização.....	63
3.3.23 Interconexões de Arquitetura de Implementação de Produto.....	63
3.3.24 Fluxos de Arquitetura de Implementação de Produto.....	64
3.3.25 Atributos de Implementação de Produto e Organização.....	65
3.3.26 Interconexão de Arquitetura de Implementação de Organização.....	66
3.3.27 Fluxos de Arquitetura de Implementação de Organização.....	67
3.3.28 Relacionamento entre Atributos.....	67
3.3.29 Arquitetura de Sistemas.....	68

3.4	Relacionamento entre os Processos de Análise e os Elementos do Método	68
3.5	Consistência entre o <i>Framework</i> e o Método	70
4	AUXÍLIO COMPUTACIONAL AO PROCESSO	71
4.1	<i>Cradle</i>	71
4.2	Método	73
4.2.1	Missão, Processos do Ciclo de Vida e Cenários	78
4.2.2	Análise de <i>Stakeholders</i> e Requisitos	81
4.2.3	Análise Funcional	82
4.2.4	Análise de Implementação	85
5	APLICAÇÃO DO MÉTODO PARA O CBERS	87
5.1	CBERS II	87
5.1.1	Descrição da Missão	88
5.1.2	Ciclo de Vida e Escopo do Esforço de Desenvolvimento	88
5.1.3	Análise de Stakeholder e de Requisitos	91
5.1.4	Análise Funcional	102
5.1.5	Análise de Implementação	115
5.1.6	Atributos	118
6	APLICAÇÃO DO MÉTODO PARA O VLM	127
6.1	VLM	127
6.1.1	Descrição da Missão	127
6.1.2	Ciclo de Vida e Escopo do Esforço de Desenvolvimento	128
6.1.3	Análise de <i>Stakeholder</i> e Análise de Requisitos	131
6.1.4	Análise Funcional	140
6.1.5	Análise de Implementação	152
6.1.6	Atributos	156
7	DISCUSSÃO	161
7.1	Evolução da Engenharia Simultânea de Sistemas com e sem Auxílio computacional	161
7.2	Melhoria no Desenvolvimento Integrado de Produto e Engenharia de Sistemas	162

7.3 <i>Team X</i> , CDF e a Engenharia Simultânea de Sistemas.....	162
7.4 Melhoria na Modelagem de Produto e Organização	163
7.5 BPMN, SysML e a Engenharia Simultânea de Sistemas	163
7.6 Auxílio Computacional ao Processo de Engenharia Simultânea de Sistemas.....	164
7.7 Vantagens do <i>Cradle</i>	164
7.8 Auxílio Computacional ao Processo de Engenharia Simultânea de Sistemas - CBERS	166
7.9 Auxílio Computacional ao Processo de Engenharia Simultânea de Sistemas - VLM	167
8 CONCLUSÃO.....	169
8.1 Objetivos Atingidos.....	169
8.2 Contribuições	170
8.3 Sugestões para Trabalhos Futuros	172
REFERÊNCIAS BIBLIOGRÁFICAS	173
APÊNDICE A - FUNCIONALIDADES DO CRADLE	179
A.1 Cobertura do ciclo de vida	179
A.2 Escopo do <i>Cradle</i> versão 6.0.....	180
A.3 Módulos do <i>Cradle</i>	180
A.3.1 PDM.....	181
A.3.1.1 Referência Cruzada	182
A.3.2 REQ	183
A.3.3 MET	185
A.3.4 SYS.....	186
A.3.4.1 Modelagem de análise	186
A.3.4.2 Contexto de Modelagem de Projeto.....	187
A.3.5 PERF	187
A.3.6 SWE.....	188
A.3.7 DOC	189
A.3.8 WEBP	189
A.3.9 WEBA	190

ANEXO A - CONCURRENT ENGINEERING OF AN ELECTRICAL GROUND SUPPORT EQUIPMENT FOR A SATELLITE ON-BOARD COMPUTER SYSTEM.....	191
ANEXO B - CONCURRENT SYSTEMS ENGINEERING OF A MICROSATELLITE LAUNCH VEHICLE	211
ANEXO C - DESENVOLVIMENTO INTEGRADO DE SISTEMAS - MICROONDAS	225

1 INTRODUÇÃO

Produtos espaciais são produtos complexos. Produtos complexos necessitam de uma abordagem de engenharia de sistemas para o seu desenvolvimento.

A engenharia de sistemas tradicional foca no produto e no processo de desenvolvimento. Mesmo com esse foco restrito, a engenharia de sistemas já necessita de auxílio computacional para, por exemplo: a realização de análise de requisitos; rastreabilidade entre *stakeholders*, requisitos, funções e implementação; gestão de configuração; modelagem descritiva e computacional; consistência entre os modelos.

Quando a engenharia de sistemas passa a ser realizada levando em conta o sistema em cada processo do seu ciclo de vida, e quando essa engenharia de sistemas reconhece que a solução é composta não somente por elementos de produto, mas também por elementos de organização, a complexidade do próprio processo de engenharia de sistemas aumenta.

Mais modelos são necessários e a garantia de consistência entre eles só passa a ser viável com uma ferramenta computacional. Mais itens precisam ser rastreados. Relacionamentos de impacto passam a ser necessários, para capturar as justificativas para o estabelecimento de requisitos ou de um determinado conceito de solução.

A engenharia de sistemas realizada levando em conta o sistema em seus diversos processos do ciclo de vida é o que esta dissertação chama de engenharia simultânea de sistemas. Uma abordagem de engenharia simultânea captura requisitos dos processos do ciclo de vida e antecipa-os para as etapas iniciais do processo de desenvolvimento. E é exatamente isso que se viabiliza para sistemas quando a engenharia de sistemas é realizada pensando, no início do processo de desenvolvimento, o sistema em cada processo do ciclo de vida. Isso permite evitar retrabalho nessas etapas posteriores, reduzir custos, cumprir e reduzir prazos e reduzir riscos

programáticos. Em se tratando de produtos espaciais ou aeroespaciais, evitar retrabalhos nas etapas de integração ou de preparação para o lançamento, por exemplo, implica em não perder uma janela de lançamento e viabilizar a concretização de uma missão planejada muitos anos antes. Dado o nível de especialização do pessoal envolvido e os valores envolvidos em um programa espacial, a redução dos custos esperada é da ordem de milhões de dólares.

O esforço de modelagem de produto e de organização, a quantidade de informação na forma de requisitos e atributos a ser extraída desses modelos, os relacionamentos necessários entre esses itens de informação determinam a necessidade do uso de uma ferramenta computacional. O desenvolvimento simultâneo e integrado do produto e das organizações que implementam os seus processos do ciclo de vida requer o uso de ferramentas computacionais.

O processo de engenharia simultânea de sistemas neste trabalho é aplicado somente a produtos espaciais, isso faz com o que o escopo deste trabalho seja o espacial.

1.1 Motivação

As ferramentas de auxílio computacional existentes e que permitem modelar algumas partes do processo de engenharia de sistema são, em geral, especializadas. Mesmo quando não o são, elas concentram seus pontos positivos em uma determinada parte do processo. Por exemplo, DOORS é uma excelente ferramenta de engenharia de requisitos; CORE é excelente na criação de modelos hierárquicos de produto; STATEMATE é excelente em modelagem de comportamento; há ferramentas especializadas em notações de modelagem. Mas não há ferramenta que cubra o processo de engenharia de sistemas desde *stakeholders* até a arquitetura de implementação, e faça isso

suportando modelagem de produto e de organização, sem a necessidade de adaptações.

A engenharia simultânea de sistemas, por ser composta de pela análise de *stakeholders*, análise de requisitos, análise funcional e análise de implementação, levando em conta os elementos de produto e de organização, simultaneamente, torna necessária uma ferramenta auxiliada por computador para capturar todos os elementos desse processo. Isso seria inviável se não fosse utilizado o auxílio computacional.

Para modelagem dos diversos elementos de produto e organização, desde *stakeholders* até a análise de implementação, seriam necessárias diversas notações e rascunhos em papel. Não seria possível manter rastreabilidade entre esses elementos sem o auxílio computacional ao processo. Rastreabilidade requer que se mantenham manter os *links* de informação entre os elementos de produto e organização e salientar os seus impactos. A engenharia simultânea de sistemas permite derivar atributos de seus modelos que permitem balancear a solução sistema. Isso é, por exemplo, capturado por *links* de impacto.

Uma abordagem, sem o auxílio computacional ao processo de engenharia simultânea de sistemas, não atingiria seu objetivo de antecipar para as etapas iniciais do projeto os requisitos que impactariam no produto e na organização. Ainda não seria possível ter uma maneira de navegar por *links* que descrevem a satisfação do *stakeholder* associado ao produto final. Não seria possível integrar o *stakeholder* ao longo de todo o processo do ciclo de vida do produto, o que é necessário para garantir a satisfação do *stakeholder*, ainda mais se tratando de altíssimos investimentos ao longo de um período de desenvolvimento de um produto espacial, que é susceptível a diversos elementos externos como, por exemplo: política externa, corte de orçamento, falta de pessoal qualificado, o alto e baixo das empresas, e nesse campo surge

uma grande oportunidade de manter o interesse do *stakeholder* perante esses diversos cenários.

Esta dissertação propõe assim adaptar uma ferramenta computacional comercial existente para o auxílio computacional a um processo de engenharia simultânea de sistemas, e será demonstrada para produtos espaciais.

Serão adaptados modelos para descrever a organização, seu contexto funcional, seu comportamento e sua estrutura, seu contexto físico, fluxos e interconexões.

Essa ferramenta permite manter os *links* que descrevem os relacionamentos e os impactos entre produto e organização, e que permite integrar desde *stakeholders* até a solução do projeto baseados na solução do sistema.

Trabalhos segundo Lai (2009), Company et al. (2009) e Leon (2009) tratam de projeto auxiliado por computador ou manufatura auxiliada por computador, ou ainda, inovação auxiliada por computador. Ambos focados no desenvolvimento de produtos. No entanto este trabalho ilustra um auxílio computacional para capturar os elementos de um processo de engenharia simultânea de sistemas, ou seja, todos os elementos desse processo necessários para a concepção do sistema, levando em consideração todo o ciclo de vida do produto e ainda as **organizações** que implementam os processos do ciclo de vida dentro do escopo do esforço de desenvolvimento. Desta maneira não se foca somente no produto, mas também nas organizações.

Atributos e requisitos de produto e de organização derivados do processo de engenharia simultânea de sistemas também são capturados e seus relacionamentos identificados, permitindo posterior análise de impactos.

1.2 Objetivo

O objetivo geral deste trabalho é apresentar uma abordagem para o auxílio computacional a um processo de engenharia simultânea de sistemas. A aplicação do auxílio computacional ao processo se dará usando o exemplo do Satélite Sino-Brasileiro de Recursos Terrestres - CBERS II (INPE, 2011a) e o Veículo Lançador de Microsatélite - VLM, desenvolvido pelo Instituto de Aeronáutica e Espaço - IAE (IAE, 2011).

Os objetivos específicos deste trabalho são:

- propor o auxílio computacional ao processo de engenharia simultânea de sistemas, usando uma ferramenta comercial chamada *Cradle*;
- aplicar o auxílio computacional ao processo de engenharia simultânea de sistemas em exemplos da área espacial;
- analisar os benefícios desse auxílio computacional.

1.3 Metodologia

A natureza dessa dissertação é de pesquisa aplicada (SILVA e MENEZES, 2001), que cobrem os objetivos gerais de forma descritiva (GIL, 1991) que aborda o problema de maneira qualitativa (MARTINS, 2000).

Os seguintes passos para a realização deste trabalho incluíram o seguinte:

- Utilizar um processo de engenharia simultânea de sistemas existente;
- Utilizar uma ferramenta comercial ambiente de engenharia de sistemas disponível;
- Aplicar o processo auxiliado por computador sobre o exemplo de um satélite já desenvolvido, e sobre o exemplo de um veículo lançador em desenvolvimento.

1.4 Estrutura do Trabalho

Este trabalho está organizado em oito capítulos:

- Capítulo 1 (Introdução): este Capítulo onde foram apresentados o escopo, contexto, motivação, objetivos, metodologia e estrutura desta dissertação.
- Capítulo 2 (Fundamentação teórica): são apresentados os conceitos necessários para a realização deste trabalho e outras ferramentas existentes para o auxílio computacional ao processo de engenharia de sistemas. Ele inclui desenvolvimento integrado de produto, as normas que abordam o processo de engenharia de sistemas, engenharia simultânea, modelagem integrada de produto e organização, e as ferramentas disponíveis.
- Capítulo 3 (Engenharia simultânea de sistemas): é apresentado o processo desenvolvimento de produtos complexos a ser auxiliado por computador. Esse processo descreve a análise de *stakeholder*, a análise de requisitos, a análise funcional e a análise de implementação, desempenhada simultaneamente para produto e organização em uma hierarquia de sistema.
- Capítulo 4 (Método): é demonstrada uma maneira de captura dos elementos do processo de desenvolvimento de produto, utilizando os diagramas do pacote de análise gráfica da ferramenta computacional. E mostrando como utilizar a ferramenta computacional para capturar os elementos do processo descrito no Capítulo 3.
- Capítulo 5 (Aplicação do método para o CBERS): é apresentado o auxílio computacional ao processo de engenharia simultânea de sistemas descrito no Capítulo 3, utilizando o método descrito no Capítulo 4, aplicado ao desenvolvimento do satélite CBERS 2.

- Capítulo 6 (Aplicação do método para o VLM): é demonstrado o auxílio computacional ao processo descrito no Capítulo 3 utilizando o método descrito no Capítulo 4, aplicado à engenharia de sistemas do VLM.
- Capítulo 7 (Discussão): apresenta as contribuições desse trabalho se comparado com outros da literatura, e com o estado da prática do desenvolvimento dos exemplos propostos.
- Capítulo 8 (Conclusão): este capítulo conclui o trabalho em relação aos objetivos desta dissertação, de acordo com a motivação descrita no Capítulo 1.

2 FUNDAMENTAÇÃO TEÓRICA

Neste Capítulo é realizada a revisão dos conceitos utilizados para a realização deste trabalho, bem como outras abordagens existentes que clamam a mesma finalidade deste trabalho.

2.1 Desenvolvimento Integrado de Produto

Andreasen e Hein (1987) propuseram o termo desenvolvimento integrado de produto como um modelo idealizado para o desenvolvimento de produto. De acordo com o modelo, a integração toma lugar em termos da criação do mercado, do produto e da produção, e entre o projeto e o gerenciamento incluindo a necessidade para o planejamento contínuo do produto.

De acordo com Andreasen e Hein (1987), integração envolve tratar simultaneamente fatores/aspectos relacionados com mercado, produto e produção; estabelecer objetivos comuns como diretrizes e a estratégia para atingi-los, conduzir projetos de desenvolvimento de produto, executar tarefas práticas; identificar a interação de controle entre projetos de desenvolvimento de produto e interação de controle entre as atividades de desenvolvimento.

Integração envolve no nível mais alto o desenvolvimento organizacional, de mercado, do produto e da produção. No nível mais baixo, integração envolve o controle da qualidade, o controle de finanças, o controle de estoque, vendas, e a análise de competidores.

O desenvolvimento integrado de produto desenvolve simultaneamente o produto, seus processos do ciclo de vida e as organizações que implementam esses processos. São levados em consideração, desde o início, os requisitos de organização dos processos, que juntos com os requisitos específicos do

produto, guiam o processo de desenvolvimento do produto (LOUREIRO e LEANEY, 2003).

Boujut e Laureillard (2002) consideram a questão do *link* entre desenvolvimento do produto e o processo ainda como um problema. Porém os autores consideram que o conceito de integração de produto e processo é uma resposta para esse desafio.

Horváth et al. (2007) consideram que as mudanças na engenharia para o desenvolvimento de produtos na direção do aumento da complexidade, estimularam a implementação de tecnologias avançadas da informação. Por esse motivo, o conceito do modelo de produto integrado é implementado para o gerenciamento do ciclo de vida do produto.

2.2 Processos de Engenharia de Sistemas

Engenharia de sistemas é uma abordagem multidisciplinar colaborativa de engenharia, para derivar uma solução balanceada ao longo do ciclo de vida, e que atende às expectativas dos stakeholders (IEEE, 2005).

O processo de engenharia de sistemas é composto pelas etapas de derivação, evolução e verificação de uma solução de sistema, desde a expressão de uma necessidade até o produto aceito e pronto para uso e operação.

2.2.1 NASA

No *Handbook* da NASA/SP-2007-6105 (NASA, 2007) está definido o conceito de engenharia de sistemas como “uma abordagem disciplinada e metódica para o **projeto, a realização, o gerenciamento técnico, as operações** e o

descarte de um sistema”. Seu processo é **focado no desenvolvimento e na realização de um produto final.**

Na Figura 2.1 são ilustrados interações e fluxos de cada conjunto dos processos do projeto do sistema, realização do produto e gestão técnica.

Nos processos do projeto do sistema inicia-se com uma equipe de estudos coletando e esclarecendo as expectativas dos *stakeholders*, incluindo os objetivos das missões, restrições, diretrizes de projeto, objetivos operacionais e critérios para a definição do sucesso da missão.

Nos processos de realização do produto são produzidos, adquiridos, utilizados ou codificados; integrados em um nível mais alto de montagem; verificados contra as expectativas de *stakeholders*; e feita a transição para o próximo nível de sistema.

Os processos de gestão técnica são as pontes entre o gerenciamento e o time técnico. Esse processo provê a integração das funções que permitem a solução do projeto a ser realizado (NASA, 2007).

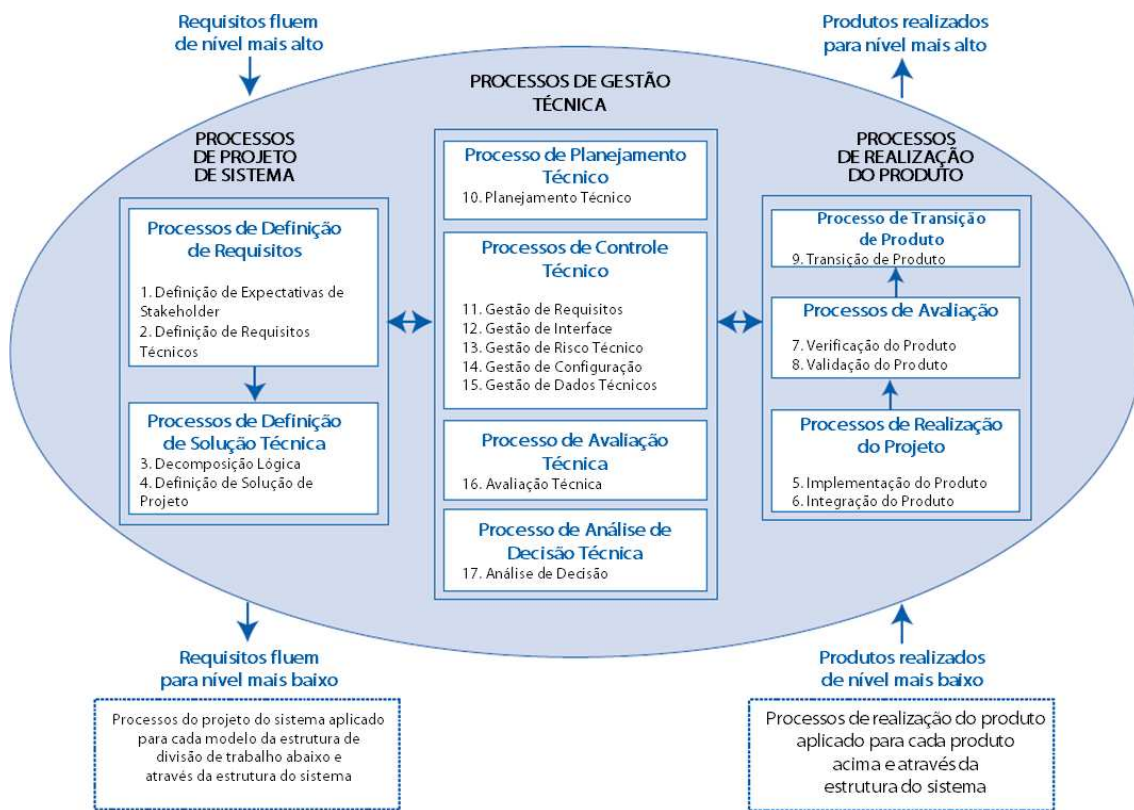


Figura 2.1 - Processo de engenharia de sistemas da NASA.
Fonte: Adaptada de NASA (2007).

2.2.2 ECSS

A norma da Agência Espacial Européia ECSS-E-ST-10C (ECSS, 2009) define engenharia de sistemas como “uma abordagem interdisciplinar governada pelo esforço total técnico para transformar requisitos numa solução de sistema”. A Figura 2.2 ilustra o processo de engenharia de sistemas conforme a norma.

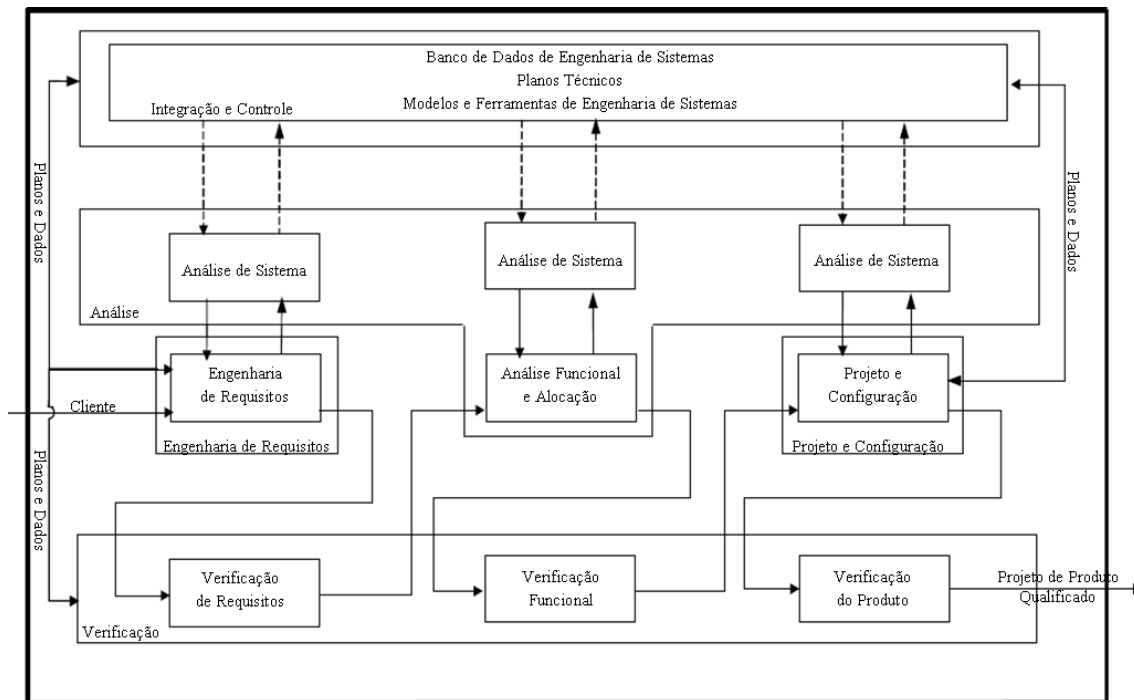


Figura 2.2 - Processo de engenharia de sistemas da ECSS.
 Fonte: Adaptada de ECSS (2009).

2.2.3 IEEE

Na norma IEEE 1220-2005 afirma que “o foco é nas atividades necessárias para guiar o **desenvolvimento do produto**, enquanto garante que o **produto** está projetado apropriadamente para fazê-lo disponível para a produção, operação, manutenção, e eventualmente o descarte, sem o risco à saúde e ao ambiente”.

Seu processo é focado na abordagem de **desenvolvimento do produto** que atenda o balanço dos fatores associados com a viabilidade do ciclo de vida do produto e na competitividade num mercado global.

A Figura 2.3 apresenta os sub-processos que compõem o processo de engenharia de sistemas de acordo com IEEE (2005).

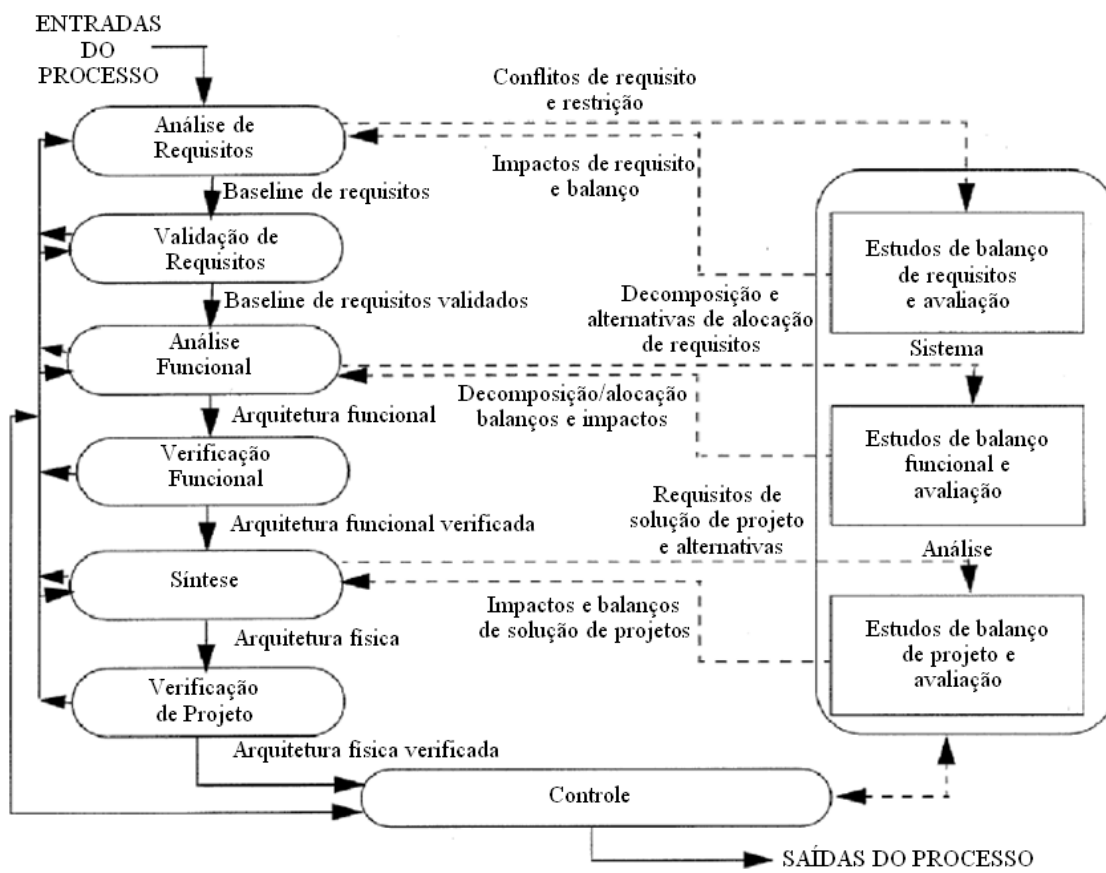


Figura 2.3 - Processo de engenharia de sistemas do IEEE.
 Fonte: Adaptada de IEEE (2005).

2.2.4 EIA

A norma da *Electronic Industries Alliance* EIA-632 (EIA, 1999) define processos para fazer a engenharia de um sistema (*engineering a system*). Os processos para isto são: aquisição e fornecimento, gestão técnica, projeto do sistema, finalização do produto e validação técnica.

Essa norma foca no produto que irá desempenhar os requisitos funcionais do sistema, que são obtidos pelos processos citados acima. Estes processos são aplicáveis para a engenharia ou reengenharia de um **produto final** que compoñha um sistema. São 33 os requisitos especificados pela norma, dos quais a organização, ou um grupo de empresas, ou um projeto, ou na aquisição

ou ainda um fornecedor deve decidir quais processos são aplicáveis; decidir quais requisitos são aplicáveis; estabelecer políticas e procedimentos para a implementação do projeto; definir as tarefas para cada requisito selecionado; e estabelecer os métodos e as ferramentas para a implementação (EIA, 1999).

A Figura 2.4 ilustra os processos para fazer a engenharia de um produto. Os requisitos definidos pela norma são derivados desses processos (EIA, 1999). Os processos definidos na norma EIA (1999) incluem o processo de aquisição e contratação de um desenvolvimento, enquanto a norma IEEE (2005) foca no processo de desenvolvimento de um produto, considerando este já contratado.

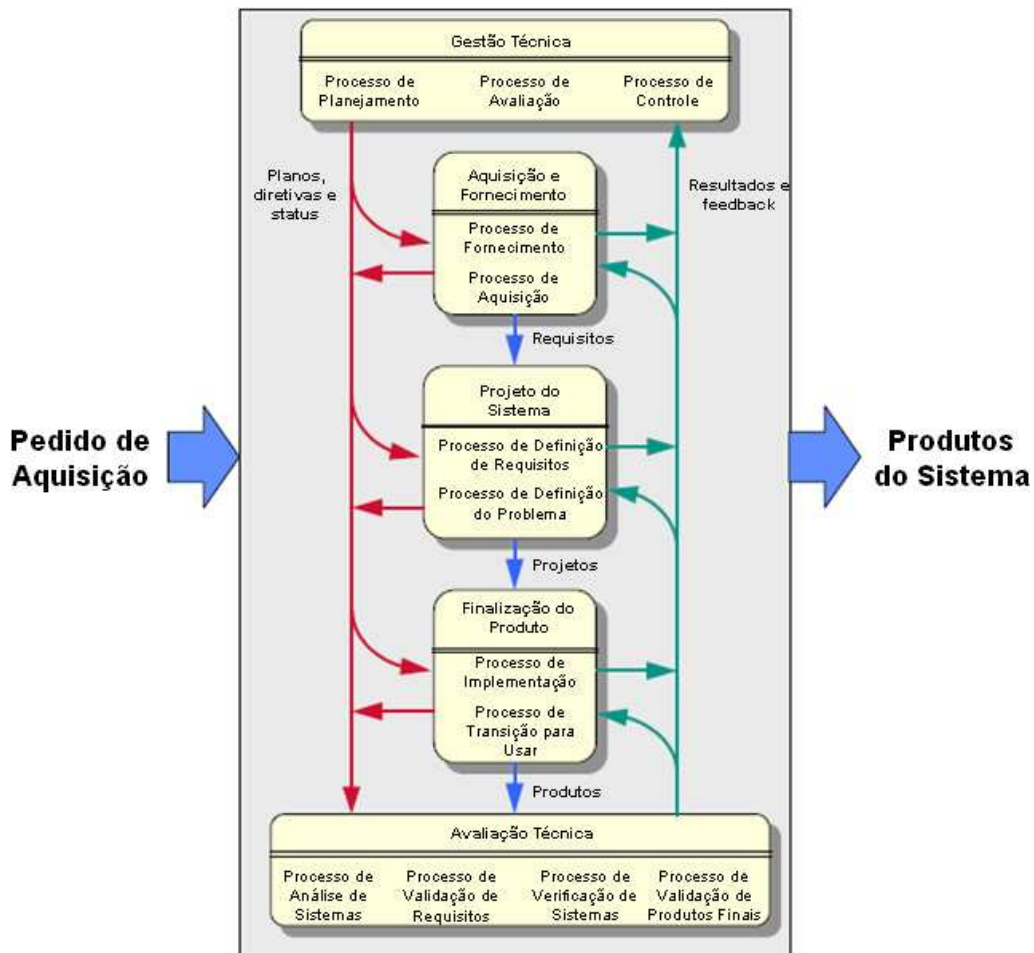


Figura 2.4 - Processos para fazer a engenharia de um sistema.
Fonte: Adaptada de EIA (1999).

2.3 Engenharia Simultânea

A engenharia simultânea é uma abordagem para o projeto simultâneo integrado de produtos e seus processos relacionados, incluindo manufatura e suporte. Essa abordagem pretende fazer com que as pessoas envolvidas no desenvolvimento considerem, desde o início, todos os elementos do ciclo de vida do produto, da concepção ao descarte, incluindo qualidade, custo, prazos e requisitos dos clientes (WINNER et al., 1988).

A engenharia simultânea reconhece o impacto, no projeto do produto, do ciclo de vida do produto, mas trata cada processo de ciclo de vida do produto como entidade isolada. Além disso, engenharia simultânea é tradicionalmente aplicada para o desenvolvimento de partes e não de sistemas (PRASAD, 1996).

A engenharia simultânea é tradicionalmente usada para o projeto de partes. Algumas técnicas, destacadas por Huang (1996):

- *Design for Assembly* ou DFA, onde se executa uma análise para a simplificação da estrutura do produto. O procedimento do DFA resulta em produtos mais simples e menos caros na montagem e manufatura;
- *Design for Manufacturing* ou DFM, essa técnica é ligada ao DFA como um procedimento que foca no processo da manufatura, para um número mínimo de partes. O DFA é conduzido primeiro para uma simplificação da estrutura, e o DFM depois para o detalhe das partes;
- *Design for Inspectability* ou DFI, é desenvolver o projeto do produto para a facilidade de inspeção. Isso é necessário quando o cliente requer alta qualidade, e serviço de segurança contínuo do produto. A facilidade de

inspeção na manufatura provê uma rápida e precisa resposta para o controle do processo;

- *Design for Reliability* ou DFR, é o método para avaliação de quais configurações de projetos tem o potencial para a confiabilidade nas fases iniciais do processo do projeto. DFR permite avaliar onde os fatores de maiores impactos de confiabilidade são identificados no projeto do produto;
- *Design for Quality* ou DFQ, é baseado nos critérios de geração e seleção. As soluções possíveis para uma função é primeiramente gerada, e a melhor solução é escolhida de acordo com o conjunto de critérios de avaliação.

2.4 TEAM X

O *Team X* é um time multidisciplinar e multifuncional de engenheiros da NASA, que usam metodologias de engenharia simultânea para o projeto, análise e validação dos projetos de concepção de missões espaciais.

Esse time fica co-localizado em um espaço físico chamado de centro de projeto, onde esses engenheiros ficam lado a lado em um mesmo ambiente. Esse ambiente possui computadores em rede, em uma infraestrutura de apoio ao gerenciamento de dados, ferramentas de simulação e modelagem, repositórios de histórico de dados e modelos de projeto compartilhado, atualizados em tempo real.

As atividades realizadas pelo time para a execução dos projetos incluem: analisar a convergência do projeto e a integração dos sistemas, levantar as concepções de arquitetura e fazer o balanço delas, analisar o impacto e o

balanço da exploração espacial, visualizar e analisar trajetórias, analisar interfaces, fazer configuração e integração, e ainda analisar o cenário dos recursos técnicos (JPL, 2011).

2.5 CDF

A infraestrutura de projeto simultâneo - *Concurrent Design Facility* da agência espacial europeia é uma facilidade equipada com computadores, dispositivos e ferramentas computacionais que permitem um time de especialistas de diversas áreas aplicarem a metodologia de engenharia simultânea, para o desenvolvimento de missões espaciais futuras.

O foco principal é a avaliação da viabilidade técnica e financeira das missões futuras, e para a concepção dos veículos lançadores.

As sessões para o desenvolvimento de um projeto são conduzidas em reuniões, contando com os representantes de todos os domínios da engenharia espacial, para analisarem de requisitos ao custo do projeto.

As salas ficam em um mesmo ambiente, separadas por grupos de análise. Cada especialista pode acessar a informação e propor uma solução, fazer um levantamento de dados ou alterar o que lhe compete no projeto.

O *layout* do CDF é dividido em posições de análise do tipo:

- líder do time: onde o líder é responsável pelo gerenciamento do estudo. Isso envolve trabalhar junto com o cliente que define o escopo do estudo e do objetivo da missão.
- sistema: durante o estudo da missão, o engenheiro de sistemas conduz a maioria das análises em nível de sistema para a missão, sob a direção do líder do time. O engenheiro de sistema é encarregado do orçamento,

da massa e energia, para verificar se o projeto atende aos requisitos da missão.

- análise da missão: o analista da missão trabalha com o líder e o engenheiro de sistema para definir o ambiente operacional do veículo por toda sua missão. Incluem o perfil de lançamento, a análise orbital e a definição da trajetória.
- sistema de solo e operações: o engenheiro do sistema de solo e das operações analisa os requisitos para o suporte ao veículo pela estação de solo, e calcula o custo da manutenção por todo o ciclo de vida.
- montagem, integração e verificação: o engenheiro é responsável por desenvolver o plano para o desenvolvimento, fabricação, teste e integração do veículo que será lançado de acordo com o cronograma.
- avaliação de risco técnico: o engenheiro responsável analisa o projeto preliminar do veículo para identificar os riscos concernentes ao cumprimento da missão. O engenheiro responsável deve comunicar o líder a respeito dos riscos identificados.
- análise de custo: a responsabilidade nessa posição é estimar o custo do projeto preliminar industrial e da produção, e por prover uma abordagem guiada a custo sobre todo o sistema ao longo do estudo.
- simulação: o engenheiro de simulação é responsável por criar a simulação da missão, de maneira a ser analisado o veículo realizando a sua missão. A simulação deve refletir o impacto do projeto de acordo com os dados e o perfil da missão.
- configuração: o engenheiro de configuração é responsável por fazer o arranjo dos componentes dentro do veículo, e garantir que os componentes tenham requisitos alocados a eles.

Seguindo o mesmo *layout* são ocupadas as posições para o desenvolvimento dos subsistemas, como estrutura, controle de atitude e órbita, propulsão, comunicação, computador de bordo, suprimento de energia, controle térmico, mecanismos e pirotecnia e instrumentos (CDF, 2011).

2.6 Modelagem de Produto

Para a modelagem de produto é necessário dividir o produto em elementos funcionais e elementos físicos. Para a análise do elemento funcional é necessário modelar dados, estados e atividades (CALVEZ, 1993). Para a análise do elemento físico é necessário expressar estrutura e comportamento (CALVEZ, 1993).

Stevens et al. (1998) declara a necessidade de representar um *layout* do produto, que pode ser representado por desenhos e protótipos. Hubka e Eder (1988) mostram que é necessário modelar energia, material e informação, e também as conexões físicas e as conexões lógicas funcionais.

2.7 Modelagem de Organização

Quatro visões devem ser abordadas para a modelagem de organização (VERNADAT, 1996).

- A visão funcional, que provê uma descrição estruturada hierarquicamente das funções, do comportamento (dinâmico), e de estrutura funcional (estática) da organização, com relevantes entradas e saídas.
- A visão da informação, que provê a descrição de um conjunto estruturado de objetos de organização, que foram identificados nas outras visões.
- A visão de recursos, que provê uma descrição da organização dos recursos, ou seja, do conjunto dos recursos requeridos para executar as operações da organização.

- A visão estrutural, que provê a descrição da estrutura da organização, as responsabilidades dos indivíduos, e as unidades da organização dentro da empresa.

Steele et al. (2001) modelam os recursos para a integração de uma organização de manufatura. A arquitetura do modelo de recursos integra a análise da manufatura para o projeto do produto, o planejamento do processo, para o custo da produção, para o controle da qualidade, para a aquisição de recursos, para o planejamento e o cronograma da produção, e para a execução das atividades. A análise adicional desses recursos da manufatura permite a modelagem da organização.

Modelagem de organização é a forma de traduzir o conhecimento da organização que adicione valores à organização, ou às necessidades a serem compartilhadas. Isso consiste em criar modelos de estrutura e comportamento da organização. Para muitas organizações estar no mercado significa, atender aos requisitos dos clientes e ser inovador; reduzir o *time-to-market* dos produtos; e produzir com qualidade e baixo custo. Todas essas necessidades precisam ser modeladas. O modelo de organização é usado como um mecanismo da unificação semântica, ou o mapeamento do conhecimento (VERNADAT, 2002).

2.8 Modelagem Integrada de Produto e Organização

Segundo Haque et al. (2003) barreiras à engenharia simultânea ou desenvolvimento integrado do produto são de contexto organizacional. Para facilitar a engenharia simultânea mudanças para o projeto e para o desenvolvimento da organização necessitam ser considerados. O primeiro passo para melhorar um projeto é a análise de sua organização. Para a realização da engenharia simultânea é necessário um alto grau de integração

entre as diferentes funções dentro da organização. Essa integração depende do nível no qual uma organização adapta sua estrutura organizacional e os seus processos.

Horváth et al. (2007) afirmam que não é suficiente a descrição dos objetos de engenharia que incluem atributos e conexões associativas, modelados de forma convencional, para tomada de decisão.

No entanto, Horváth et al. (2007) propõem uma modelagem orientada a conteúdo, que difere do modelo de orientação a dados. Modelo orientado a dados descreve os objetos de engenharia, incluem atributos, conexões associativas, e especifica os procedimentos de modelagem. Dados do objeto e suas conexões associativas não contêm informação suficientes para o propósito das explicações na tomada de decisões. Para o modelo orientado a conteúdo leva-se em conta os métodos para os atributos, o histórico das tomadas de decisões, e as intenções humanas.

A orientação a dados deve complementar a orientação a conteúdo, com mais ou menos informações que ajudem na tomada de decisões. Orientação a conteúdo melhora a comunicação entre humanos e procedimentos de modelagem, e estende a descrição de objetos de engenharia além de simples dados (HORVÁTH et al., 2007).

2.9 BPMN

A notação de modelagem de processo de negócio (*business process modeling notation* - BPMN) tem por principal objetivo prover uma notação que seja compreensível por todos os usuários de negócios, a partir dos analistas de negócios que criam os rascunhos dos processos, até os desenvolvedores técnicos responsáveis pela implementação da tecnologia que irá desempenhar

os processos, e finalmente, as pessoas envolvidas nos negócios que irão gerenciar e monitorar esses processos (OMG, 2011a).

2.10 SysML

O SysML - *Systems Modeling Language*, é uma linguagem de modelagem gráfica de propósito geral, para a análise, especificação, projeto, verificação e validação de sistemas complexos, sendo que nesses sistemas podem estar incluídos hardware, software, dados, pessoas, procedimentos, facilidades e outros elementos de sistemas naturais (FRIEDENTHAL et al, 2009).

O SysML pode representar estes sistemas, componentes ou outras entidades por (FRIEDENTHAL et al, 2009):

- composição estrutural, interconexão e classificação;
- comportamento baseado em função, baseado em mensagem e baseado em estados;
- restrições nas propriedades físicas e de desempenho;
- alocação entre comportamento, estrutura e restrições (ex: funções alocadas aos componentes);
- requisitos e seus relacionamentos para outros requisitos, elementos de projetos e casos de teste.

Apesar disso, o SysML é feito como uma extensão do UML - *Unified Modelling Language* (OMG, 2011b) para possibilitar a modelagem de sistemas compostos por hardware e software. O foco do SysML ainda é no produto.

2.11 Ferramentas disponíveis

O INCOSE - *International Council on Systems Engineering* (INCOSE, 2010a) oferece uma base de dados de ferramentas para, gestão de requisitos, para

arquitetura de sistemas e de medidas de sistemas. Isso auxilia os interessados na busca e escolha em ferramentas de engenharia de sistemas. (INCOSE, 2011).

As informações a respeito das ferramentas são obtidas por um formulário padrão, disponibilizado pelo grupo de trabalho da base de dados das ferramentas (INCOSE, 2010b). Os fornecedores das ferramentas se comprometem com a veracidade da informação ao preencherem esse questionário. O questionário aborda questões de forma que posteriormente seja possível comparar ferramentas que possuam funcionalidades para gestão de requisitos e para a arquitetura de sistemas.

Esse grupo de trabalho busca disponibilizar as informações mais úteis das ferramentas, de forma que o usuário interessado possa comparar na base de dados as ferramentas lado a lado (INCOSE, 2010b). Ver Figura 2.5.

Ferramentas	ARTISAN Studio	CORE 5.1	inteGREAT Version 4.7	SLATE 6.5 (a UGS Teamcenter product)	UGS Teamcenter for Systems Engineering 2005	OPCAT	Cradle 6.3
Data de Resposta	20 de Junho, 2005	07 de Julho, 2006	02 de Agosto, 2010	09 de Junho, 2005	11 de Junho, 2005	19 de Fev, 2006	29 de Agosto, 2010
1.6 Relaciona descrição de entradas, saídas e processo para cada elemento de arquitetura?	Full	Full	Full	Part	Part	Full	Full
1.12 Relaciona conceito operacional para elementos de arquitetura?	Part	Full	Full	Full	Full	Full	Full
1.13 Permite estimar custos por um recurso de planilha?	Part	Full	Full	Full	Full	Full	Full
2.2 Permite uma visão de conectividade de interface?	Full	Full	None	Full	Full	Full	Full
2.3 Apoio a várias arquiteturas físicas (em diversos níveis)?	Full	Full	None	Full	Full	Full	Full
4.4 Provê uma biblioteca de ícones padrões para desenvolvimento de arquitetura?	Full	Part	Full	Full	Full	Full	Full
4.5 Permite controle de arrastar e soltar ícones dentro da arquitetura?	None	Part	Full	Full	Full	Full	Full
4.6 Seu produto permite o cliente personalizar ícones com definições para arquiteturas específicas?	Full	Full	Full	Full	Full	Part	Full
5.3 Interface para software de rastreabilidade de requisitos? Quais produtos?	Full	Full	Full	Part	Full	Full	Full
5.5 Interface CAD e projeto de software e ferramentas de codificação? Quais produtos?	Full	Full	Full	Part	Full	Full	Full
6.6 Possui diagrama de contexto, FFBD, hierarquia de diagramas, conectividade e layouts físicos?	Full	Full	Full	Part	Full	Full	Full
9.4 Permite usuário criar macros e comandos para automatizar tarefas tediosas?	Full	Full	None	Part	Part	None	Full
10 Apoio e manutenção?	Full	Full	Full	Part	Part	Full	Full

Figura 2.5 - Tabela comparativa das ferramentas disponibilizadas no site do INCOSE.

Em comparação a Seção 2.3 e 2.11, a Figura 2.6 ilustra a cobertura das ferramentas de engenharia simultânea e de engenharia de sistemas com o *Cradle* sobre o processo de engenharia simultânea de sistemas. A Seção 2.3 apresenta ferramentas da engenharia simultânea tais como: DFA, DFM, DFI, DFR, DFQ que são utilizadas em partes dos processos do ciclo de vida de um produto.

Um software que implementa o DFA e o DFM é o software chamado DFMA de Boothroyd Dewhurst, Inc. (BOOTHROYD DEWHURST, 2011). Dessa maneira o DFA e o DFM são auxiliados por computador para projeto de partes e do processo de manufatura (BOOTHROYD DEWHURST, 2011).

A Figura 2.6 ilustra o processo de engenharia simultânea de sistemas diante de algumas ferramentas. As ferramentas como DOORS, por exemplo, são aplicadas para a engenharia de requisitos. A ferramenta de engenharia simultânea DFA, por exemplo, é aplicada para a montagem de um dado produto. A Figura 2.6 serve para ilustrar e comparar que uma ferramenta ambiente de engenharia de sistemas como o *Cradle* cobre o processo de engenharia simultânea de sistemas, desde o nível da missão até o de componentes do sistema, enquanto a ferramenta DFMA cobre somente o processo de manufatura e montagem de partes. Por exemplo, na Figura 2.6 ela está representada na seta horizontal no nível de implementação, como no caso de trabalhar com projeto para a manufatura, onde se tem elementos físicos para serem alterados.

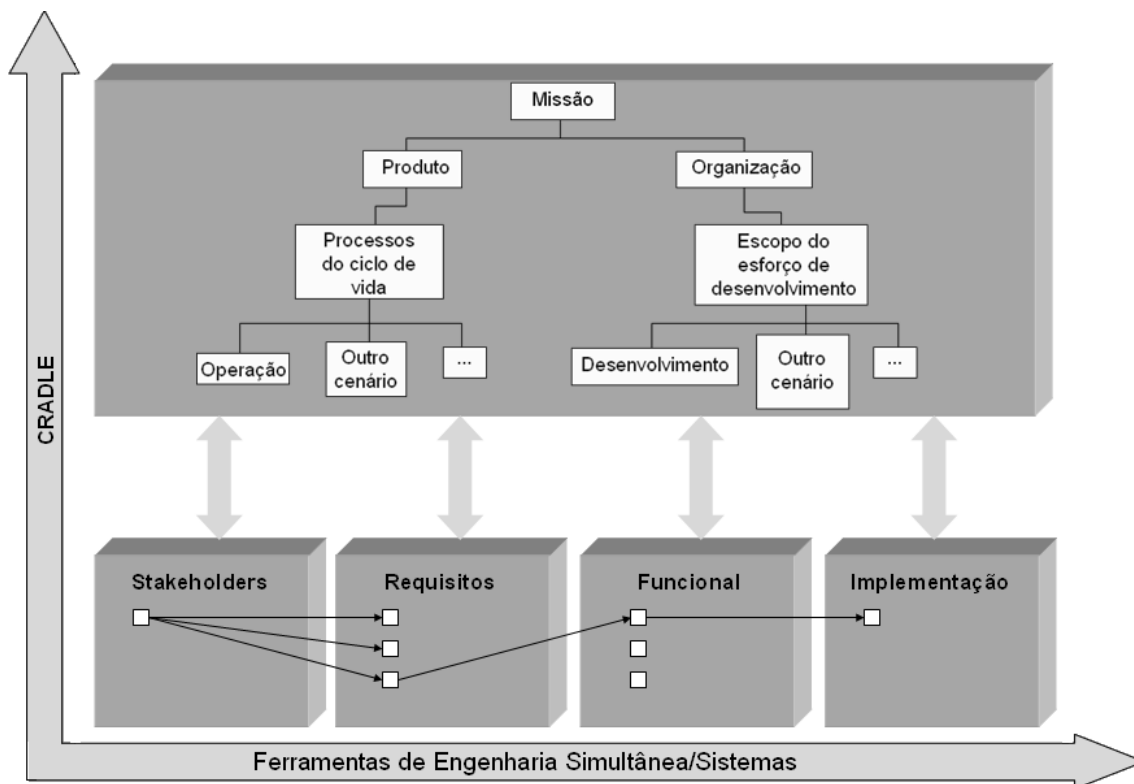


Figura 2.6 - Processo de engenharia simultânea de sistemas face às ferramentas de engenharia simultânea e engenharia de sistemas.

2.12 Framework e Método para Desenvolvimento de Produto Complexo

Vernadat (1996) define *framework* de modelagem como uma coleção de princípios de modelagem, métodos, ou ferramentas relevantes para um domínio de aplicação

Loureiro (1999) demonstra um *framework* e um método para o desenvolvimento integrado de produtos complexos. O *framework* é chamado de *Total View Framework* e o método de *Concurrent Structured Analysis Method*.

O *framework* e o método integram os conceitos de engenharia de sistemas e engenharia simultânea. Eles provêm um conjunto total de elementos de produto, seus processos de ciclo de vida, as organizações que implementam

esses processos e as interações entre esses elementos desde o início do processo de desenvolvimento. O *framework* estende a aplicação do processo de engenharia de sistemas para as organizações que implementam os processos de ciclo de vida do produto e aplica engenharia simultânea em todos os níveis da árvore do produto.

As saídas do método são requisitos, atributos funcionais, atributos físicos e as interações entre eles (LOUREIRO, 1999).

2.12.1 Framework de Visão Total

A abordagem do *framework* é para evoluir de requisitos para atributos, no processo de desenvolvimento integrado de produto. O *framework* ajuda gerenciar a complexidade associada ao produto complexo, as interações entre os atributos de produto, dos processos do ciclo de vida do produto e das organizações que implementam esses processos. O *framework* integra por modelagem, o produto, seus processos de ciclo de vida e as organizações que implementam esses processos, pela análise de requisitos, a análise funcional e pela análise física, em todos os níveis da árvore do produto (LOUREIRO, 1999).

Na Figura 2.7 são ilustradas as três dimensões do *framework*, a dimensão da análise, a dimensão de integração e a dimensão de estrutura.

Na dimensão da análise definem-se os diferentes tipos de análises, que são realizados simultaneamente, para identificar os requisitos e os atributos do produto, dos processos e das organizações. Os tipos de análise são: análise de requisitos, análise funcional e análise física.

Na dimensão de integração definem-se os elementos a serem integrados, dentro do sistema a ser desenvolvido. Esses elementos são, produto, processo e organização.

Na dimensão da estrutura define-se como o sistema complexo é estruturado hierarquicamente, podendo ser decomposto em níveis de detalhe. Cada seção horizontal da pirâmide representa um nível de detalhamento do sistema.

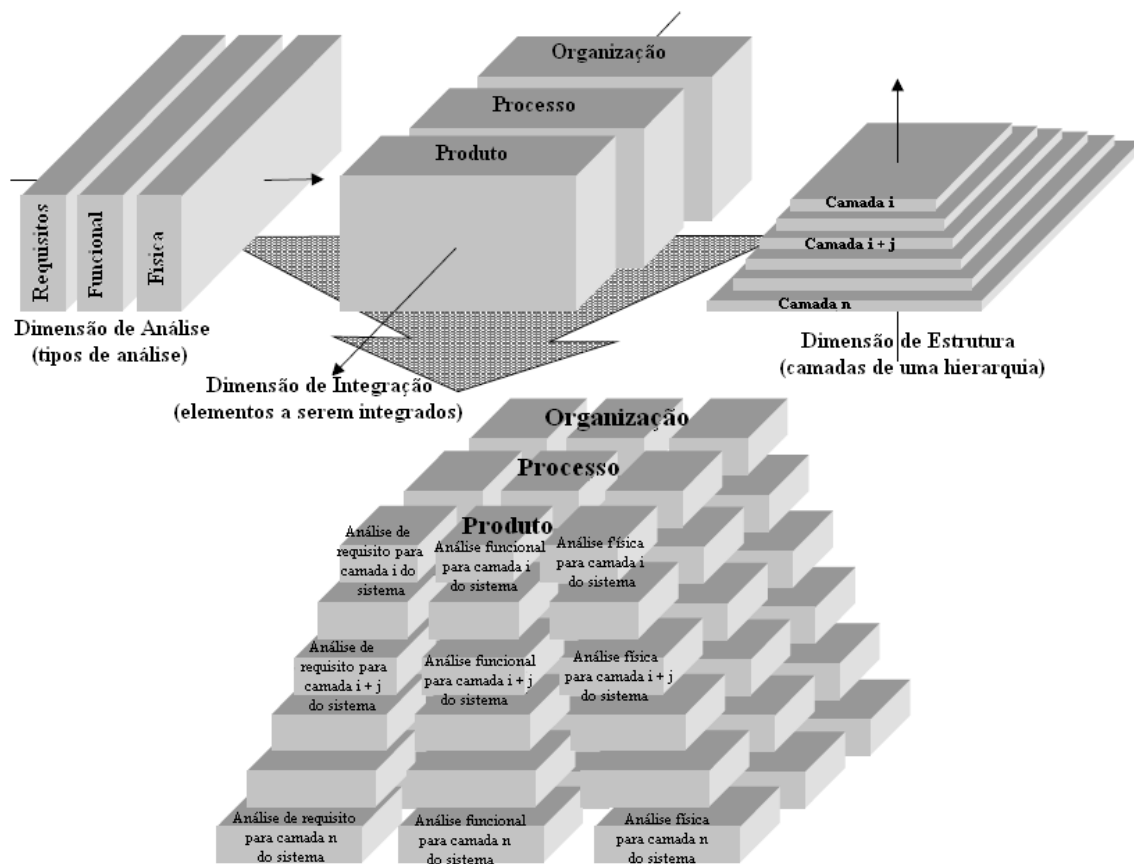


Figura 2.7 - Framework de visão total.
Fonte: Adaptada de Loureiro (1999).

2.12.2 Método de Análise Estruturada Simultânea

O método proposto por Loureiro (1999) é um método de análise estruturada dentro do *Total View Framework*, que deriva os requisitos e atributos de produto, processo e organização. O método consiste basicamente da análise simultânea do produto, seus processos do ciclo de vida e das organizações que implementam esses processos. O processo de análise é composto da

análise de requisitos, da análise funcional e da análise física. Estes processos de análise usam e adaptam métodos de análise estruturada (YOURDON, 1990). Estes métodos de análise estruturada são baseados em técnicas de modelagem. Como resultado dos processos de análise, requisitos e atributos são identificados e seus relacionamentos capturados (LOUREIRO, 1999).

O processo proposto na Figura 2.8 pretende prover uma definição estruturada e iterativa do problema e do desenvolvimento da solução. A natureza iterativa do processo de análise é caracterizada pelos *loops* de requisitos, *loops* de projeto e os *loops* de verificação. O *loop* de requisitos representa, por exemplo, que se novas funções são identificadas, novos requisitos necessitarão ser definidos. O *loop* de projeto garante que as decisões de projeto sejam adequadas ao conceito projetado. O *loop* de verificação garante que o domínio da solução mapeie corretamente ao domínio do problema. A matriz de representação de relacionamento representa o relacionamento entre requisitos e atributos.

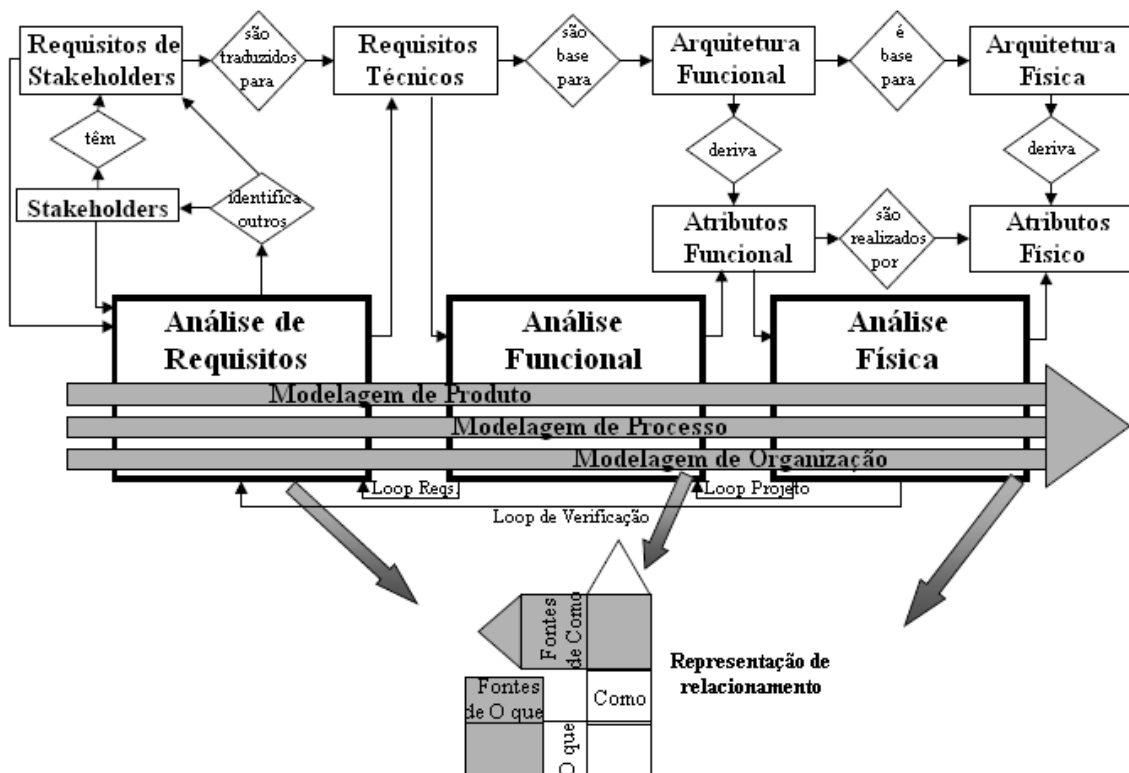


Figura 2.8 - Método de análise simultânea estruturada.
Fonte: Adaptada de Loureiro (1999).

O *framework* e o método descritos nas Subseções 2.12.1 e 2.12.2 respectivamente, evoluíram para o *framework* e o método descritos a seguir nas Seções 3.1 e 3.2 respectivamente. Esta abordagem será apresentada no Capítulo 3.

3 ENGENHARIA SIMULTÂNEA DE SISTEMAS

A abordagem aqui descrita é a evolução da tese desenvolvida por (LOUREIRO, 1999) ilustrada na Seção 2.10. As lições aprendidas desta evolução podem ser encontradas em (LOUREIRO, 2010b). A aplicação desta abordagem em diversos exemplos é ilustrada nos seguintes artigos: (FULINDI et al., 2010; LOUREIRO et al., 2010a; 2010b; 2010c; 2010d; 2010e).

3.1 Framework

O *framework* apresentado na Subseção 2.12.1 possui três tipos de análises, a análise de requisitos, a análise funcional e a análise física. Na sua dimensão de integração ele possui três itens a serem integrados, o produto, o processo e a organização. O *framework* ilustrado na Figura 3.1 ganha mais um tipo de análise, a análise de *stakeholders*. Essa análise surge da evolução do *framework* e do método com a iniciativa de integrar o próprio *stakeholder* durante todo o processo, sendo assim a solução sistema baseada em *stakeholders*. Portanto passa a ter quatro tipos de análises, a análise de *stakeholders*, a análise de requisitos, a análise funcional e a análise de implementação. A análise de implementação é a análise física. A palavra física exclui soluções que não contém elementos materiais ou físicos tais como soluções de software.

Na dimensão de integração, o *framework* da Subseção 2.12.1 possui três elementos a serem integrados: o produto, o processo e a organização. O *framework* ilustrado na Figura 3.1 considera que o elemento de processo já está incluído na análise funcional da organização. Isso porque os elementos de processos são os elementos funcionais da organização. Os processos são funções da organização. Portanto os elementos a serem integrados passam a ser, o produto e a organização (LOUREIRO, 2009; 2010a).

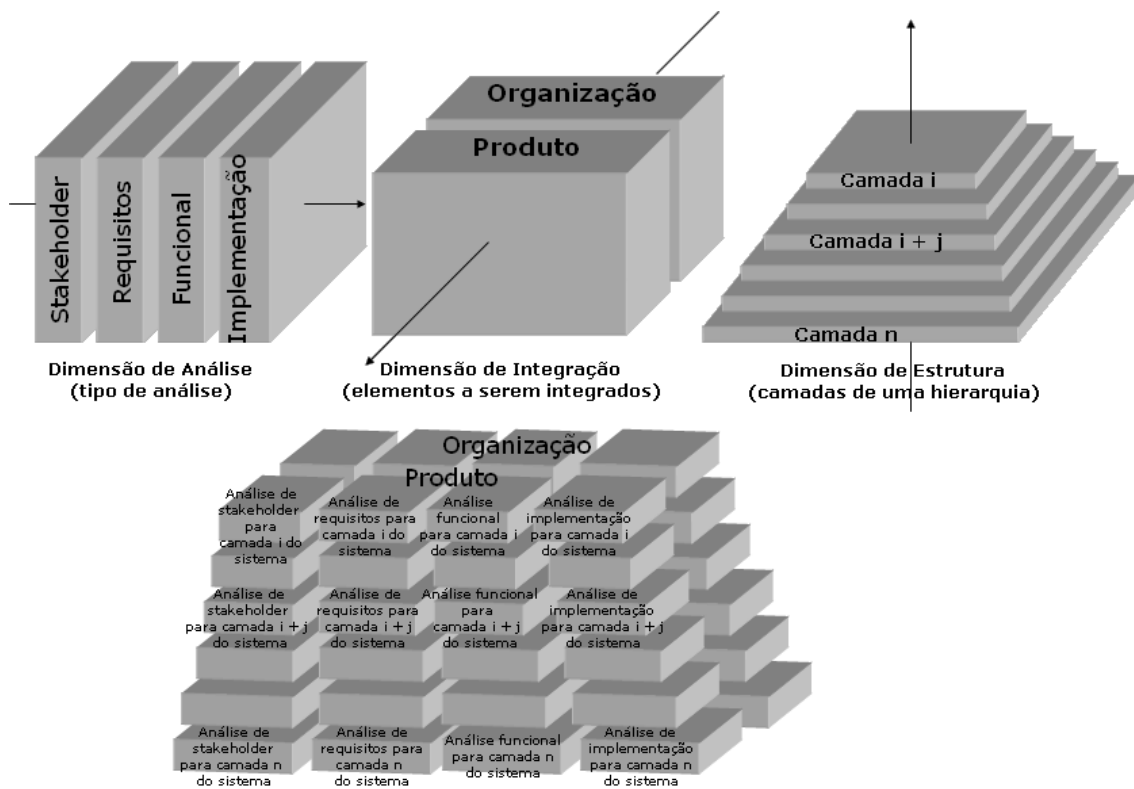


Figura 3.1 - Framework de visão total a partir da evolução de Loureiro (1999).
 Fonte: Adaptada de Loureiro (2010b).

3.2 Método

O método ilustrado na Figura 3.2 que está inserido no *framework* (Figura 3.1) passa a desempenhar a análise de *stakeholders*. O método realiza simultaneamente para produto e organização, a análise de *stakeholders*, análise de requisitos, análise funcional e análise de implementação.

O *loop* entre a análise de requisitos auxilia na realimentação de requisitos que atendam as necessidades dos *stakeholders*, e que auxilie na interpretação do que o *stakeholder* quer.

Requisitos técnicos são identificados, iterativamente, no *loop* entre análise de requisitos e análise funcional. Requisitos são inicialmente alocados às funções do modelo da análise funcional.

Soluções de projeto são derivadas durante o *loop* de projeto, onde funções, com seus requisitos associados, são alocadas aos elementos de implementação que as desempenharão.

O loop de verificação visa confirmar que o sistema a ser implementado atende os requisitos identificados na análise de requisitos.

A validação do sistema ocorre com a satisfação do *stakeholder*, ou seja, com a confirmação de que o produto atende os requisitos de *stakeholders*.

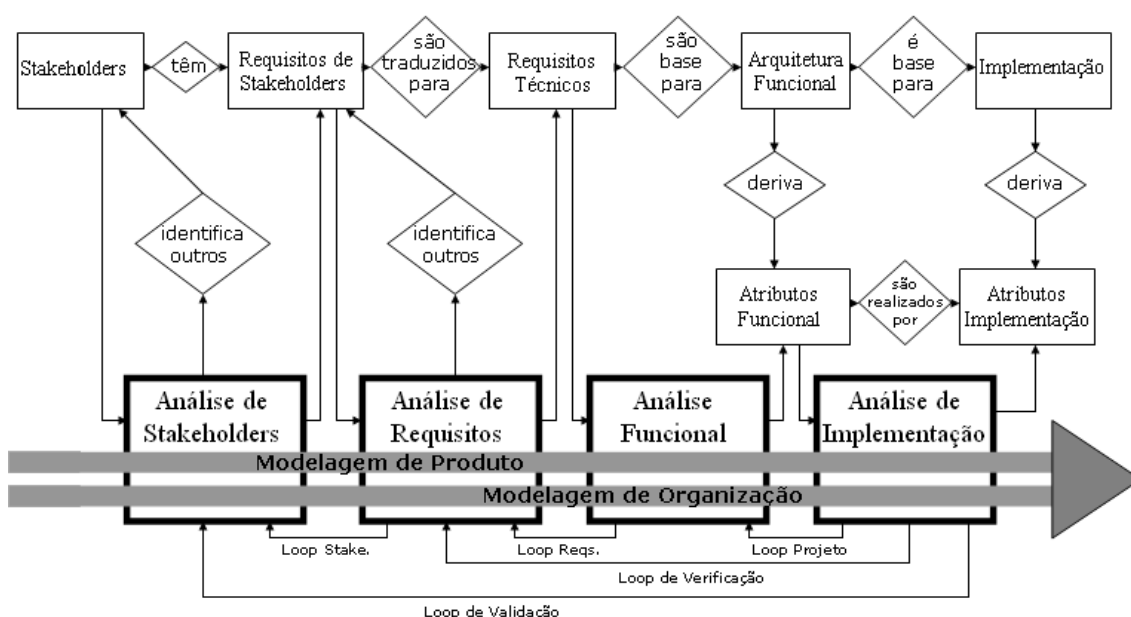


Figura 3.2 - Método dentro do *framework* de visão total.
Fonte: Adaptada de Loureiro (2010b).

A Figura 3.3 ilustra os elementos da abordagem de engenharia simultânea de sistemas. Nela foram acrescentados alguns elementos que não fazem parte da figura original (LOUREIRO, 2010b), são eles:

- atributos funcionais de produto;
- atributos funcionais de organização;
- atributos de implementação de produto;
- atributos de implementação de organização;
- relacionamento entre atributos.

A identificação dos atributos de produto e de organização e de seus relacionamentos é necessária para a identificação de impacto entre os atributos de produto e de organização, e a antecipação dos requisitos dos processos do ciclo de vida para o início do processo de desenvolvimento do produto.

Na Seção 3.3 são descritos os elementos ilustrados na Figura 3.3.

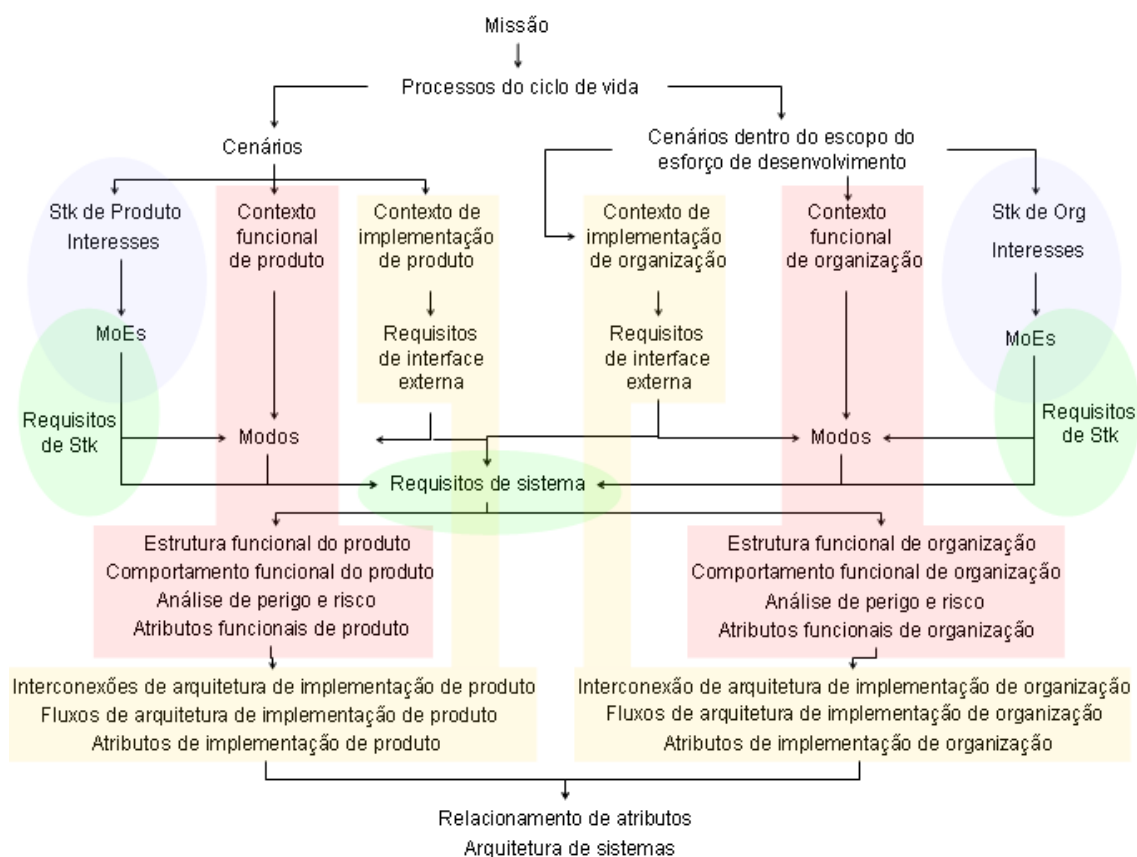


Figura 3.3 - Elementos do processo de engenharia simultânea de sistemas.
 Fonte: Adaptada de Loureiro (2010b).

3.3 Descrição do Processo de Engenharia Simultânea de Sistemas

O método ilustrado na Figura 3.2 dentro do *framework* ilustrado na Figura 3.1, resulta numa visão geral do processo de engenharia simultânea de sistemas, ilustrado na Figura 3.3. Os elementos ilustrados no processo (Figura 3.3) são descritos nas Subseções de 3.3.1 a 3.3.29.

As figuras das subseções mostram uma maneira ilustrativa, de capturar os elementos para produto e organização, simultaneamente para esse processo.

As figuras para exemplificar esse processo pertencem ao exemplo do Anexo C, a engenharia de sistemas de um forno de microondas.

O processo inicia-se com a declaração da missão, a definição dos processos do ciclo de vida do produto e seus respectivos cenários, e dos cenários dentro do escopo do esforço de desenvolvimento. Após a definição desses elementos realiza-se o processo para o desenvolvimento do sistema, e por final a arquitetura de sistema resultante desse processo.

3.3.1 Missão

A missão é a declaração do objetivo a ser realizado pelo sistema. Isso inclui o propósito do sistema ou a razão de ser do sistema.

A missão pode ser capturada pela declaração:

“O sistema deve aquecer alimentos através da frequência de ressonância das moléculas de água.”

Dessa forma é descrito o propósito do sistema, ou a razão de ser.

3.3.2 Processos do Ciclo de Vida

Os processos do ciclo de vida descrevem as etapas da vida do sistema, e pode incluir outras séries de estágios, como transporte e entrega.

Os processos do ciclo de vida podem ser capturados por um diagrama de comportamento, mostrando a sequência de processos do ciclo de vida do sistema. A Figura 3.4 ilustra a notação para a elaboração de um diagrama de comportamento. Onde aparece ‘Função x’ na Figura 3.4 são representados os processos do ciclo de vida do produto a ser desenvolvido. A Figura 3.5 ilustra uma representação seqüencial dos processos do ciclo de vida.

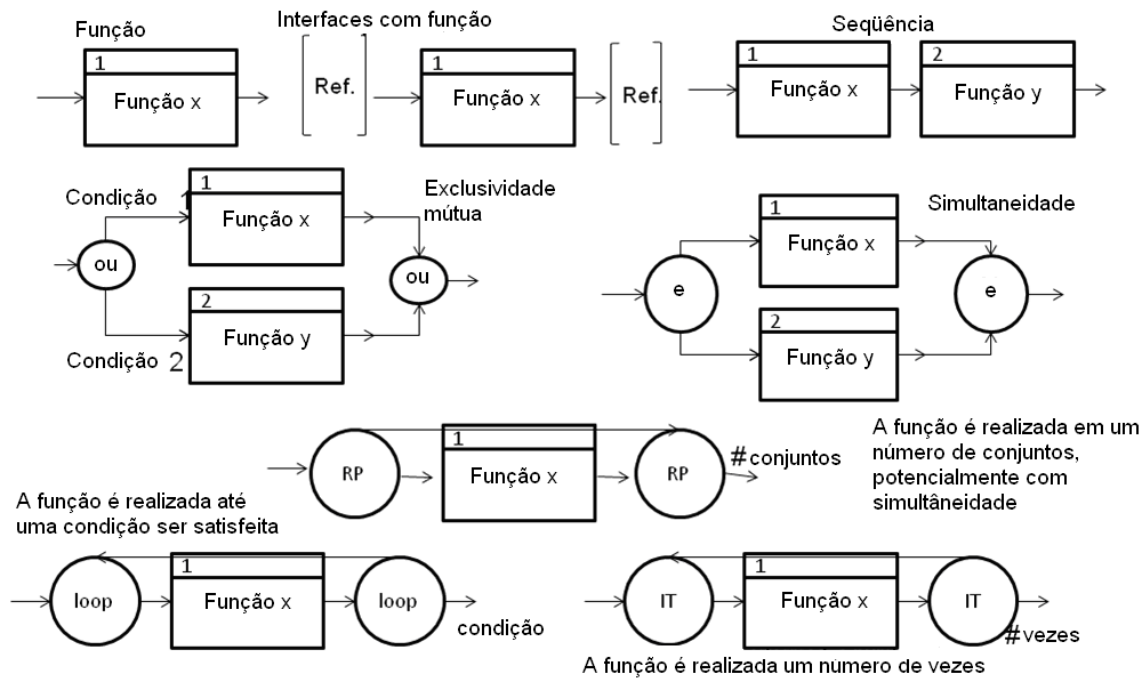


Figura 3.4 - Notação do diagrama de comportamento para os processos do ciclo de vida do sistema.
 Fonte: Adaptada de: 3SL (2009).

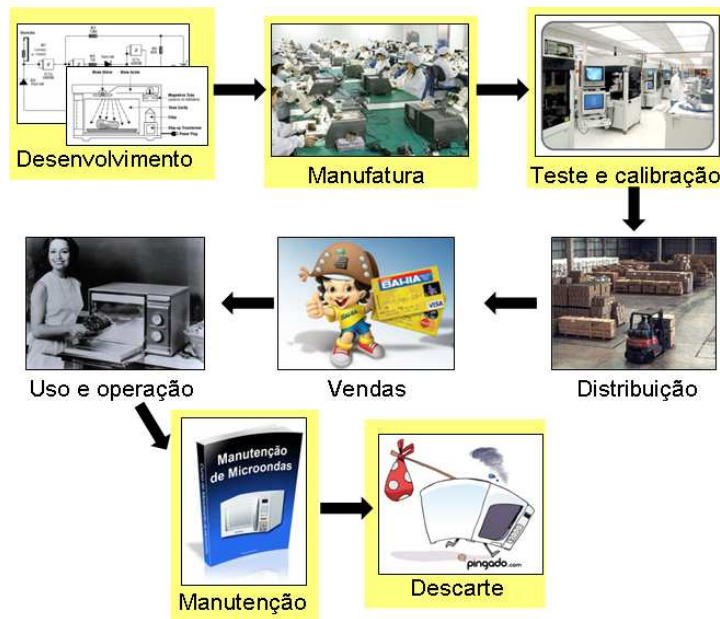


Figura 3.5 - Exemplo estilizado de uma seqüência de processos.

3.3.3 Cenários

Os cenários dos processos do ciclo de vida são as alternativas em cada processo, ou a decomposição de um processo.

Os cenários podem ser obtidos a partir do desdobramento do diagrama de comportamento, quando representar os processos do ciclo de vida do sistema. Neste caso, as alternativas ou a decomposição dos processos do ciclo de vida são destacadas de forma escrita, ao lado da imagem que representa os cenários dos processos do ciclo de vida.

A Figura 3.6 ilustra algumas alternativas dos processos do ciclo de vida para cada cenário.



Figura 3.6 - Desdobramento dos processos do ciclo de vida em cenários, para o caso de decomposição.

3.3.4 Cenários Dentro do Escopo do Esforço de Desenvolvimento

A organização que desenvolve o produto também realiza outros processos, ou cenários. Os processos ou cenários realizados pela organização que desenvolve o produto fazem parte o escopo do esforço de desenvolvimento. Por exemplo, a EMBRAER desenvolve e presta serviços de manutenção para

as aeronaves que desenvolve. Sendo assim o escopo do esforço de desenvolvimento inclui os processos de ‘desenvolvimento’ e de ‘manutenção’. Deve-se ilustrar quais processos ou cenários, serão parte do escopo do esforço de desenvolvimento, como ilustrado na Figura 3.7.



Figura 3.7 - Cenários que serão realizados pela organização.

A aplicação do processo deve ser realizada, para todos os cenários dos processos do ciclo de vida para o produto. E também para todos os cenários e processos dentro do escopo do esforço de desenvolvimento da organização. Para efeito de ilustração do método, o método será aplicado para dois cenários do processo do ciclo de vida do produto, e para dois cenários dentro do escopo do esforço de desenvolvimento da organização.

3.3.5 Stakeholder de Produto

Stakeholders de produto são as pessoas que afetam ou são afetadas pelo produto durante o ciclo de vida do produto.

Para cada cenário identificado, os *stakeholders* são capturados como ilustra a Figura 3.8. A Figura 3.8 ilustra a captura de *stakeholders* de produto para os cenários, ou processos do ciclo de vida para dois desses cenários ou processos. No entanto, os *stakeholders* devem ser identificados para todos os processos do ciclo de vida.

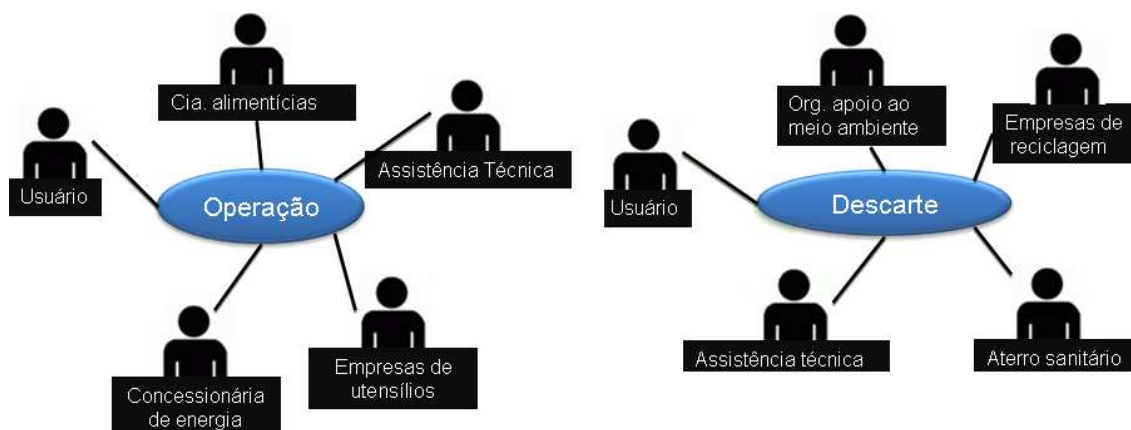


Figura 3.8 - *Stakeholders* de produto para o produto em operação, e para o produto em outro processo do ciclo de vida ou cenário.

Para a identificação dos *stakeholders* de produto, as seguintes perguntas são feitas:

- 1) Quem são as pessoas ou organizações de pessoas, que provêem entrada para o produto em um dado cenário ou processo?
- 2) Quem são as pessoas ou organizações de pessoas, que recebem saídas do produto em um dado cenário ou processo?
- 3) Quem são as pessoas ou organizações de pessoas, que provêem controles para o produto em um dado cenário ou processo?
- 4) Quem são as pessoas ou organizações de pessoas, que provêem mecanismos para o produto em um dado cenário ou processo?

3.3.6 Stakeholder de Organização

Stakeholders de organização são as pessoas que afetam, ou são afetadas, ou tem um interesse pelos negócios da organização.

A Figura 3.9 ilustra a captura de *stakeholders* de organização, para os cenários dentro do escopo do esforço de desenvolvimento.

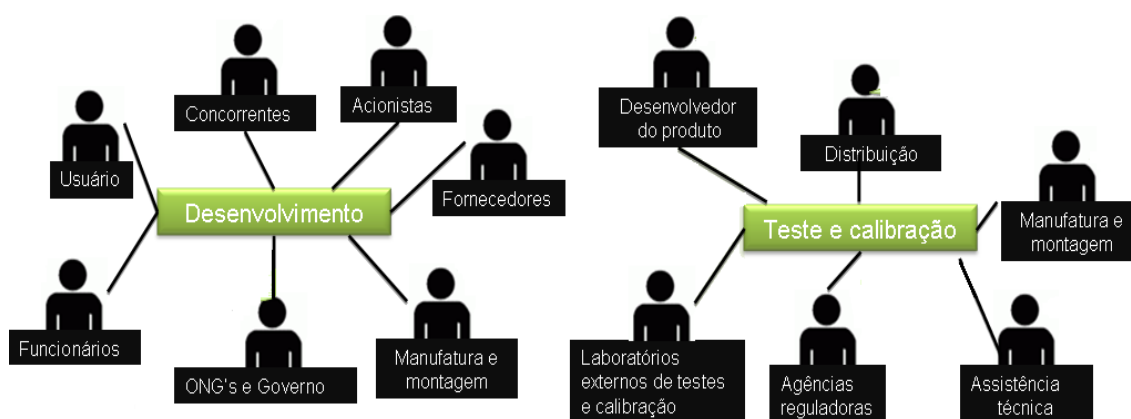


Figura 3.9 - *Stakeholders* de organização.

Para a identificação dos *stakeholders* de organização, as seguintes perguntas são feitas:

- 1) Quem são as pessoas ou organizações de pessoas, que provêm entrada para a organização que realiza um dado cenário ou processo?
- 2) Quem são as pessoas ou organizações de pessoas, que recebem saídas da organização que realiza um dado cenário ou processo?
- 3) Quem são as pessoas ou organizações de pessoas, que provêm controles para a organização que realiza um dado cenário ou processo?
- 4) Quem são as pessoas ou organizações de pessoa, que provêm mecanismos para o produto em um dado cenário ou processo?

3.3.7 Interesses de Stakeholder de Produto

Os interesses são as expressões das preocupações dos *stakeholders* de produto, em cada cenário dos processos do ciclo de vida do produto, ou seja, os aspectos pelos quais os *stakeholders* se sentem afetados ou afetando o produto. Interesses de *stakeholders* de produto são, por exemplo, para o motorista de um carro: conforto, dirigibilidade.

A Figura 3.10 ilustra a captura desses interesses para os cenários do produto em operação, e em um outro processo do ciclo de vida. A Figura 3.10 ilustra um interesse por *stakeholder*, mas um determinado *stakeholder* pode ter mais de um interesse.

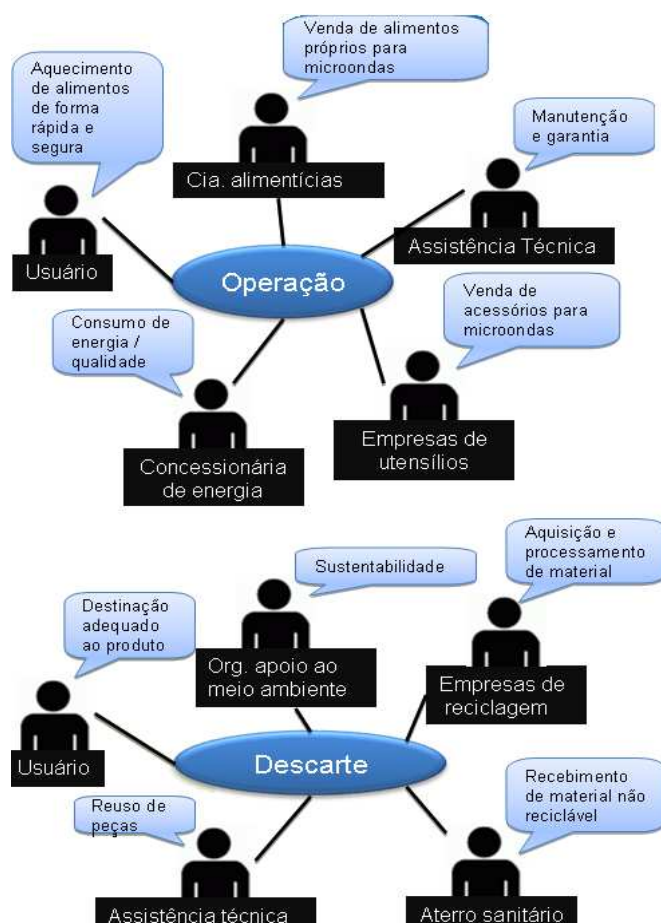


Figura 3.10 - Interesses de *stakeholders* de produto.

3.3.8 Interesses de *Stakeholder* de Organização

O interesse é a expressão da preocupação dos *stakeholders* de organização, realizando cada cenário dentro do escopo do esforço de desenvolvimento, ou seja, os aspectos pelos quais os *stakeholders* se sentem afetados pela organização ou afetando a organização, que realiza um dado cenário ou processo. Interesses de *stakeholders* de organização são, por exemplo, para uma agência ambiental que regulamenta a manufatura de um determinado produto: o atendimento à regulamentação ambiental.

A Figura 3.11 ilustra a captura desses interesses para as organizações, que realizam um cenário de desenvolvimento ou um cenário de um outro processo do ciclo de vida. A Figura 3.11 ilustra um interesse por *stakeholder*, mas um determinado *stakeholder* pode ter mais de um interesse.

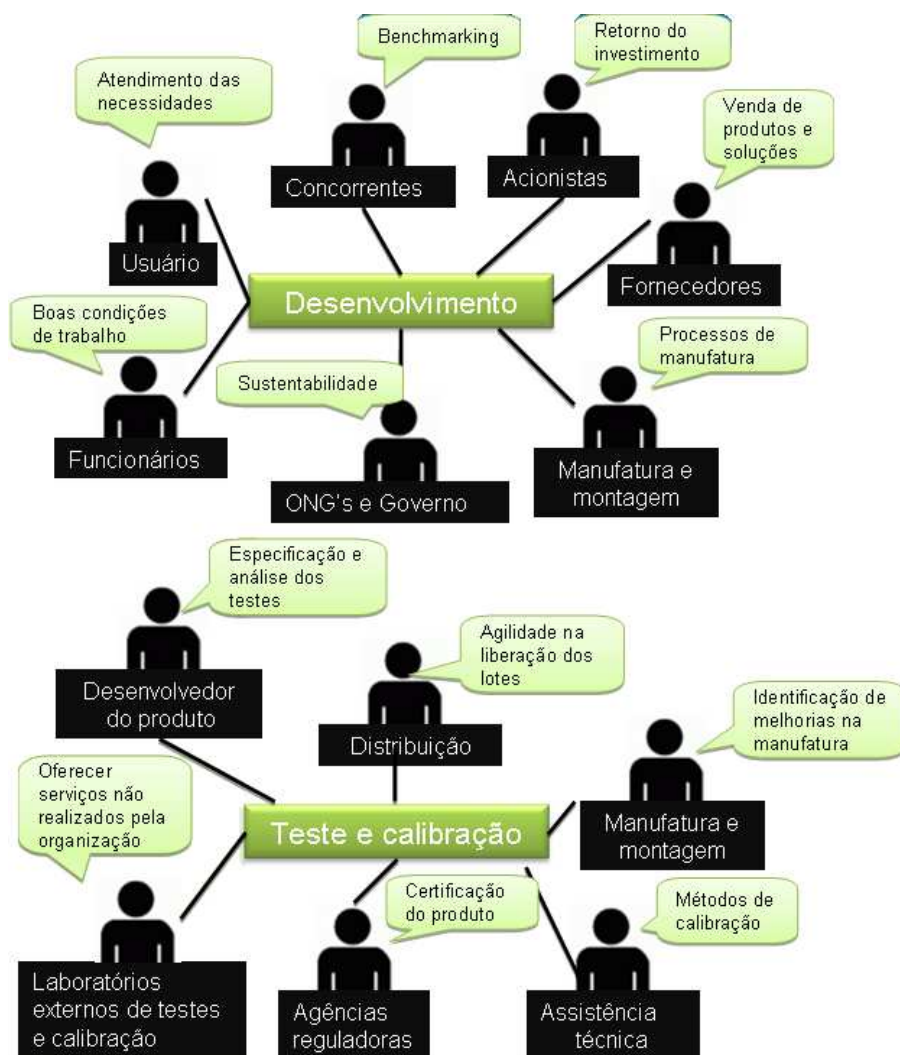


Figura 3.11 - Interesses de *stakeholders* de organização.

3.3.9 MoEs e Requisitos de *Stakeholder* de Produto e de Organização

A partir dos interesses de *stakeholders* de produto e de *stakeholders* organização, medidas de efetividade (MoEs) e requisitos de *stakeholders* são identificados.

Os interesses são desdobrados usando o método GOAL, QUESTION, METRIC (SOLINGEN e BERGHOUT, 1999), ou uma simples árvore de objetivos. Cada interesse está no topo da árvore. Pergunta-se como medir se aquele interesse foi atendido. As primeiras respostas derivarão métricas (ou seja, uma palavra tal como custo, prazo, economia de combustível, produtividade, que engloba um conjunto de medidas). As métricas são sucessivamente desdobradas até que as medidas (objetivas e quantificáveis) sejam obtidas. Essas métricas e medidas identificadas são as Medidas de Efetividade.

Para produto elas representam uma forma de avaliar a satisfação dos *stakeholders* com o produto a ser desenvolvido. Para a organização elas representam uma forma de avaliar a satisfação dos *stakeholders* com a organização, que realizará os processos dentro do escopo do esforço de desenvolvimento. A atribuição de valor e tolerância a essas medidas geram requisitos de *stakeholders*.

As Figuras 3.12 e 3.13 ilustram o desdobramento das medidas de efetividade para produto e organização respectivamente.



Figura 3.12 - Medidas de efetividade para produto.

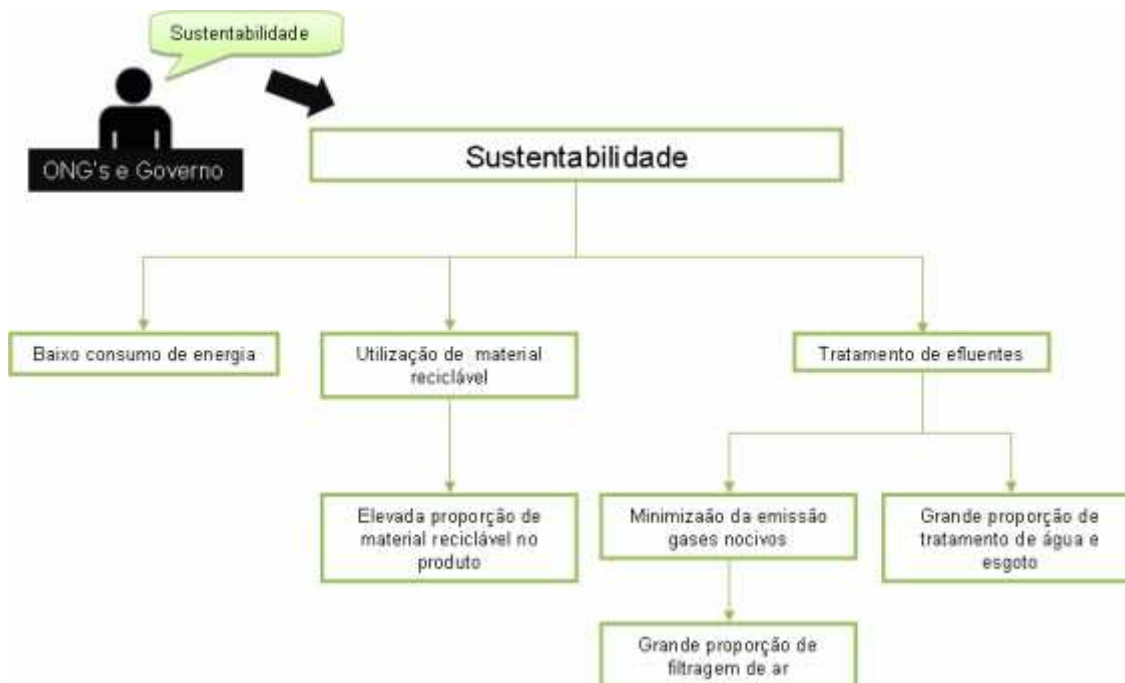


Figura 3.13 - Medidas de efetividade para organização.

As Figuras 3.14 e 3.15 ilustram um exemplo de requisito de *stakeholder* de produto e um exemplo de requisito de *stakeholder* de organização. Esse exemplo ilustra o método da lista de evento, onde a resposta é o requisito de sistema.

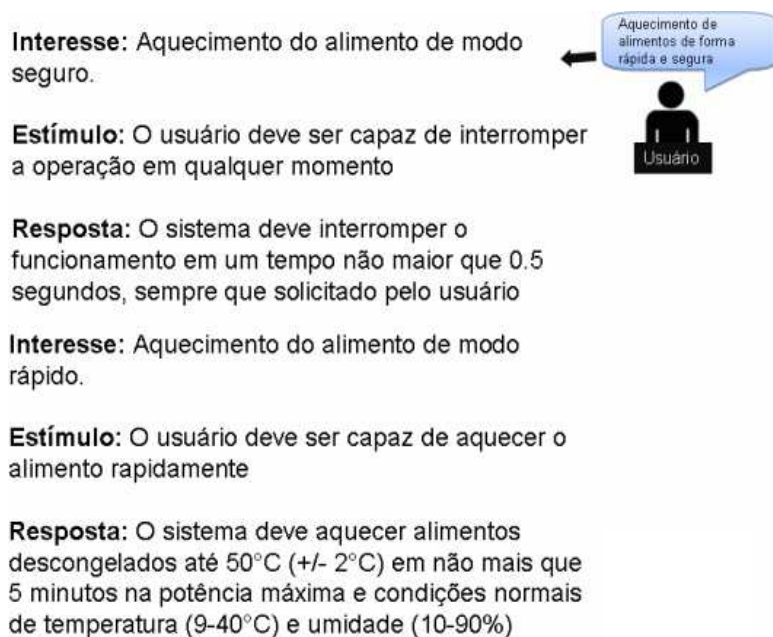


Figura 3.14 - Lista de eventos para produto.

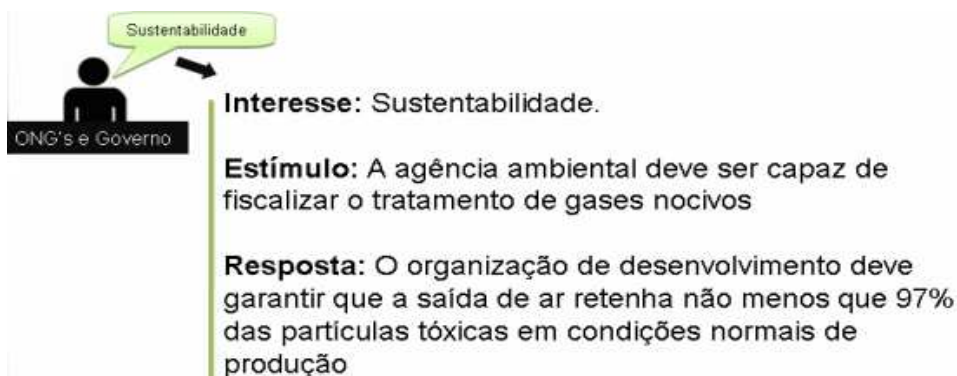


Figura 3.15 - Lista de eventos para organização.

3.3.10 Requisitos de Sistema

Os requisitos de sistema são derivados a partir da análise dos requisitos de *stakeholders*, que por sua vez são derivados dos interesses e MoEs. A análise de requisitos de *stakeholders* inclui a negociação com os *stakeholders* daqueles requisitos, que farão e que não farão parte do sistema, da alocação de requisitos para produto e para organização, da identificação de que requisitos de *stakeholders* derivarão capacidades (o que o sistema deve fazer) e aqueles que derivarão restrições (e.g. desempenho, limitações físicas ao que o sistema deve fazer).

A Figura 3.16 ilustra requisitos de sistemas para produto e organização.

Requisitos do sistema	Função/ Desempenho/ Condição	Produto/ Organização	Mandatário/ Desejável	Status	Verificação	
					Método	Procedimento
1. Aquecimento de alimentos de forma rápida e segura						
1.1. Sistema de deve interromper o processo...	Função	Produto	Mandatário	TBD	demonstração	TBD
1.1.1 ...em um tempo não maior que 0.5 segundos...	Desempenho					
1.1.2 ...sempre que solicitado pelo usuário.	Condição					
1.2. Sistema deve aquecer o alimento descongelado...	Função	Produto	Mandatário	TBD	teste	TBD
1.2.1 ...até 50°C (+/- 2°C) em não mais que 5 minutos...	Desempenho					
1.2.2 ...na potência máxima e condições normais de temperatura (9-40°C) e umidade (10-90%).	Condição					
2. Sustentabilidade						
2.1. A Organização deve tratar efluentes gasosos...	Função	Organização	Desejável	TBD	teste	TBD
2.2.1 ...garantindo que a saída de ar retenha não menos que 97% das partículas tóxicas...	Desempenho					
2.2.2 ...em condições normais de produção.	Condição					

Figura 3.16 - Requisitos de sistema para produto e para organização.

3.3.11 Contexto Funcional de Produto e Modos

Contexto funcional define o que é sistema, o contorno do sistema, e o que está no ambiente do sistema. O ambiente do sistema contém os elementos que estão fora do sistema, mas que trocam material, energia e informação com o sistema. O contorno do sistema é definido pelos fluxos de material, energia e informação que o atravessam. O contexto funcional de um produto em um dado cenário deve ser identificado em cada cenário do processo do ciclo de vida de produto. Os elementos do ambiente podem ter diferentes estados, e o conjunto desses estados são chamados de circunstâncias. O sistema deve ter diferentes modos, dependendo das circunstâncias.

A Figura 3.17 ilustra os elementos do ambiente que trocam energia, material e informação com o sistema.

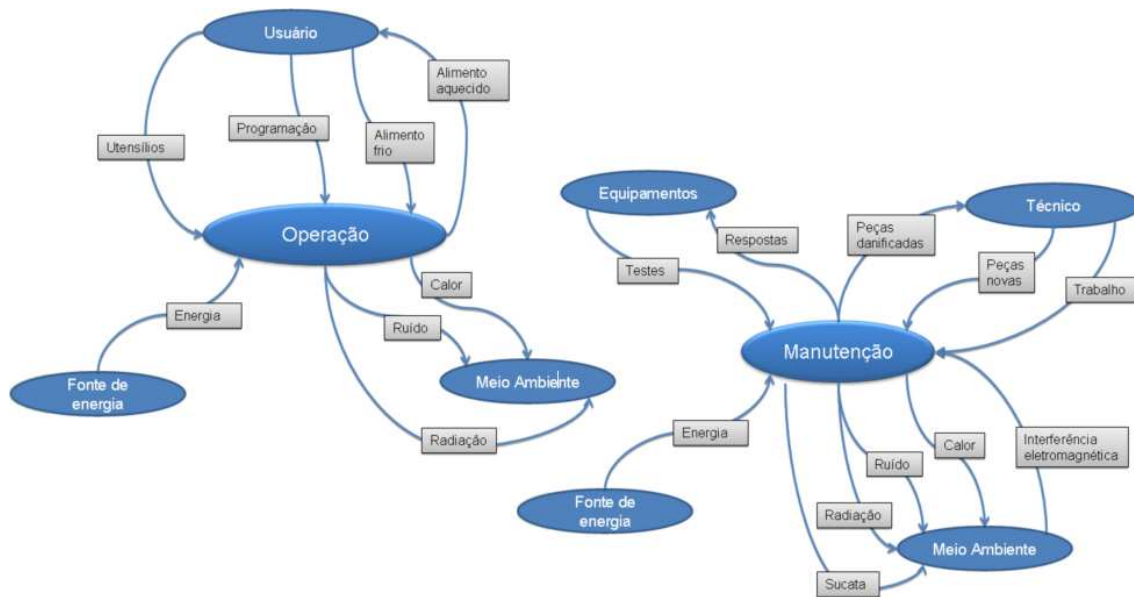


Figura 3.17 - Contexto funcional do sistema para produto.

A Figura 3.18 ilustra como capturar as circunstâncias ou os estados dos elementos do ambiente.



Figura 3.18 - Circunstâncias nos cenários de produto.

Os modos são um conjunto de funcionalidades que o produto realiza dependendo das circunstâncias. Uma tabela deve ser construída, relacionando em uma coluna as circunstâncias e em outra os modos. Exemplos de modos são: voo em modo manual, voo em piloto automático.

3.3.12 Contexto Funcional de Organização e Modos

O contexto funcional de organização é identificado em cada cenário do processo do ciclo de vida, dentro do escopo do esforço de desenvolvimento. O contexto funcional de organização também identifica os elementos do ambiente, que trocam energia, material e informação. E os estados dos elementos do ambiente são identificados, no contexto funcional de organização.

A Figura 3.19 ilustra o contexto funcional da organização, e os elementos do ambiente.

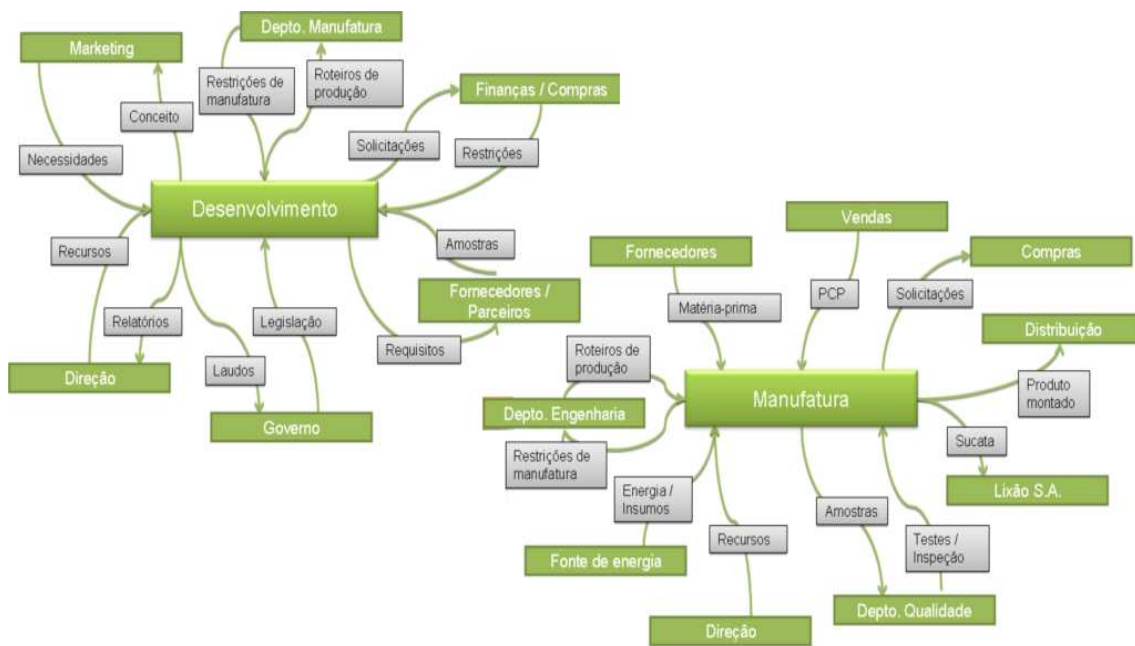


Figura 3.19 - Contexto funcional de organização.

A Figura 3.20 ilustra a captura dos estados dos elementos do ambiente ou circunstâncias.

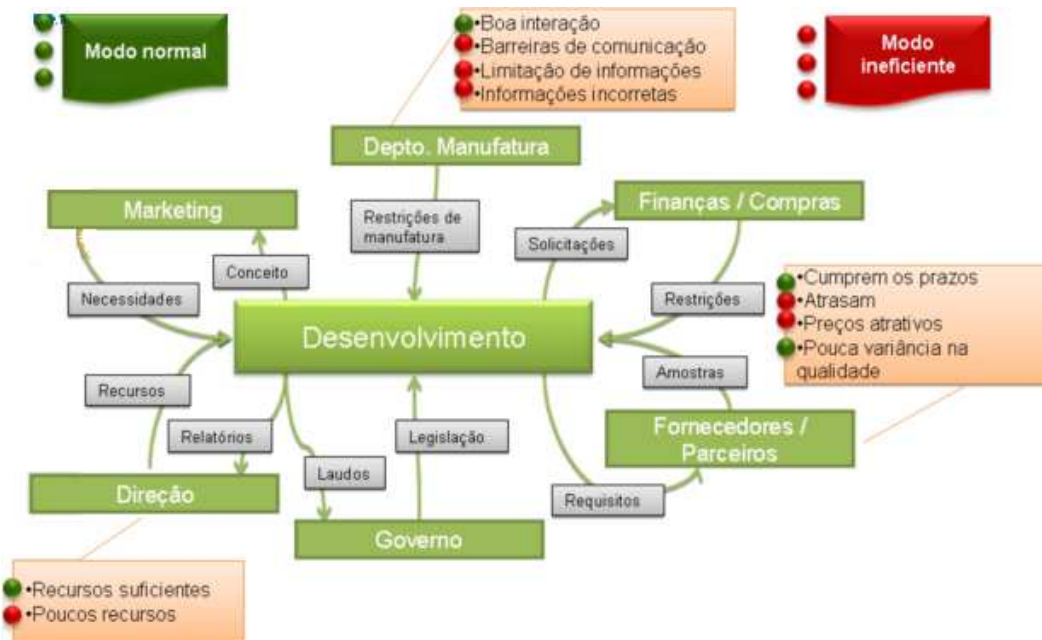


Figura 3.20 - Circunstâncias nos cenários de organização.

Os modos são um conjunto de funcionalidades que a organização realiza dependendo das circunstâncias. Uma tabela deve ser construída, relacionando em uma coluna as circunstâncias e em outra os modos. Exemplos de modos são: em operação normal, em concordata.

3.3.13 Estrutura Funcional do Produto

A partir do contexto do produto, para a identificação das funções internas do produto em um determinado cenário, elabora-se uma lista de eventos (YOURDON, 1990) levando em conta os elementos do ambiente e os fluxos correspondentes. Os eventos são frases que informam o que o produto faz com os fluxos. As respostas são frases que informam o que o produto faz com os fluxos dado um evento. Essas respostas serão então as funções iniciais do produto. A Figura 3.21 ilustra o método da lista de eventos.

A partir dessas funções iniciais, funções necessárias para interligar essas funções iniciais são derivadas. As funções e os fluxos de material, energia e informação entre elas compõem a estrutura funcional do produto. A Figura 3.22 ilustra um diagrama contendo as funções da estrutura funcional do produto. Esses são exemplos de modelos de estrutura porque contêm elementos e os relacionamentos entre os elementos. Neste caso, os elementos são as funções e os relacionamentos são as trocas de material, energia e informação entre elas.

Produto em Operação

1. Usuário acessa interior do forno	1. O sistema provê acesso ao interior do forno
2. Usuário fornece informação de modo de preparo	2. O sistema processa informação de modo de preparo
3. Usuário aguarda pedido de confirmação	3. O sistema processa informação de modo de preparo
4. Usuário envia comando de início de preparo	4. O sistema recebe comando de início de preparo
5. Usuário aguarda preparo	5. Sistema executa preparo
6. Usuário recebe informação de término de preparo	6. Sistema provê informação de término de preparo
7. Usuário fecha compartimento	7. Sistema identifica porta fechada

Organização de Desenvolvimento

1. O MKT envia os necessidades dos clientes para O.D.	1. A Organização de Desenvolvimento analisa as necessidades dos clientes
2. A manufatura envia feedback de outros produtos	2. A O.D. define os requisitos do produto
2. Os fornecedores/parceiros enviam informações à O.D.	2. A O.D. define os requisitos do produto
2. O órgão certificador impõe requisitos de certificação	2. A O.D. define os requisitos do produto
2. O MKT envia informações para a O.D.	2. A O.D. define os requisitos do produto
3. Os fornecedores/parceiros enviam informações à O.D.	3. A O.D. define os requisitos dos sub-sistemas
3. A manufatura envia informações à O.D.	3. A O.D. define os requisitos dos sub-sistemas
4. Os fornecedores/parceiros enviam informações à O.D.	4. A O.D. define a arquitetura
4. A manufatura envia informações à O.D.	4. A O.D. define a arquitetura
5. Os fornecedores/parceiros enviam informações à O.D.	5. A O.D. realiza a prototipagem e a integração do sistema
5. A manufatura envia informações à O.D.	5. A O.D. realiza a prototipagem e a integração do sistema
6. O órgão certificador impõe requisitos de certificação	6. A O.D. certifica o produto
7. Os fornecedores/parceiros enviam informações à O.D.	7. A O.D. define a manufatura
7. A manufatura envia informações à O.D.	7. A O.D. define a manufatura

Figura 3.21 - Lista de eventos para identificar funções de produto e organização.

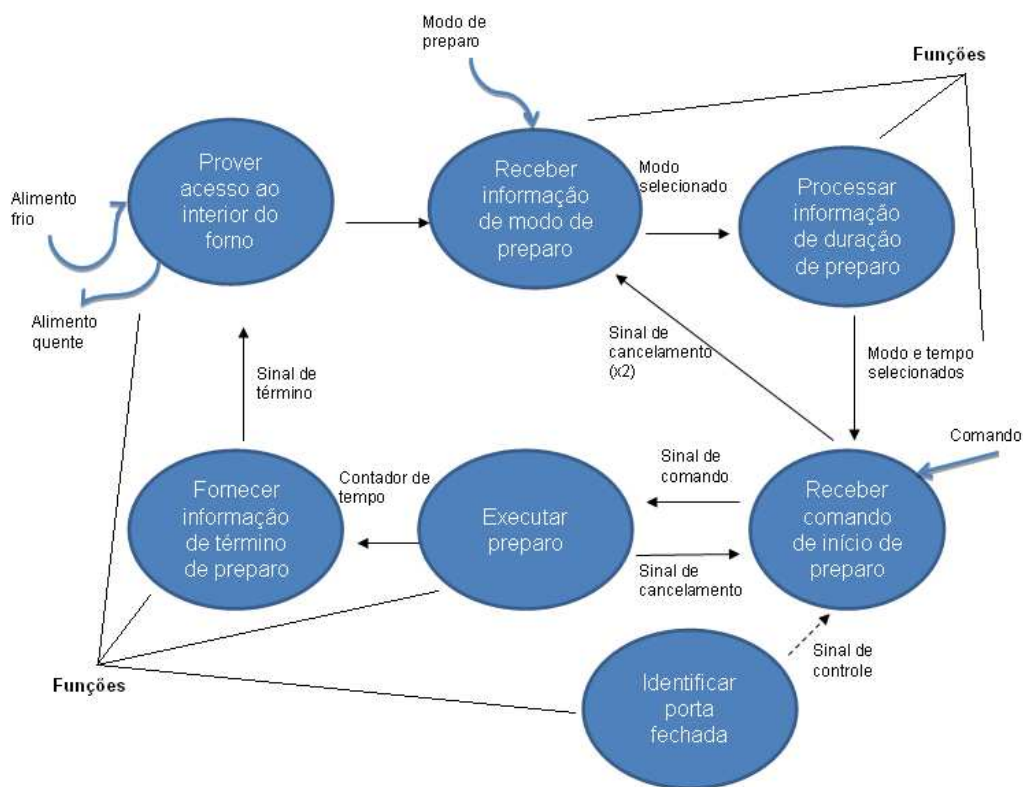


Figura 3.22 - Diagrama de estrutura funcional de produto.

3.3.14 Comportamento Funcional do Produto

O comportamento funcional do produto indica quando e sob que condições uma determinada função é iniciada, encerrada, gatilhada, interrompida, a sequência no tempo da realização das funções, seus aspectos causais e lógicos. É o comportamento funcional que define as transições entre estados e entre modos do sistema. A Figura 3.23 ilustra um *template* de um diagrama de estado, onde os retângulos representam os estados, e as flechas, as transições.

Cada transição é descrita por uma barra horizontal. Na parte superior tem-se as condições para a mudança de estado e na parte inferior as ações decorrentes da mudança de estado ou as funções que são realizadas a partir

da mudança de estados. Modelos comportamentais capturam os aspectos dinâmicos dos elementos do sistema.

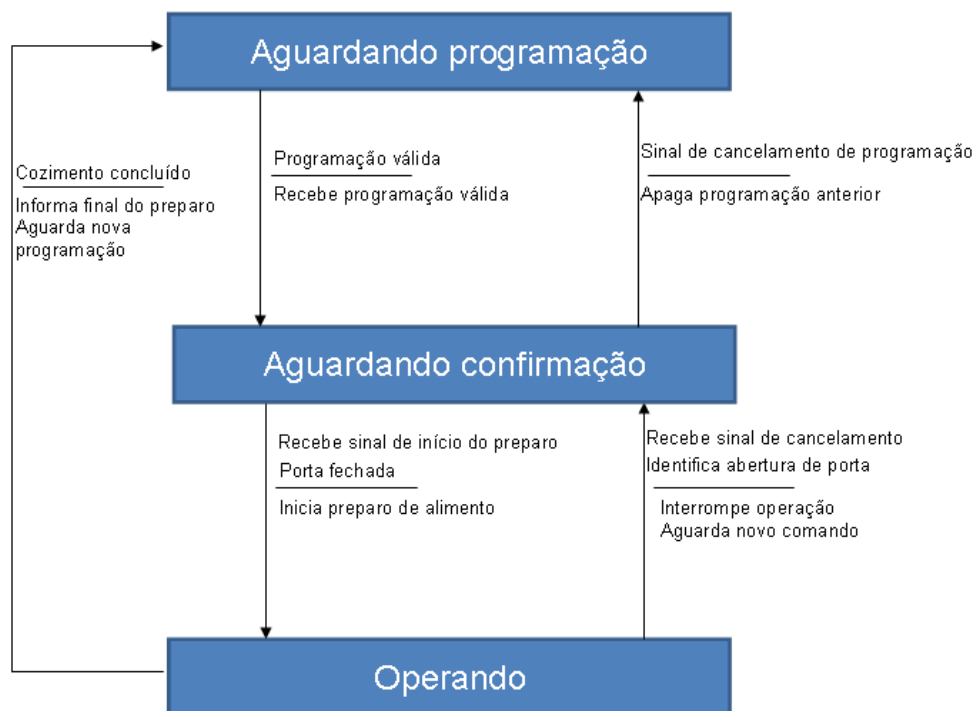


Figura 3.23 - Diagrama de transição de estados de produto.

3.3.15 Análise de Perigo e Risco para Produto e Organização

A análise de perigo e risco é realizada para cada perigo potencial identificado a partir das circunstâncias, de falhas nos fluxos entre o produto e os elementos do ambiente do produto e, das falhas nas funções internas do produto. E para a organização é identificado a partir das circunstâncias, de falhas nos fluxos entre a organização e os elementos do ambiente da organização e nas falhas nas funções internas da organização.

Um perigo é:

- qualquer dano imposto pelo **produto** ao próprio produto, ou a qualquer elemento no ambiente do produto;

- qualquer dano imposto pela **organização** à própria organização, ou a qualquer elemento no ambiente da organização.

Uma tabela de falhas e de análise crítica FMECA (JURAN e GODFREY, 1998) é construída listando as circunstâncias, as falhas nos fluxos e as falhas nas funções internas, suas conseqüências, sua gravidade, suas causas, sua probabilidade e ocorrência, sua dificuldade de detecção e o risco computado a partir da composição das avaliações de gravidade, probabilidade e dificuldade de detecção.

Dependendo da avaliação do risco, funções de mitigação do risco são identificadas. Essas funções podem ser de detecção, de prevenção, de proteção, de correção.

A Figura 3.24 ilustra uma tabela de falhas e de análise crítica para produto e organização.

	TIPO	PERIGO	CONSEQUÊNCIA	GRAV (1-5)	CAUSAS	PROB (1-5)	RISCO	FUNÇÕES
PRODUTO	circunstância	Erro do usuário - acionamento do forno com objetos metálicos em seu interior	Risco de incêndio e danos ao forno	5	efeito fotoelétrico em objetos metálicos	2	10	proteção
	passaporte	Impossibilidade de programação ou programação equivocada	Mal preparo do alimento ou impossibilidade de preparo	4	Falha das teclas	1	4	corretiva
	não função	Equipamento não aquece alimento	Não prepara o alimento	3	Módulo de potência queimada	2	6	detecção
ORGANIZAÇÃO	circunstância	Fornecedor não atende demanda	Prejudica fluxo de produção	2	Falta de peças	4	8	protetiva
	passaporte	Falha de comunicação entre desenvolvimento e manufatura	Dificuldades na manufatura	8	Barreira na comunicação	3	8	preventiva
	não função	Quebra de maquinário da linha de produção	Para linha de produção	3	Máquina quebrada	2	6	preventiva

Figura 3.24 - Tabela FMECA.

3.3.16 Atributos Funcionais de Produto e Organização

Os atributos funcionais de produto e de organização descrevem e qualificam os elementos da estrutura e comportamento funcional do produto e da organização. Eles são relacionados ao elemento como uma forma de, por exemplo, descrever seu desempenho funcional.

São normalmente apresentados ao lado do elemento do modelo, em forma textual.

3.3.17 Estrutura Funcional de Organização

As funções internas de uma organização podem ser derivadas da mesma forma como foram derivadas as funções internas de produto. No entanto, a forma mais usual de representar funções de organização é por um diagrama de atividades seguindo um *template* IDEF0 (*Integration Definition for Function Modeling*).

Esse diagrama só representa as funções ou atividades nos retângulos e os fluxos de material, energia e informação entre essas funções ou atividades. O *template* IDEF0 distingue entrada, saídas, controles e mecanismos dependendo da localização do fluxo no entorno da função. Entradas são colocadas à esquerda, saídas à direita, controles acima e mecanismos abaixo.

A Figura 3.25 ilustra esse *template* para a estrutura funcional de organização.

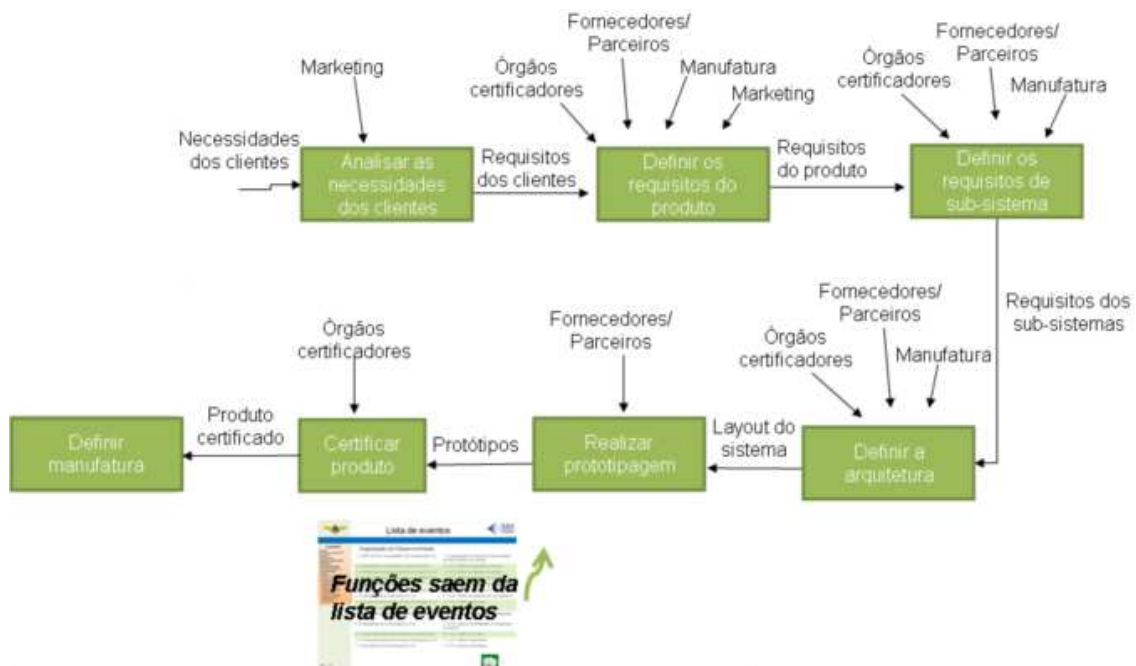


Figura 3.25 - Diagrama de estrutura funcional de organização.

3.3.18 Comportamento Funcional de Organização

O comportamento funcional de organização permite identificar a sequência em que as funções ou atividades serão executadas, para os cenários dentro do escopo do esforço de desenvolvimento.

A Figura 3.26 ilustra um diagrama para o comportamento funcional da organização.

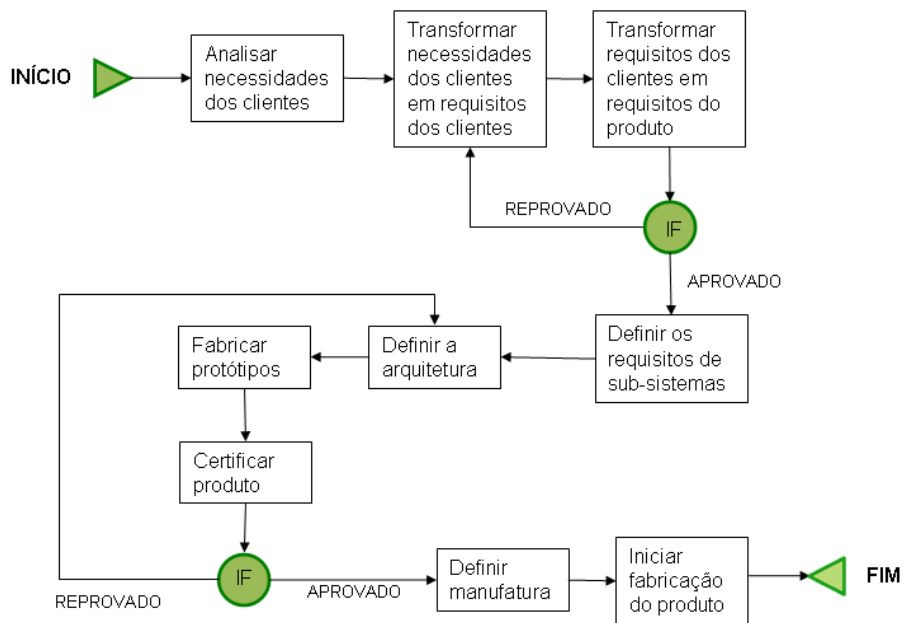


Figura 3.26 - Diagrama de comportamento funcional de organização.

3.3.19 Contexto de Implementação de Produto

O contexto de arquitetura ou de implementação para o produto é identificado, em cada cenário do processo do ciclo de vida do produto. O contexto de implementação apresenta o produto, os elementos do ambiente, e as conexões físicas entre os elementos. A diferença entre o contexto de implementação, e o contexto funcional, está nos elementos de ligação entre o sistema e os elementos do seu ambiente. No primeiro, são apresentados os fluxos de material, energia e informação. No último, são apresentadas as conexões entre o sistema e os elementos do seu ambiente. Os elementos do ambiente nos dois diagramas são os mesmos. Os fluxos do contexto funcional fluem pelas conexões do contexto de implementação.

A Figura 3.27 ilustra os elementos do ambiente e suas interconexões físicas com o sistema, no contexto de implementação.

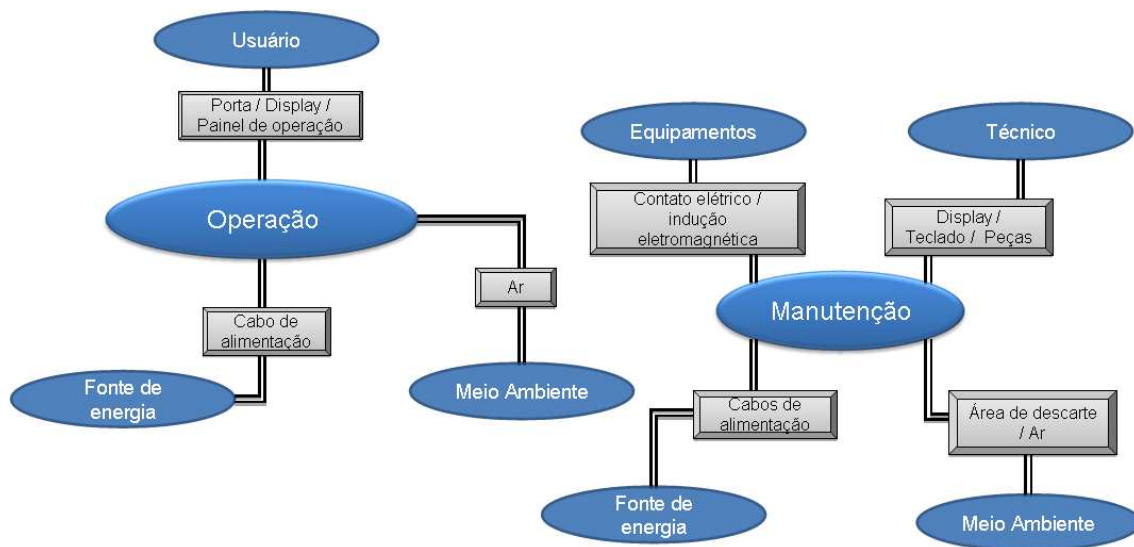


Figura 3.27 - Contexto de implementação de produto.

3.3.20 Contexto de Implementação de Organização

De maneira análoga ao contexto de implementação do produto, o contexto de implementação de organização apresenta a organização e suas conexões físicas com o seu ambiente. Conexões físicas, em se tratando de organizações, são os recursos da organização que interfaceiam com o seu ambiente. Podem ser setores organizacionais (e.g. setor de recebimento). Podem ser recursos físicos organizacionais como por exemplo, a portaria.

A Figura 3.28 ilustra um *template* representando um contexto de implementação de uma organização (retângulo). Para organização, somente são identificados os contextos de implementação para os cenários dentro do escopo do esforço de desenvolvimento.

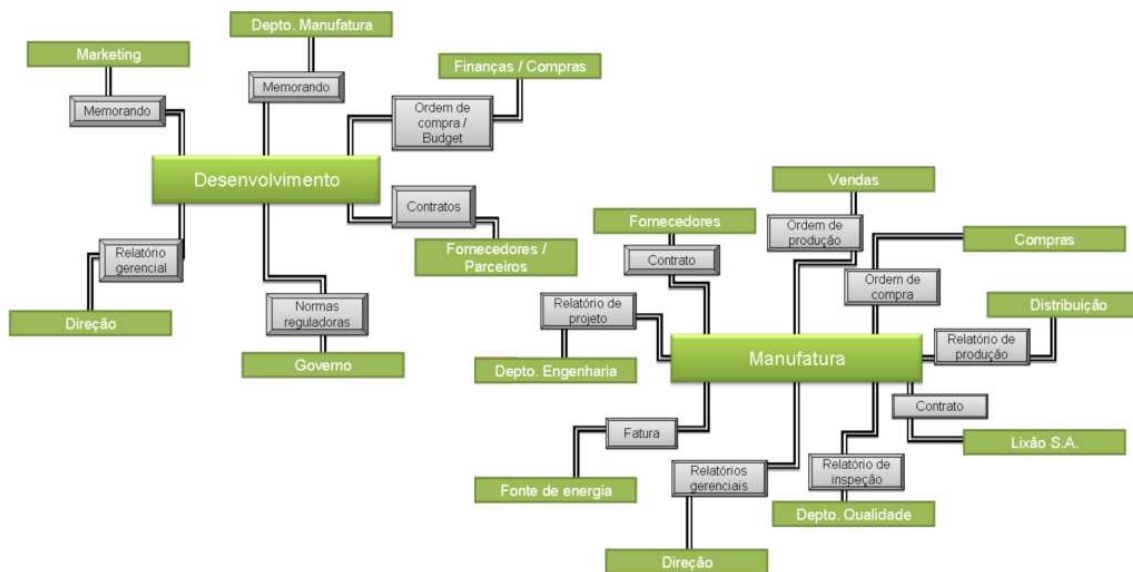


Figura 3.28 - Contexto de implementação de organização.

3.3.21 Requisitos de Interface Externa de Produto

As conexões físicas entre o sistema e os elementos do ambiente, definem os requisitos de interface externa de produto.

3.3.22 Requisitos de Interface Externa de Organização

Para a organização que implementará os processos do ciclo de vida, também são definidos os requisitos de interface externa, pelas conexões físicas entre a organização e os elementos do ambiente.

3.3.23 Interconexões de Arquitetura de Implementação de Produto

O modelo que representa as interconexões de arquitetura da implementação de um produto, é um desdobramento do contexto de arquitetura de

implementação. Esse modelo contém as partes constituintes do produto, e as conexões físicas entre essas partes. Essas conexões físicas são as interfaces físicas internas. A Figura 3.29 ilustra as interconexões de arquitetura de implementação para produto.

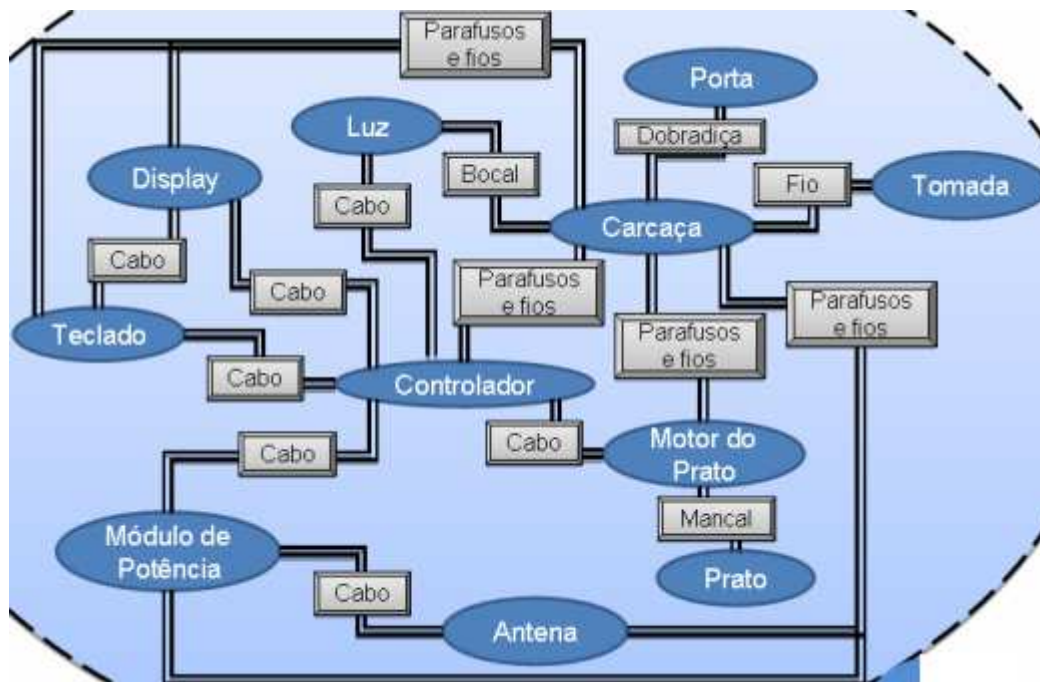


Figura 3.29 - Interconexão de arquitetura de implementação de produto.

3.3.24 Fluxos de Arquitetura de Implementação de Produto

O modelo que representa os fluxos de arquitetura da implementação de um produto, é um desdobramento do contexto de arquitetura de implementação. Esse modelo contém as partes constituintes do produto e o que flui sobre as conexões físicas entre essas partes. A Figura 3.30 ilustra os fluxos de arquitetura de implementação para produto.

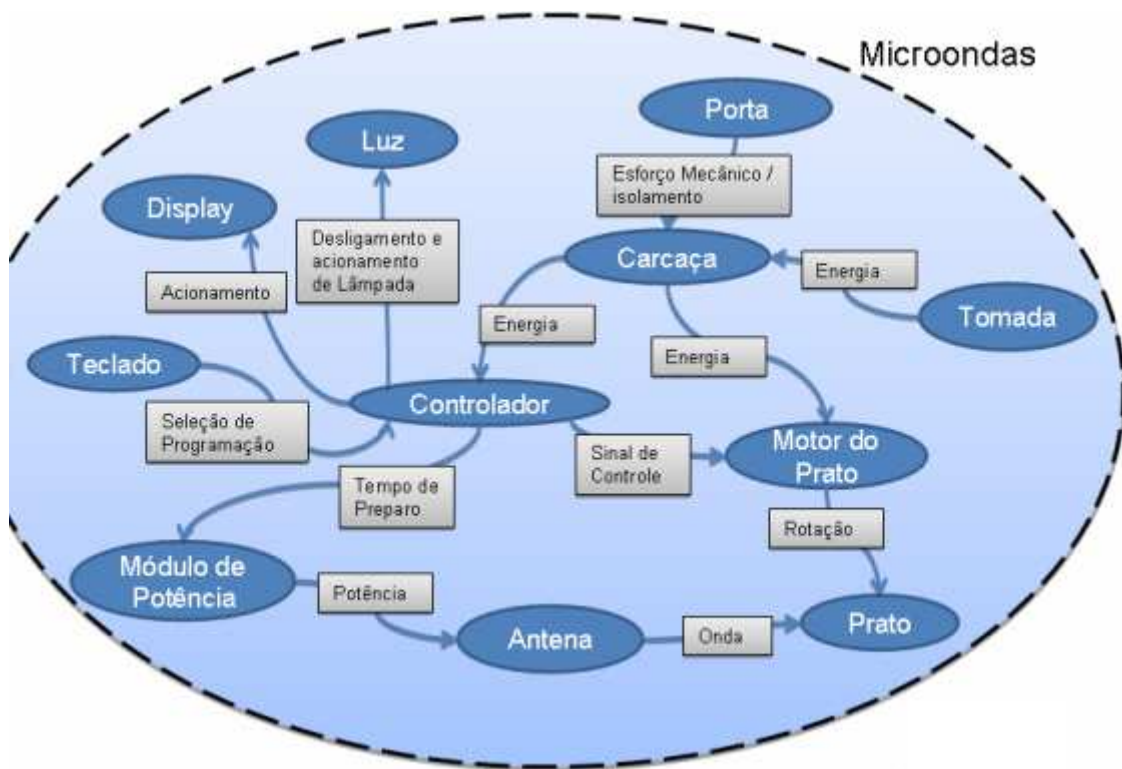


Figura 3.30 - Fluxos de arquitetura de implementação de produto.

3.3.25 Atributos de Implementação de Produto e Organização

Os atributos de implementação do produto descrevem os elementos dos modelos de arquitetura de implementação do produto, sejam eles partes componentes, fluxos ou interconexões físicas.

Os atributos de implementação da organização descrevem os elementos dos modelos de arquitetura de implementação da organização, sejam eles setores/recursos componentes, fluxos ou interconexões físicas.

Os atributos de implementação acrescentam características de desempenho, ou limitações físicas a cada elemento dos modelos de implementação. Eles são

apresentados textualmente e escritos ao lado de cada elemento dos modelos de implementação.

3.3.26 Interconexão de Arquitetura de Implementação de Organização

O modelo que representa as interconexões de arquitetura da implementação de uma organização, é um desdobramento do contexto de arquitetura de implementação. Esse modelo contém os setores ou recursos componentes da organização, e os recursos ou setores que interligam esses componentes. Esses recursos ou setores que interligam outros são as interfaces físicas internas. A Figura 3.31 ilustra a interconexão de arquitetura de implementação para organização.

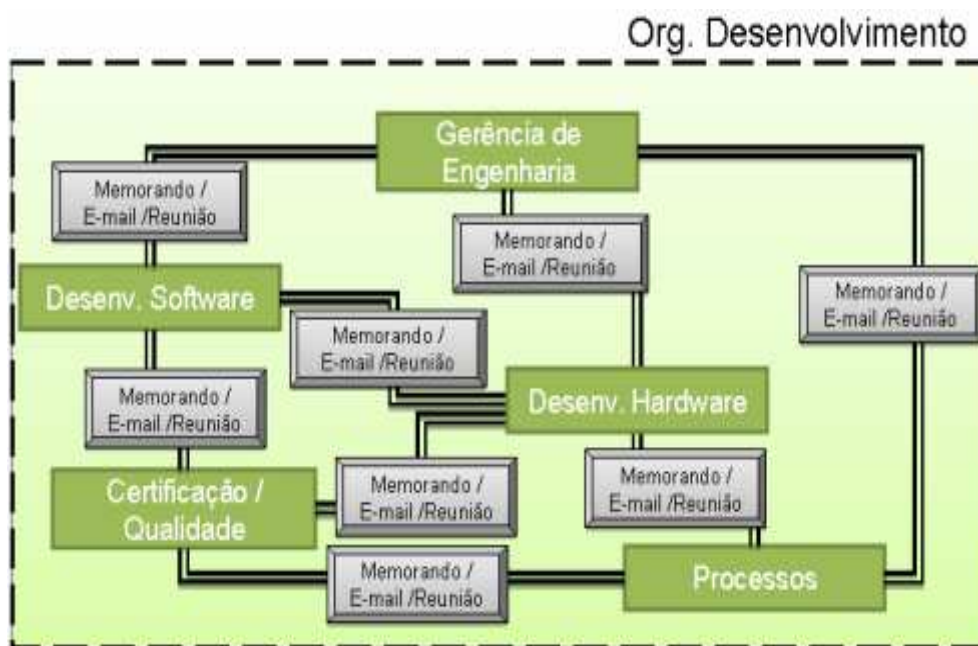


Figura 3.31 - Interconexão de arquitetura de implementação de organização.

3.3.27 Fluxos de Arquitetura de Implementação de Organização

O modelo que representa os fluxos de arquitetura de implementação de uma organização, é um desdobramento do contexto de arquitetura de implementação. Esse modelo contém os setores ou recursos componentes da organização, e que flui nas interfaces entre esses componentes. A Figura 3.32 ilustra os fluxos de arquitetura de implementação para organização.

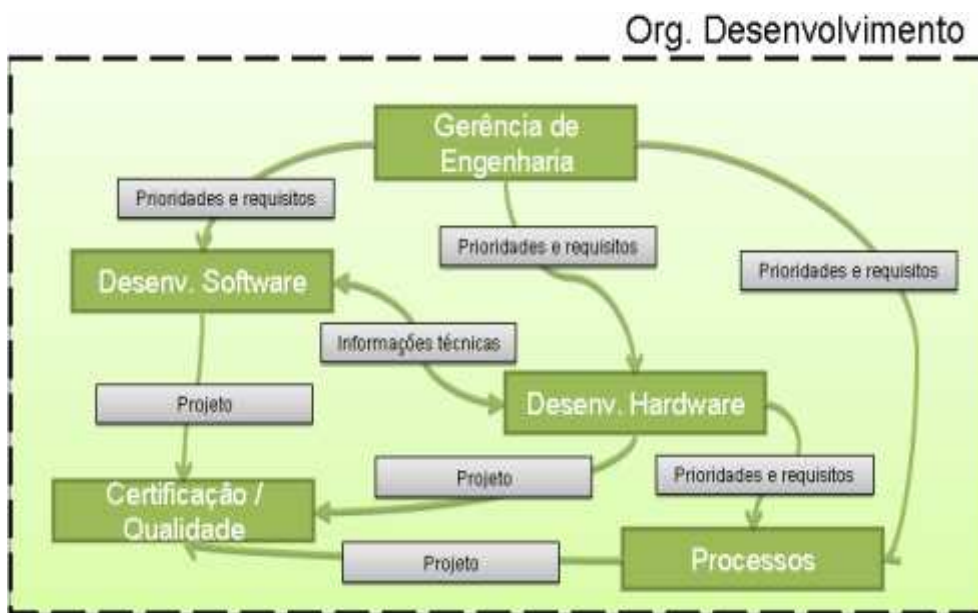


Figura 3.32 – Fluxos de arquitetura de implementação de organização.

3.3.28 Relacionamento entre Atributos

Os atributos obtidos ao longo do processo de engenharia simultânea de sistemas descritos, são dispostos em matrizes e relacionados entre si. Essas matrizes podem conter em suas linhas e colunas, os diversos tipos de atributos obtidos, sejam eles funcionais de produto, funcionais de organização, de implementação de produto ou de implementação de organização. O relacionamento entre atributos funcionais e de implementação representam a

alocação e a rastreabilidade entre os modelos funcionais e físicos. Os relacionamentos entre atributos de produto e de organização representam os impactos mútuos entre produto e organização, e precisam ser identificados para o desenvolvimento de soluções de engenharia simultânea de sistemas.

3.3.29 Arquitetura de Sistemas

A arquitetura do sistema é o resultado de todo o processo descrito acima, com todos os elementos identificados e realizados ao longo desse processo. Ela apresenta então os elementos de produto e de organização, sejam eles partes, fluxos, interconexões físicas, atributos e relacionamento entre os atributos. A arquitetura de sistemas provê as entradas necessárias para os processos de projeto detalhado, aquisição ou reuso de partes componentes, integração e testes do sistema composto por elementos de produto e de organização.

3.4 Relacionamento entre os Processos de Análise e os Elementos do Método

Os elementos descritos nas Subseções 3.3.1 a 3.3.29 são os elementos destacados na Figura 3.3.

Esses são os elementos que pertencem à análise de *stakeholders*, à análise de requisitos, à análise funcional e à análise de implementação, realizadas simultaneamente para produto e organização, correspondentes a dimensão de análise ilustrada no *framework* da Figura 3.1.

A seguir são agrupados os elementos dentro de cada análise, são eles:

Na análise de *stakeholders*:

- *stakeholders* de produto e seus interesses;
- *stakeholders* de organização e seus interesses;

- as medidas de efetividade (MoEs).

Na análise de requisitos:

- os requisitos de *stakeholders* de produto;
- os requisitos de *stakeholders* de organização;
- os requisitos de sistema.

Na análise funcional:

- contexto funcional de produto, e contexto funcional de organização;
- modos para produto e para organização;
- estrutura funcional de produto, e estrutura funcional de organização;
- comportamento funcional de produto, e comportamento funcional de organização;
- análise de perigo e risco para produto e organização;
- atributos funcionais de produto, e atributos funcionais de organização.

Na análise de implementação:

- contexto de implementação de produto, e contexto de implementação de organização;
- requisitos de interface externa de produto, e requisitos de interface externa de organização;
- interconexões de arquitetura de implementação de produto, e interconexões de arquitetura de implementação de organização;
- fluxos de arquitetura de implementação de produto, e fluxos de arquitetura de implementação de organização;
- atributos de implementação de produto, e atributos de implementação de organização.

3.5 Consistência entre o *Framework* e o Método

A Figura 3.33 mostra a consistência entre o *framework* (Figura 3.1) e o método (Figura 3.2). Missão, processos, cenários dos processos, sistema e arquitetura de sistemas apresentam diferentes camadas da dimensão de estrutura do *framework*. Essa estrutura é decomposta sucessivamente, da mesma forma, a partir dos subsistemas. A dimensão de integração do *framework* está claramente representada pela engenharia simultânea de produto e organização. A dimensão de análise permeia o processo desde a identificação dos *stakeholders*, passando pela análise de requisitos, análise funcional até chegar aos modelos de arquitetura de implementação.

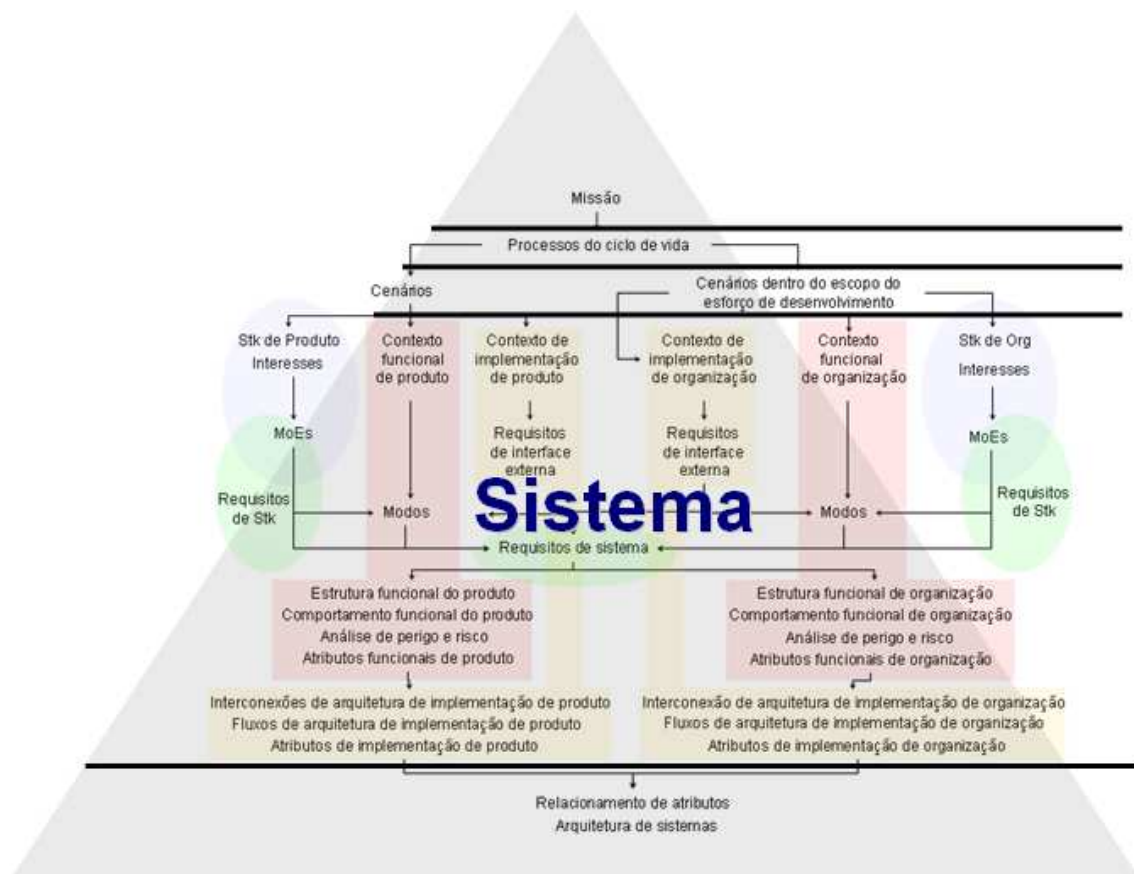


Figura 3.33 - Elementos do processo de engenharia simultânea de sistemas em níveis de abstração.

4 AUXÍLIO COMPUTACIONAL AO PROCESSO

Este Capítulo descreve um auxílio computacional ao processo descrito no Capítulo 3 usando a ferramenta computacional comercial ambiente de engenharia de sistemas chamada *Cradle*.

A Seção 4.1 descreve o que é o *Cradle* e o que se pode fazer com o *Cradle*, e são descritas as notações dos diagramas que o *Cradle* possui.

A Seção 4.2 descreve um método de auxílio computacional ao processo de engenharia simultânea de sistemas, descrito no Capítulo 3. Esse método usa o *Cradle*.

4.1 *Cradle*

Cradle é um ambiente de engenharia de sistemas e de gerenciamento de requisitos, desenvolvido pela 3SL (3SL, 2011b). Essa ferramenta permite gestão de requisitos, análise e projeto de sistema, definição de arquitetura de sistemas e modelagem de processo de negócios. É possível manter rastreabilidade desde a fase de concepção do projeto até sua implementação pelos *links* entre os itens de informação, elementos dos diagramas que representam modelagem funcional e elementos dos diagramas que representam modelagem de implementação (3SL, 2011a).

Cradle possui um pacote de ferramentas de modelagem gráfica para construir processos de negócio, análise e modelos de arquitetura e projeto. É pela combinação de notação de modelos que o *Cradle* permite descrever o processo de engenharia de sistemas, em vários níveis de abstração (3SL, 2011a).

As notações para modelos de estrutura são:

- diagrama de definição integrada para modelagem funcional - *integrated definition for functional modeling* (IDEF0);
- diagrama de fluxo de dados - *data flow diagram* (DFD);
- diagrama de interconexão de arquitetura - *architecture interconnect diagram* (AID);
- diagrama de arquitetura física - *physical architecture diagram* (PAD).

As notações para modelos de comportamento são:

- diagrama de transição de estados - *state transition diagram* (STD);
- diagrama de bloco de fluxo funcional - *enhanced functional flow block diagram* (eFFBD).

Os diagramas da UML 2.0 complementam o pacote de notações do *Cradle* (3SL, 2009).

Os diagramas no *Cradle* estão inseridos em dois domínios, o domínio essencial e o domínio de implementação. O domínio essencial é para descrever a concepção de um sistema no mais alto nível de abstração, sem levar em consideração as alternativas de implementação do sistema. Já no domínio de implementação o sistema é descrito pelos seus componentes físicos, suas conexões e associações.

As funcionalidades do *Cradle* são integradas para cobrir todo o ciclo de vida do processo de desenvolvimento, e permite com o uso de *links* manter a rastreabilidade do projeto (3SL, 2009).

É com esse ambiente engenharia de sistemas comercial que serão capturados os elementos da dimensão de análise, da dimensão de integração e da dimensão da estrutura do sistema conforme *framework* ilustrado na Figura 3.1,

ilustrados nos exemplos do CBERS e do VLM nos Capítulos 5 e 6 respectivamente.

4.2 Método

Esta Seção descreve o método em que se capturam os elementos do processo de engenharia simultânea de sistemas, apresentado no Capítulo 3, utilizando o pacote de diagramas que o *Cradle* possui para descrever o domínio essencial, e o domínio de implementação.

Usando os diagramas e os itens de informação do *Cradle*, descreve-se o sistema ou uma hierarquia de sistemas, partindo do mais alto nível de abstração. *Cradle* permite que desde o mais alto nível de abstração, o sistema seja modelado em seus diversos processos do ciclo de vida.

A rastreabilidade entre o domínio essencial e o domínio de implementação, é mantida por *links* entre os elementos dos diagramas, e dos itens de informação.

Os diagramas do domínio essencial que o *Cradle* possui são:

- diagrama de fluxo de dados - *data flow diagram* (DFD);
- diagrama entidade relacionamento - *entity relationship diagram* (ERD);
- diagrama de transição de estados - *state transition diagram* (STD);
- diagrama de estrutura de dados - *data structure diagram* (DSD);
- diagrama de definição integrada para modelagem funcional - *integrated definition for functional modeling* (IDEF0);
- diagrama de bloco de fluxo funcional - *enhanced functional flow block diagram* (eFFBD);
- diagrama de fluxo de processo - *process flow diagram* (PFD);
- diagrama de caso de uso - *use case diagram* (UCD);

- diagrama de pacote - *package diagram* (PD);
- diagrama de sequência - *sequence diagram* (SQD);
- diagrama de colaboração - *collaboration diagram* (COD);
- diagrama de classe - *class diagram* (CD);
- diagrama de estados - *statechart diagram* (SCD);
- diagrama de atividades - *activity diagram* (ACD).

No domínio de implementação são compreendidas as mesmas notações do domínio essencial, mais as seguintes notações:

- diagrama de interconexão de arquitetura - *architecture interconnect diagram* (AID);
- diagrama de arquitetura física - *physical architecture diagram* (PAD);
- diagrama de componentes - *component diagram* (CPD);
- diagrama de desdobramento - *deployment diagram* (DPD);
- diagrama de estrutura gráfica - *structure chart* (STC);
- diagrama de arquitetura de software - *software architecture* (SAD).

A Figura 4.1 ilustra os domínios essencial e o domínio de implementação do *Cradle*. Cada domínio possui um pacote de notações gráficas, como descrito nesta Seção.

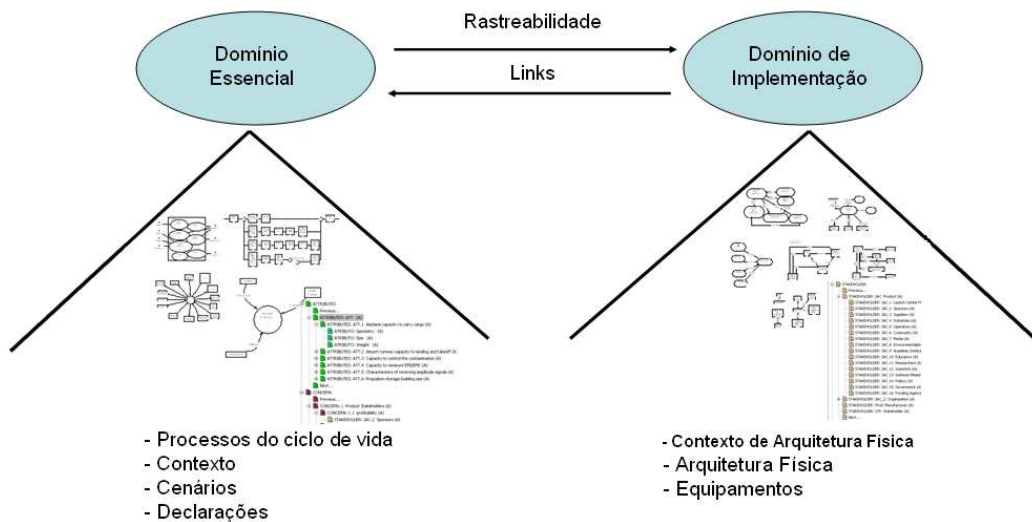


Figura 4.1 - Modelo de rastreabilidade do projeto.

A rastreabilidade é mantida entre o domínio essencial e o domínio de implementação pelos *links* entre os elementos correspondentes do projeto. Os *links* podem ser do tipo de associação de declarações de interesses no processo do ciclo de vida do produto. Os *links* auxiliam na descrição do impacto no desenvolvimento do produto. Pelos *links* é possível descrever as associações entre os itens de informação e os elementos associáveis dos diagramas do domínio essencial e do domínio de implementação.

No mais alto nível de abstração eles são associados aos elementos que não descrevem a solução do problema. No nível de implementação é possível descrever os elementos da solução do problema.

A Figura 4.2 ilustra um item de informação no *Cradle*.

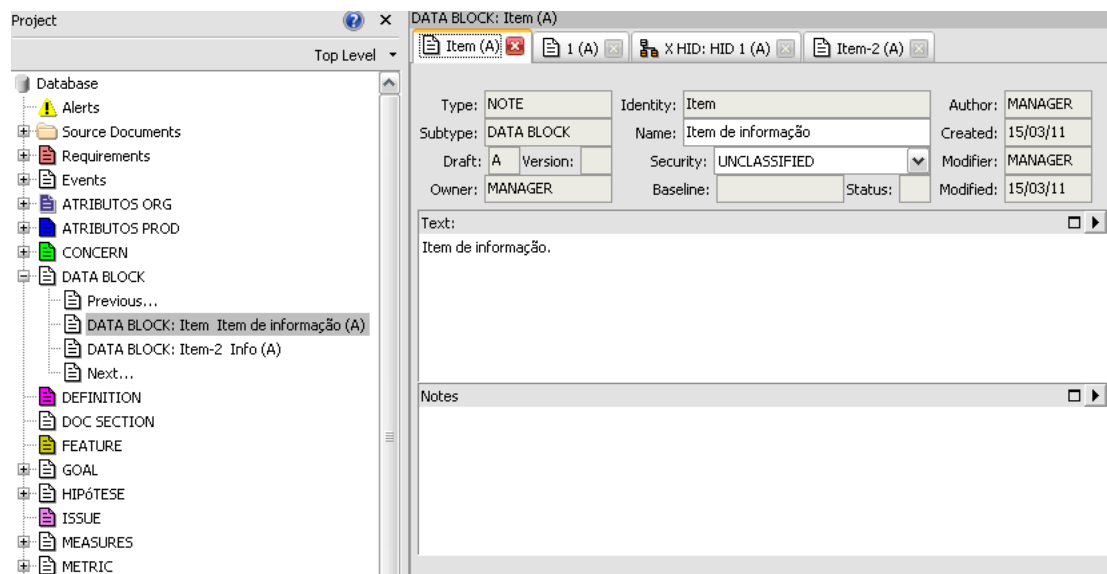


Figura 4.2 - Item de informação gerado no *Cradle*.

A Figura 4.3 ilustra como o item de informação é exibido por associações usando os links de informação.

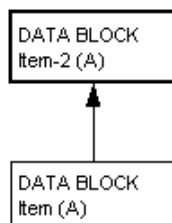


Figura 4.3 - Associação entre itens de informação por *links*.

A Figura 4.4 ilustra os elementos do processo de engenharia simultânea de sistemas cobertos pelos diagramas do *Cradle* para o auxílio computacional ao processo, capturando cada elemento do processo de engenharia simultânea de sistemas.

Elementos do Processo de ESS	Modelagem do Processo usando Cradle
Missão	Item de informação do tipo requisito
Processos do ciclo de vida	eFFBD (Essencial)
Escopo do esforço de desenvolvimento	UCD (Essencial)
Cenários	eFFBD (Essencial)
Stakeholder de produto e organização	Itens de informação e DFD (Essencial)
Interesses	Itens de informação e DFD (Essencial)
MoE's	Itens de informação
Requisitos de produto e organização	Itens de informação do tipo requisito
Contexto funcional de produto e organização	DFD (Essencial)
Comportamento funcional de produto	STD (Essencial)
Comportamento funcional de organização	eFFBD (Essencial)
Análise de perigo e risco	Itens de informação do tipo risco
Contexto de implementação de produto	PAD (Implementação)
Contexto de implementação de organização	AID (Implementação)
Atributos de produto e organização	Itens de informação
Interconexões de arquitetura de implementação de produto	PAD (Implementação)
Interconexões de arquitetura de implementação de organização	AID (Implementação)
Fluxos de arquitetura de implementação de produto	PAD (Implementação)
Fluxos de arquitetura de implementação de organização	AID (Implementação)
Relacionamentos	Referências cruzadas (links)

Figura 4.4 - Elementos do processo de engenharia simultânea de sistemas cobertos pelos diagramas do *Cradle*.

Para os modelos descreverem os processos do ciclo de vida por exemplo, é utilizado um diagrama de comportamento do tipo eFFBD para tornar seqüencial os processos e serem posteriormente desdobrados em cenários.

Para os *stakeholders* de produto e organização e seus interesses são utilizados os diagramas do tipo DFD. Esse diagrama permite capturar os *stakeholders* em seus terminadores sendo o centro da bolha o processo ou cenário do produto ou organização, e nas setas entre terminadores e a bolha são capturados os interesses de *stakeholders*.

Os requisitos de produto e organização são itens de informação no *Cradle* do tipo requisito. Esse item do tipo requisito possui atributos para a gestão e engenharia dos requisitos, que serão adiante associados aos elementos que serão parte do desdobramento do sistema pelos modelos. Eles serão associados aos diversos outros tipos de itens de informação necessários para auxiliar a descrever o processo de engenharia simultânea de sistemas.

Para o contexto funcional de produto e da organização é utilizado o diagrama do tipo DFD, onde seus terminadores capturam os elementos do ambiente que trocam material, energia e informação com o sistema. Por exemplo, a bolha do DFD representa o produto em operação ou a organização de desenvolvimento, e as setas entre os elementos do ambiente capturam os fluxos de material, energia e informação.

Para o contexto de implementação de produto é utilizado o diagrama do tipo PAD, onde são capturados os elementos físicos do produto, suas interconexões e os fluxos. Para o contexto de implementação de organização é utilizado o diagrama do tipo AID, onde são capturados os elementos de implementação da organização, assim como suas interconexões e fluxos.

Para associar os itens de informação gerados pelo *Cradle* são utilizados os *links* de informação. Esses *links* possuem atributos que descrevem a razão, o motivo e permite a gestão do usuário ao criar ou gerenciar esses itens.

4.2.1 Missão, Processos do Ciclo de Vida e Cenários

Nesta Subseção são representados os elementos do processo de engenharia simultânea de sistemas descritos nas Subseções 3.3.1 a 3.3.4.

A declaração da missão do sistema pode ser representada por um item de informação. Esse item de informação pode ser do tipo requisito, que pode ser associado aos interesses de *stakeholders*. A Figura 4.5 ilustra a captura da missão por um item de informação.

The screenshot shows a software requirement management tool interface. At the top, there are two tabs: 'untitled' and 'REQ-0 (A)'. The main window displays a form for a requirement item. The 'Name' field is filled with 'Missão'. The 'Identity' field contains 'REQ-0'. The 'Item Status' is a dropdown menu. The 'Author' is 'MANAGER'. The 'Classification' is 'UNCLASSIFIED'. The 'Created on' date is '17/03/11'. The 'Draft' is 'A' and the 'Version' is empty. The 'Baseline' and 'Status' fields are empty. The 'Last modifier' is 'MANAGER' and the 'Last modified' date is '17/03/11'. Below these fields, there is a 'Picture' dropdown menu. To the right, there are three dropdown menus for 'Req Cat:', 'Priority:', and 'Domain:'. Below these are three buttons: 'Requirement Details:', 'Acceptance Criteria:', and 'Compliance:', each with a checkbox and a right-pointing arrow. At the bottom right, there is an 'Annotations' button. The 'Notes' field contains the text 'O sistema deve ...'. The 'Reference:' field is empty.

Figura 4.5 - Declaração da missão do sistema.

Em um nível mais alto de abstração, pode ser representado pelo diagrama de comportamento do tipo eFFBD, para ilustrar a seqüência dos processos do ciclo de vida do sistema. O eFFBD é utilizado para ilustrar os processos do ciclo de vida do produto, e pode ser desdobrado em outros elementos do diagrama, para representar os cenários do processo do ciclo de vida.

O diagrama ilustrado na Figura 4.6 descreve a seqüência dos processos do ciclo de vida. Os processos 3 e 4 são ilustrados para mostrar que são realizados simultaneamente. As atividades iniciam-se no processo 1 e

encerram-se no processo 5. As atividades são encerradas, quando todos os processos forem executados.

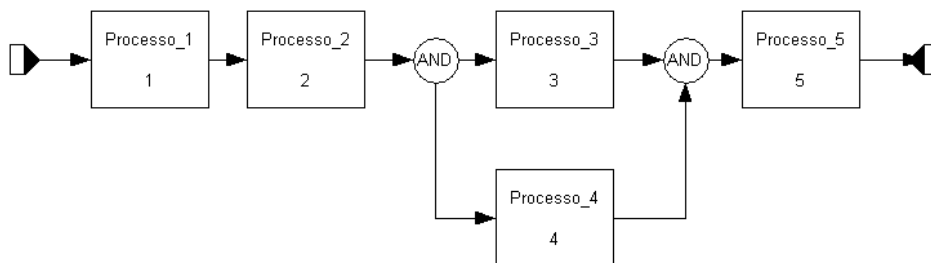


Figura 4.6 - Diagrama de comportamento eFFBD do domínio essencial do *Cradle*, representando os processos do ciclo de vida.

Para um nível mais detalhado o eFFBD pode ser usado para descrever, as atividades que serão desempenhadas para a realização do processo. E podem ser desdobradas para representar os cenários dos processos do ciclo de vida. A Figura 4.7 ilustra o desdobramento do processo 2 nos seus cenários.

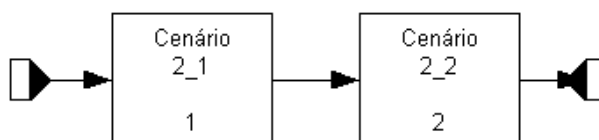


Figura 4.7 - Desdobramento de processo em cenários.

Os cenários dos processos do ciclo de vida ilustrados pelo diagrama eFFBD, podem ser desdobrados para outros diagramas como por exemplo DFD. Nesta etapa, todos os elementos gerados ficam armazenados no domínio essencial.

4.2.2 Análise de *Stakeholders* e Requisitos

Nesta Subseção são representados os elementos do processo das Subseções de 3.3.5 a 3.3.10, que estão compreendidas dentro da análise de *stakeholders*, como descrito na Seção 3.4.

Os interesses dos *stakeholders* para os processos, ou cenários do ciclo de vida, são capturados utilizando um diagrama DFD. A Figura 4.8 ilustra o *stakeholder* e seu interesse, para um cenário do processo do ciclo de vida.

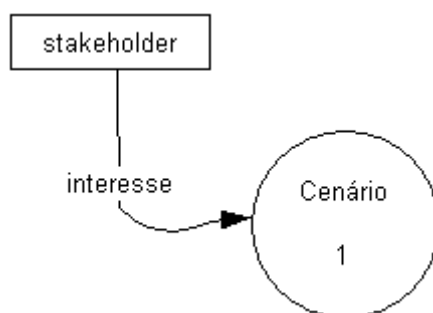


Figura 4.8 - *Stakeholder* e seu interesse no cenário do processo do ciclo de vida do sistema.

A partir do diagrama da Figura 4.8 são capturados os *stakeholders* e seus interesses, usando itens de informação, que servirão para derivar as medidas de efetividade (MoEs) e os requisitos de sistema.

A Figura 4.9 ilustra o item de informação *stakeholder* e seu interesse, o requisito de *stakeholder*, a MoE, o requisito de sistema e a associação entre eles. São pelos *links* que os associam que é possível manter a rastreabilidade.

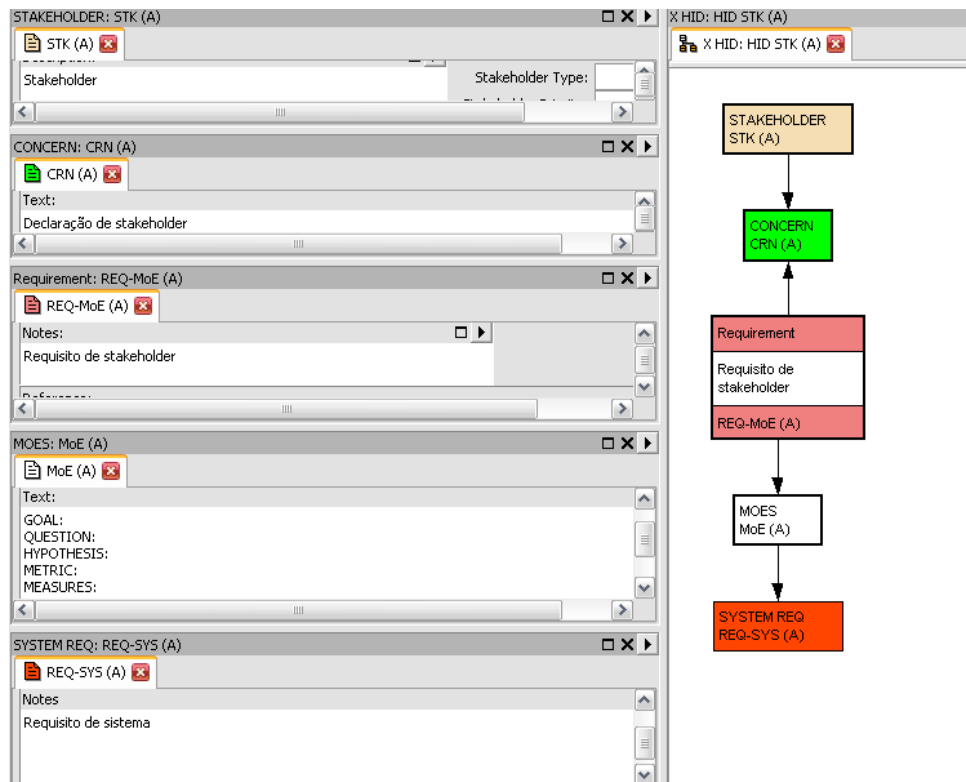


Figura 4.9 - Elementos da análise de requisitos associados por itens de informação.

4.2.3 Análise Funcional

Nesta Subseção são representados os elementos do processo de engenharia simultânea de sistemas, descritos nas Subseções de 3.3.11 a 3.3.18, compreendidos na análise funcional descrita na Seção 3.4.

Para capturar as interações do sistema com os elementos do ambiente, são utilizados os diagramas DFD.

Os fluxos são descritos nas setas que ligam os elementos do ambiente ao processo, sistema, subsistema ou parte.

A Figura 4.10 ilustra um diagrama DFD utilizado para a captura dos fluxos de energia, material e informação, dentro de um contexto ou em um modo do sistema.

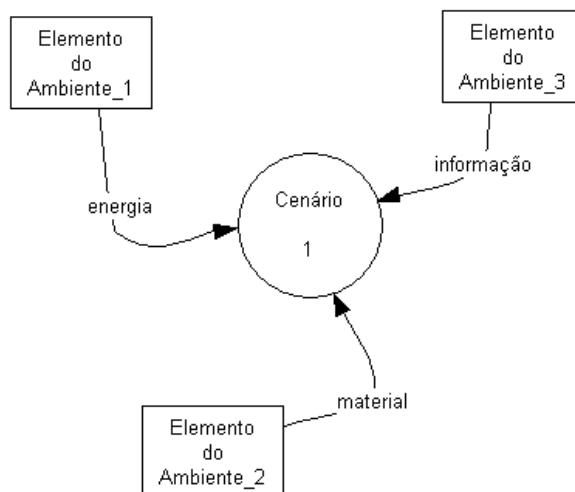


Figura 4.10 - Diagrama representando o sistema em um contexto funcional, ou em um modo, que troca energia, material e informação.

O diagrama no contexto funcional é desdobrado para identificar as funções que o sistema irá desempenhar. Para isso o DFD pode ser desdobrado em um modelo de comportamento, e em um modelo de estrutura, conforme descrito na Seção 4.1. O *Cradle* possui um item de informação chamado *events*, que é utilizado para realizar a lista de eventos, e auxiliar na identificação das funções.

Para as funções identificadas, são associados itens de informação do tipo risco, para a análise de perigo e risco. A Figura 4.11 ilustra essa descrição.

Na Figura 4.11 onde está representado o número 1, é a captura do cenário e os elementos do ambiente, onde cada elemento do ambiente possui estados. O conjunto desses estados é chamado circunstâncias, e pode ser descrito no próprio item que representa o elemento do ambiente no *Cradle*, como ilustra a Figura 4.12.

No número 2 está representado o comportamento do sistema, e no número 3 está representada a estrutura do sistema. A partir desses modelos, cada elemento do diagrama mais os elementos do ambiente do cenário, preenchem a lista de eventos, para a identificação de funções. A lista de eventos está ilustrada no número 4.

O número 5 ilustra uma forma de análise no *Cradle*. De forma que é possível enxergar a associação entre os itens de informação. O item de informação do tipo risco está associado a uma função, que é originada do desdobramento do cenário funcional.

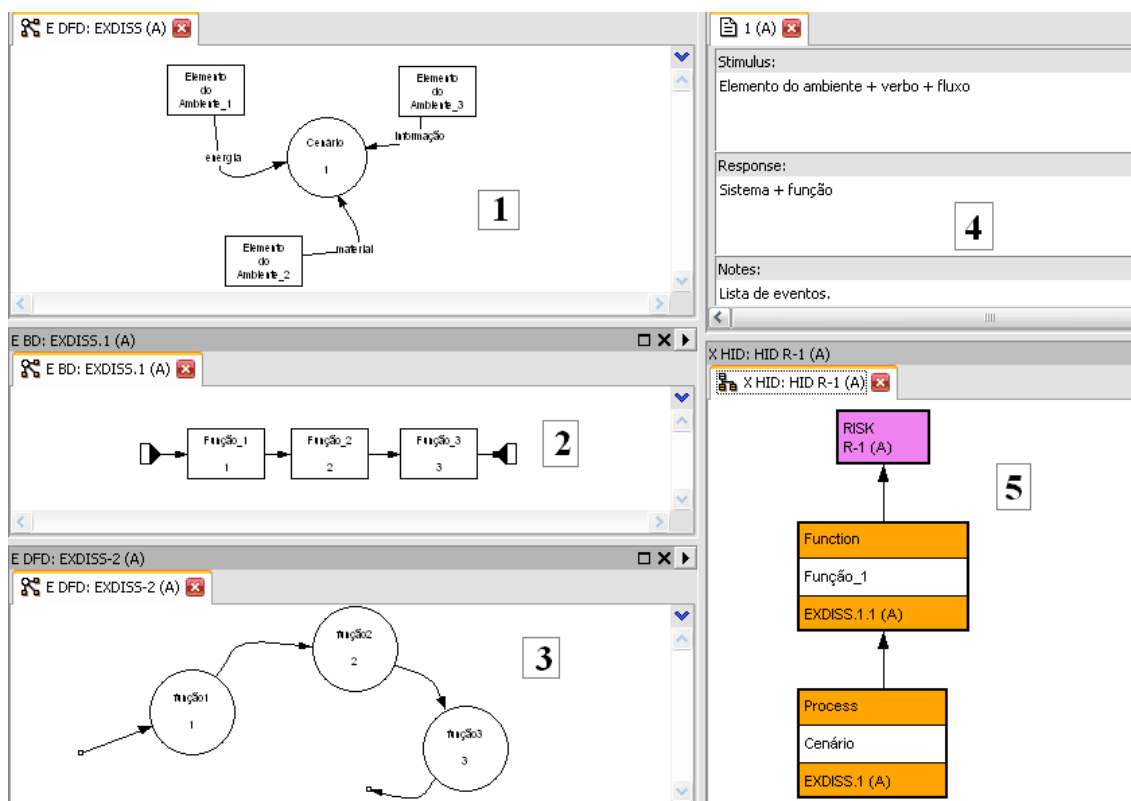


Figura 4.11 - Contexto funcional desdobrado em comportamento e estrutura, para identificar as funções do sistema, e o risco associado a cada função.

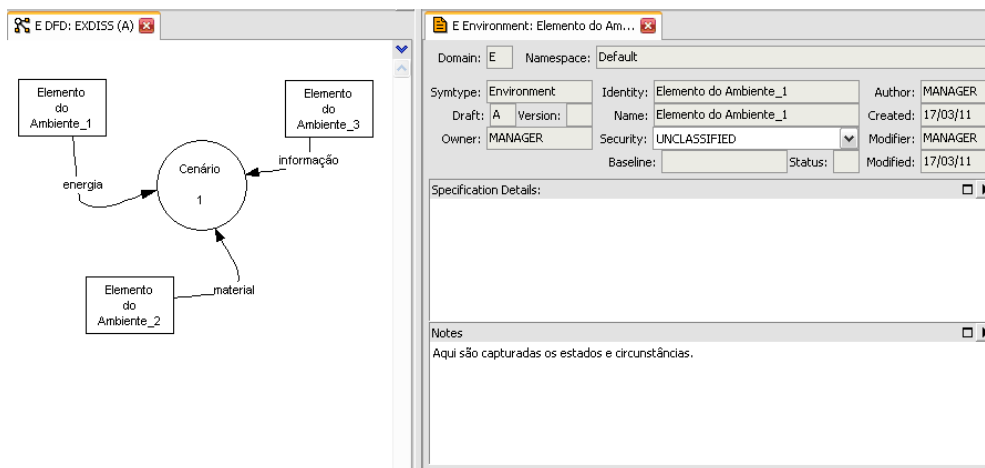


Figura 4.12 - Captura dos estados e circunstâncias.

4.2.4 Análise de Implementação

Nesta Subseção são representados os elementos do processo de engenharia simultânea de sistemas, descritos nas Subseções de 3.3.19 a 3.3.27, compreendidas na análise de implementação descrita na Seção 3.4.

A partir de um problema descrito no domínio essencial, pelos diagramas que representam o nível de abstração mais alto, o processo de desenvolvimento do sistema parte para um nível menor de abstração, até o nível de implementação dos elementos da solução do sistema.

Para ilustrar a implementação de um sistema, subsistema ou componente, os diagramas utilizados são os diagramas que representam a arquitetura física e a interconexão física do sistema ou produto no *Cradle*. A seta indicando a conexão entre os elementos do diagrama, pode ser um canal de transferência de dados, sinais, cabos ou interfaces. Estes diagramas pertencem ao domínio de implementação.

A Figura 4.13 ilustra um diagrama de arquitetura física, PAD e AID respectivamente, em 1 e 2. No número 1, estão representados dois equipamentos, ou elementos físicos para produto. No número 2 estão representados os elementos de implementação da organização e suas interfaces.

As setas aqui também permitem serem associadas aos itens de informação. São estes itens que capturam o tipo de fluxo ou de conexão de arquitetura, e as associações entre os elementos físicos dos diagramas no domínio de implementação.

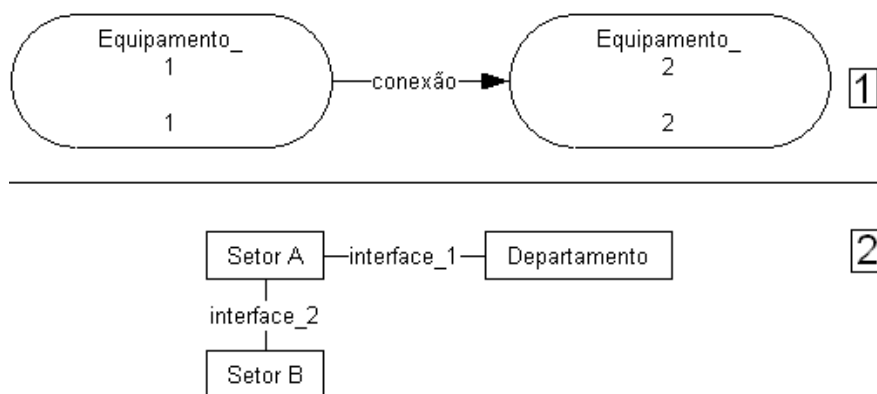


Figura 4.13 - Elementos físicos da análise de implementação.

5 APLICAÇÃO DO MÉTODO PARA O CBERS

Neste Capítulo é aplicado o método descrito no Capítulo 4, capturando os elementos processo descrito no Capítulo 3. São ilustrados os processos do ciclo de vida e do esforço do escopo de desenvolvimento, os interesses dos *stakeholders*, a troca de energia, material e informação entre os elementos do ambiente que interagem com o sistema, e as interfaces físicas e os fluxos de arquitetura para a análise de implementação do sistema.

São ilustrados os itens de informação que endereçam os *stakeholders*, seus interesses, as medidas de efetividades e os riscos associados à função que o sistema deve desempenhar. Esses itens de informação permitem ser associados aos elementos gerados pelos diagramas da análise funcional e de implementação. São pelos *links* que se podem navegar entre os elementos do projeto, e manter a rastreabilidade desde a concepção do sistema até a implementação de sua arquitetura.

Atributos são derivados dos modelos da análise funcional e da análise de implementação. Seus relacionamentos permitem identificar requisitos impostos pelo produto, e também pela organização que desempenhará os processos do ciclo de vida do produto.

5.1 CBERS II

A aplicação do processo de engenharia simultânea de sistemas auxiliado por computador é exemplificada aqui para o Satélite Sino-Brasileiro de Recursos Terrestres - *China-Brazil Earth Resources Satellite (CBERS)*. O CBERS é uma parceria entre Brasil e China para o desenvolvimento de satélite de sensoriamento remoto. (INPE, 2011b).

5.1.1 Descrição da Missão

Aqui é descrita a missão conforme a Subseção 3.3.1, e para o método usando *Cradle* descrito na Subseção 4.2.1.

A missão do CBERS para a parte brasileira é a de monitorar os recursos terrestres do vasto território brasileiro. Suas imagens são usadas em campos como o controle de desmatamento e queimadas da Amazônia Legal, o monitoramento de recursos hídricos, áreas agrícolas, crescimento urbano, ocupação do solo e entre outras diversas aplicações. (INPE, 2011b).

5.1.2 Ciclo de Vida e Escopo do Esforço de Desenvolvimento

Aqui são descritos os processos do ciclo de vida, o escopo do esforço de desenvolvimento e os cenários, conforme Subseções 3.3.2 a 3.3.4, usando o *Cradle* para o método, como descrito na Subseção 4.2.1.

Os processos do ciclo de vida são capturados e descritos pelo diagrama de comportamento (eFFBD) ilustrado na Figura 5.1. Os processos são desdobrados em outros componentes do diagrama. Por exemplo, na Figura 5.1 o processo 'Fase C Definição Detalhada' é desdobrado em 'Desenvolvimento do Projeto e Construção' na Figura 5.2.

Os cenários levados em consideração para ilustrar a aplicação do método para o exemplo do CBERS são os cenários para o produto em 'Teste' e em 'Operação', e para a organização de 'Desenvolvimento do Projeto e Construção' e de 'Montagem'.

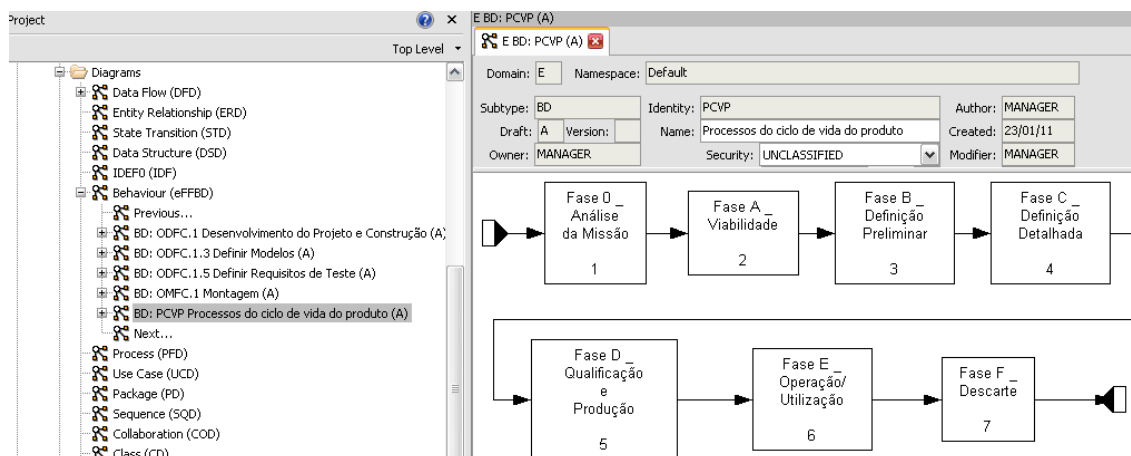


Figura 5.1 - Processos do ciclo de vida do produto.

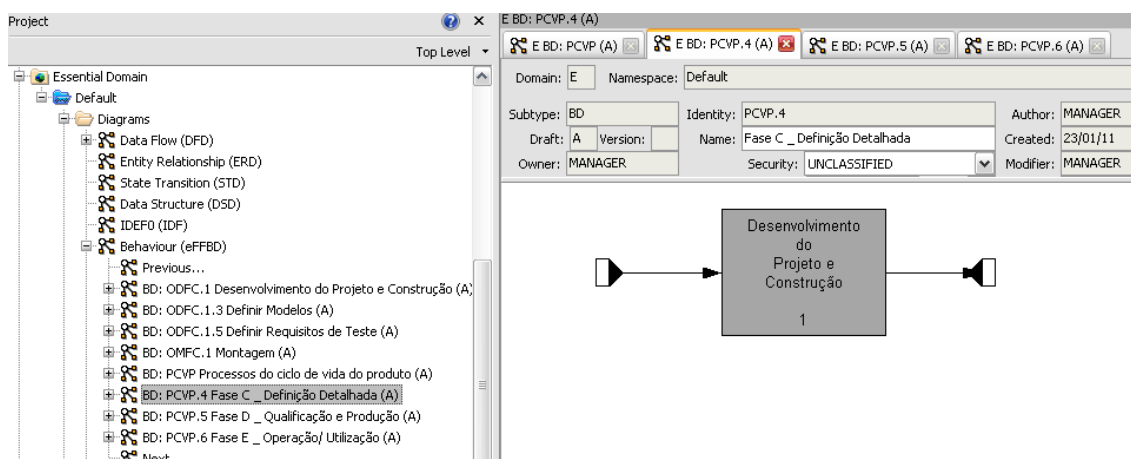


Figura 5.2 - Cenário de desenvolvimento de projeto e construção.

Figura 5.3 mostra os cenários desdobrados a partir do processo do ciclo de vida da 'Fase D Qualificação e Produção' ilustrado na Figura 5.1. O cenário de 'Montagem' será explorado para organização, e o cenário de 'Teste' para produto. Para o processo do ciclo de vida 'Fase E Operação/Utilização' é desdobrado um cenário de operação para produto, ilustrado na Figura 5.4.

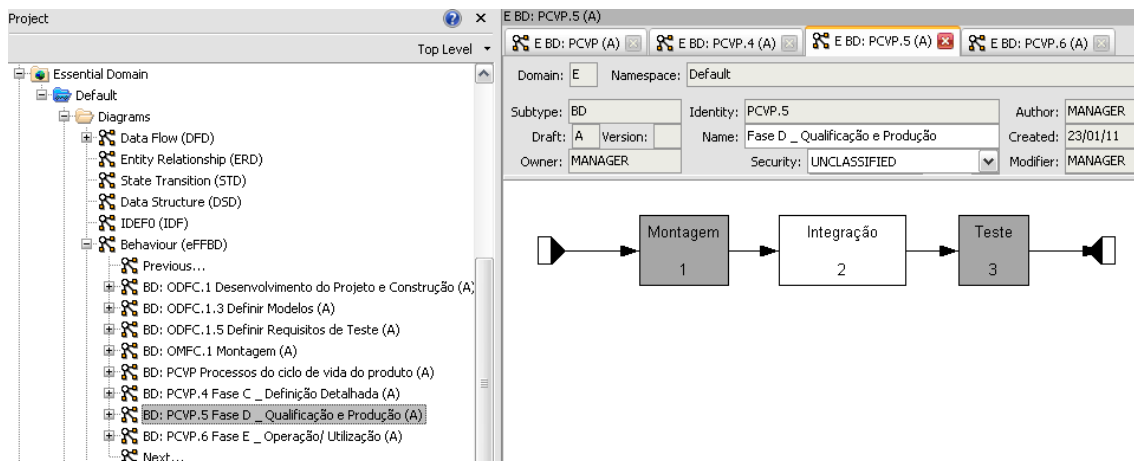


Figura 5.3 - Cenário de montagem e teste.

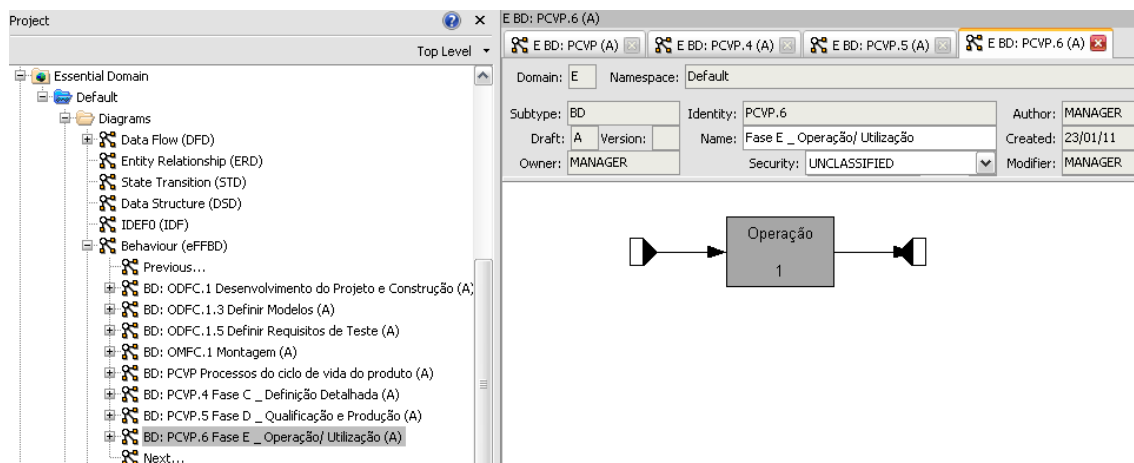


Figura 5.4 - Cenário de operação para produto.

A partir desses cenários são capturados os interesses dos *stakeholders* para produto e organização, explorado a seguir na Subseção 5.1.3.

Pelo diagrama de caso de uso, são ilustrados o processos que serão implementados pela organização. A Figura 5.5 ilustra o escopo do esforço de desenvolvimento da organização.

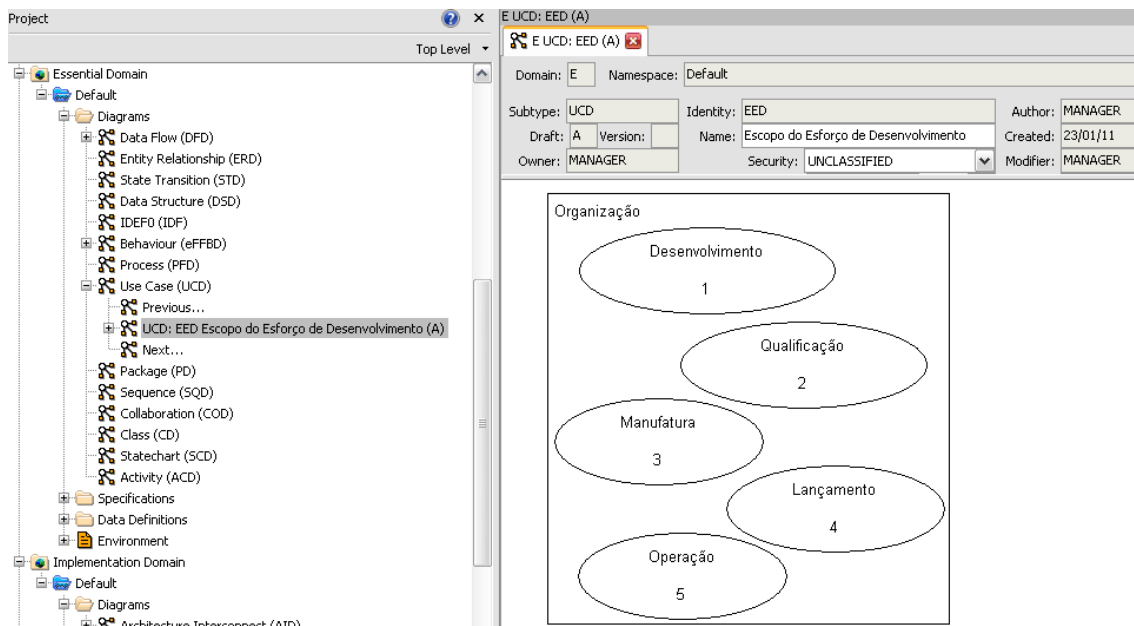


Figura 5.5 - Escopo do esforço de desenvolvimento.

Os cenários dos processos do ciclo de vida ilustrados nas figuras acima serão realizados para a análise de *stakeholder*, de onde serão derivados os requisitos de *stakeholders* a partir dos interesses dos *stakeholders*. Requisitos de *stakeholders* de organização são identificados por cenário do ciclo de vida, dentro do escopo do esforço de desenvolvimento.

5.1.3 Análise de Stakeholder e de Requisitos

Esta Subseção apresenta a análise de *stakeholders* e a análise de requisitos conforme descrita nas Subseções de 3.3.5 a 3.3.10, utilizando o *Cradle* para o auxílio computacional do processo como na Subseção 4.2.2.

As Figuras 5.6 a 5.9 ilustram a captura dos *stakeholders* e seus interesses para os cenários de produto e de organização usando o diagrama (DFD). Os interesses dos *stakeholders* são representados nas setas que saem de um

elemento que representa o *stakeholder*, e vai para a bolha que contém o nome do cenário do processo do ciclo de vida. É a partir do interesse do *stakeholder* que os requisitos de *stakeholders* são derivados. Usando os *links* que relacionam os itens de informação que representam os stakeholders, seus interesses, os requisitos de *stakeholder*, as medidas de efetividades e as funções, é possível manter rastreabilidade.

A Figura 5.6 ilustra os *stakeholders* e seus interesses para o cenário de teste. O diagrama auxilia a capturar o que mais tarde será descrito em forma de requisitos que impactarão o produto sob teste.

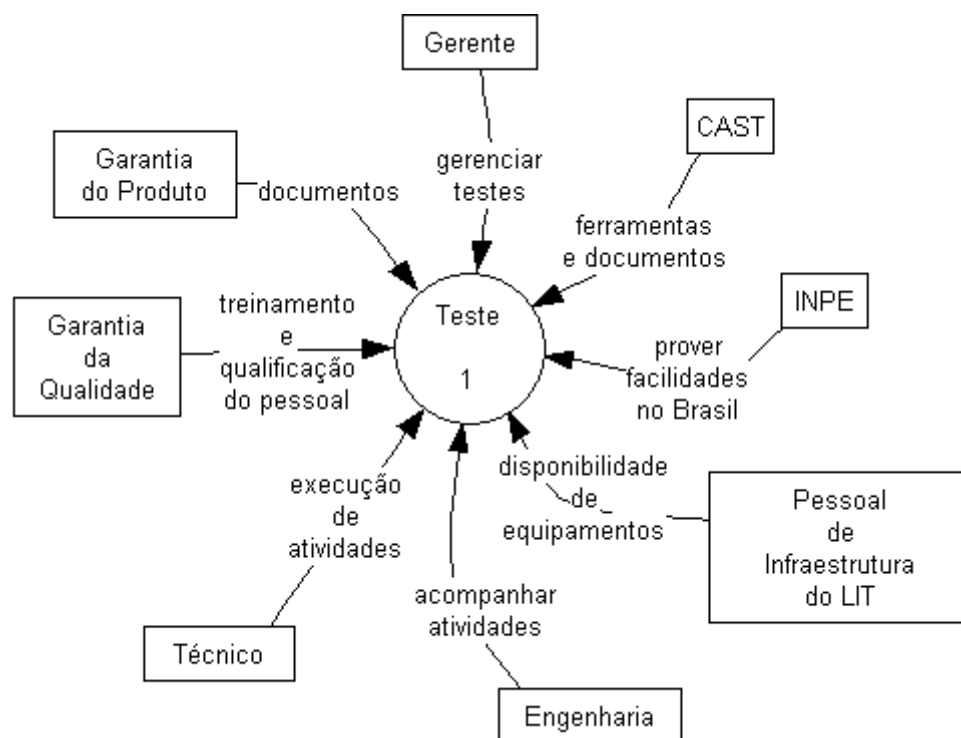


Figura 5.6 - *Stakeholders* de produto e seus interesses para o contexto de teste.

A Figura 5.7 ilustra os *stakeholders* e seus interesses para a operação do produto.

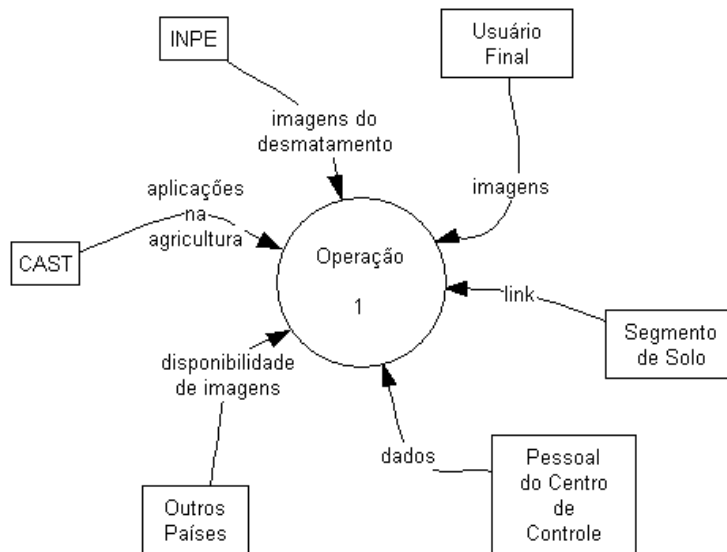


Figura 5.7 - Stakeholders de produto e seus interesses para o cenário de operação.

A Figura 5.8 ilustra os stakeholders para o cenário de desenvolvimento do projeto e construção. Aqui são levantados os recursos para a execução das atividades.

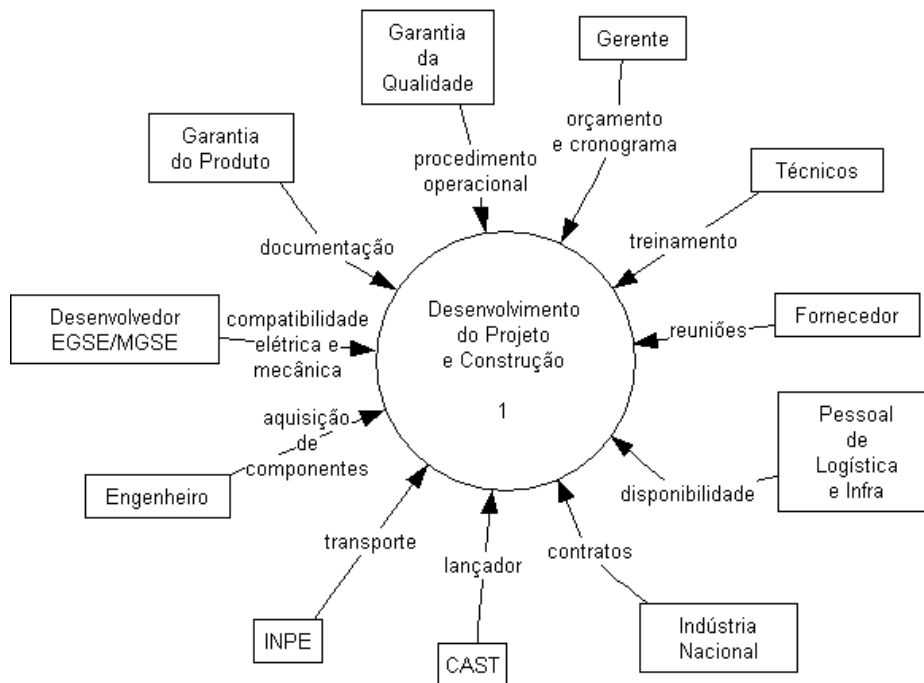


Figura 5.8 - Stakeholders de organização e seus interesses para o cenário de desenvolvimento do projeto e construção.

A Figura 5.9 ilustra os *stakeholders* e seus interesses no cenário de montagem do satélite. Aqui são representados os recursos que devem ser atendidos pela organização ao desempenhar o processo de montagem. Requisitos de organização impactam não só no produto, mas também nos recursos a serem empregados no processo de montagem.

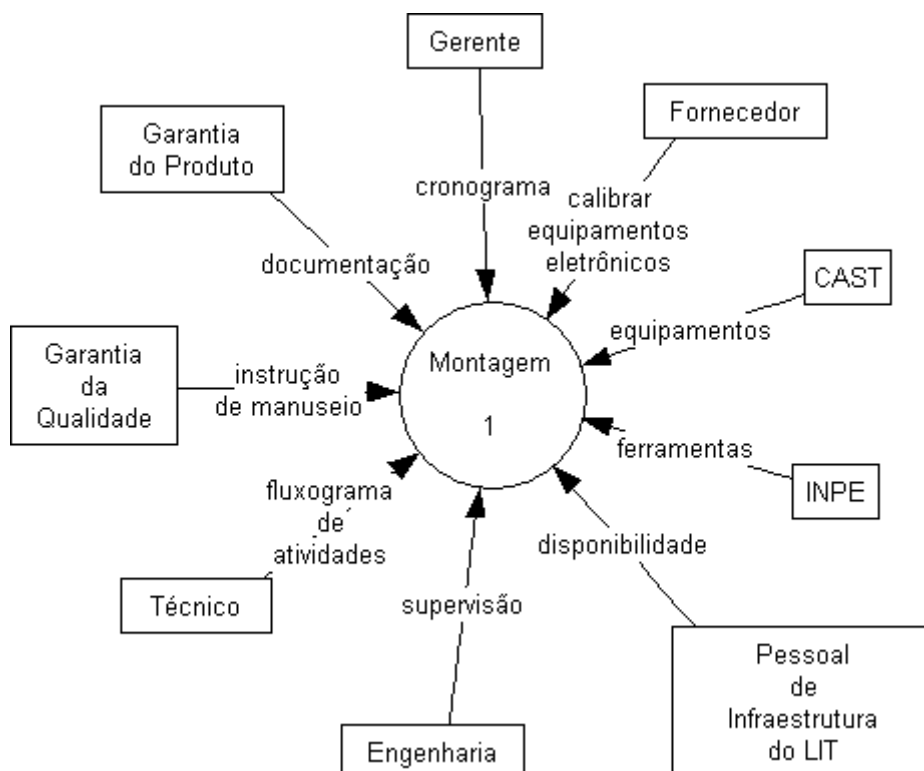


Figura 5.9 - *Stakeholders* de organização e seus interesses para o cenário de montagem.

Para gerar os *stakeholders* e associar informações, atributos, criar *links* e gerenciá-los, é necessário criar itens de informação para representar os *stakeholders* identificados nos diagramas. Assim é possível manter toda informação relacionada ao *stakeholder* em um formulário. A Figura 5.10 ilustra os itens de informação de *stakeholders* para representar os *stakeholders* identificados nos diagramas.

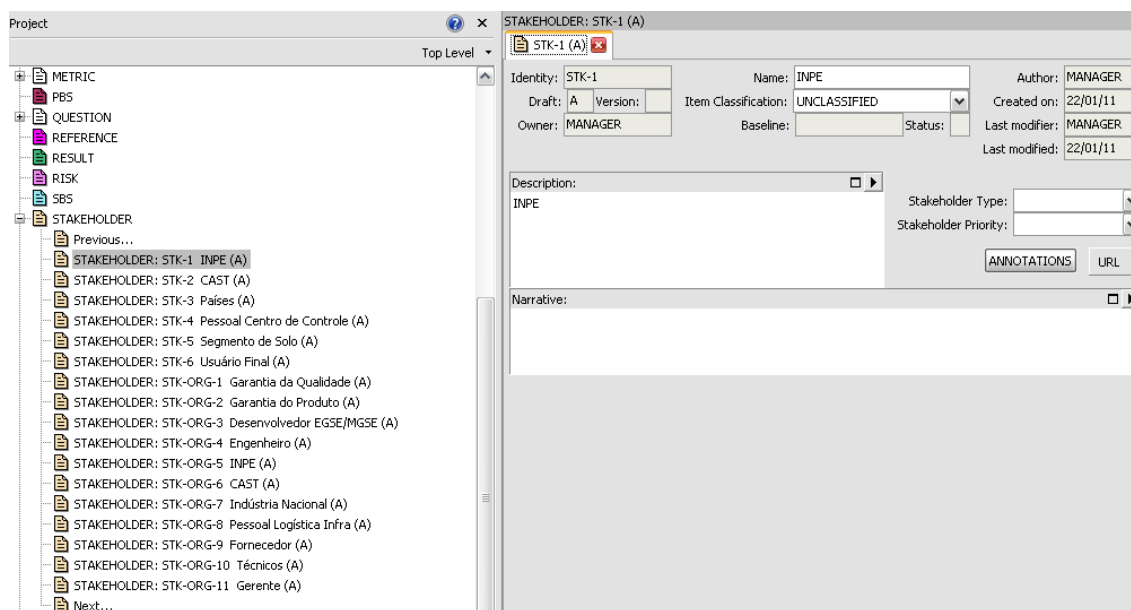


Figura 5.10 - Item de informação gerado para *stakeholder*.

Assim como é necessário criar itens de informação para *stakeholders*, seus interesses também são representados em forma de item de informação. A Figura 5.11 ilustra os itens de informação *concern* para representar os interesses de *stakeholders*.

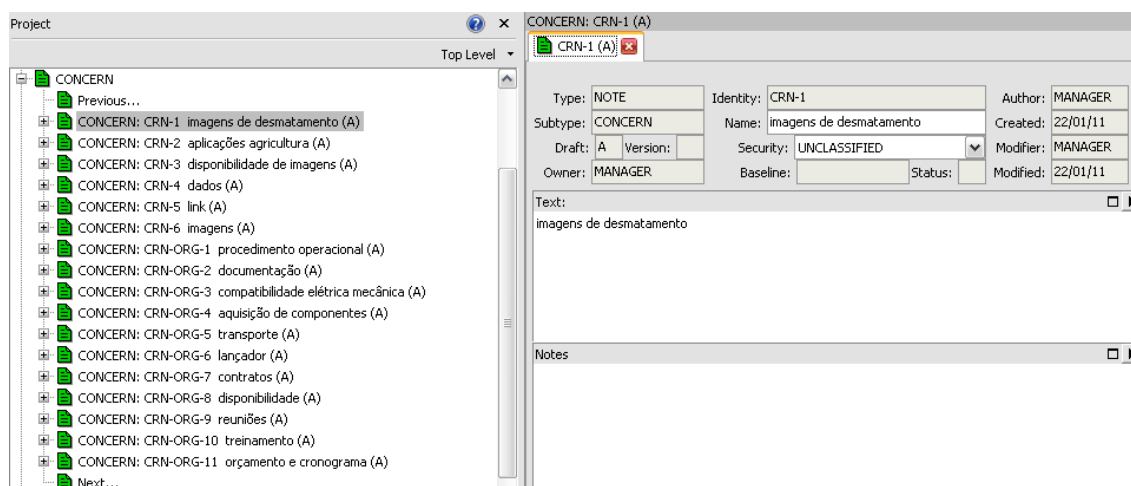


Figura 5.11 - Item de informação gerado para interesse de *stakeholder*.

O item de informação é um formulário que pode receber textos e atributos. Por exemplo, o nome de *login* do usuário do sistema que criou o item e a sua data de criação, como também o nome do usuário que alterou o item e a sua data de alteração, são atributos gerenciáveis do item de informação.

Ao criar o diagrama de stakeholders e seus interesses, as setas que conectam *stakeholders* e seus interesses no cenário do processo do ciclo de vida, geram itens do tipo fluxo (*flow*) no *Cradle*, onde se pode criar um *link* entre itens de informação por referência cruzada. Portanto estes itens de informação são associados pelos *links* aos interesses dos *stakeholders*. A Figura 5.12 ilustra o *link* entre o item de informação *concern* que ilustra o interesse do *stakeholder* ao fluxo representado pela seta da Figura 5.9, chamado *Data (Ess)*. Ou seja, o *stakeholder* declara seu interesse que está associado ao diagrama. O item de informação permite descrever informações a respeito do interesse do *stakeholder*.

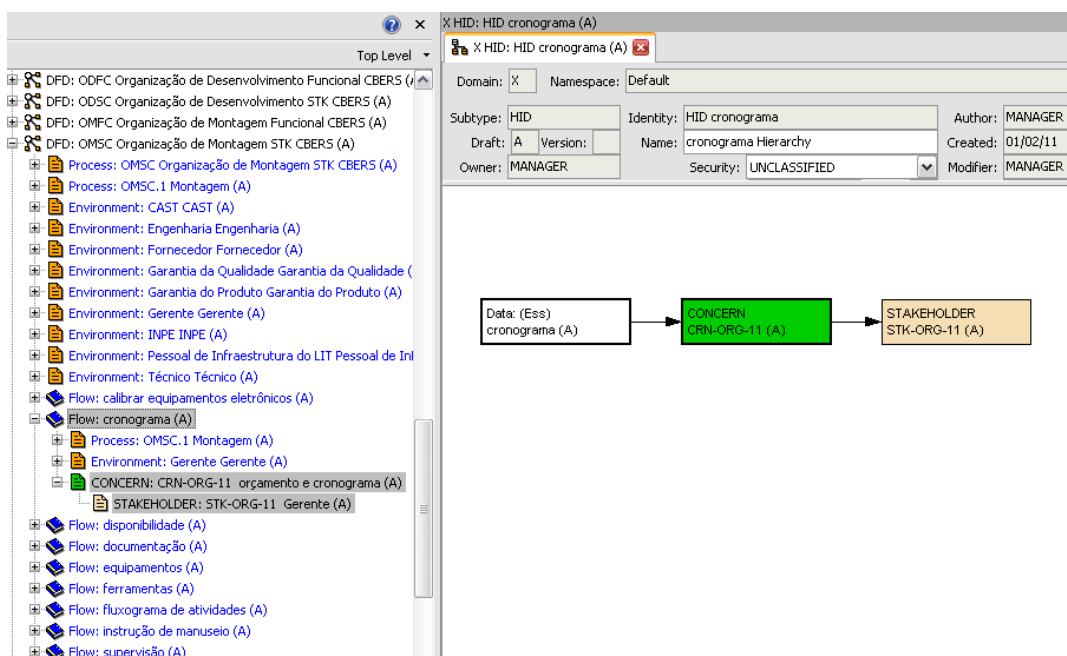


Figura 5.12 - *Link* entre elemento do diagrama e itens de informação.

A Figura 5.13 ilustra que o *flow* gerado pelo diagrama pode receber texto e também os atributos de gerenciamento.

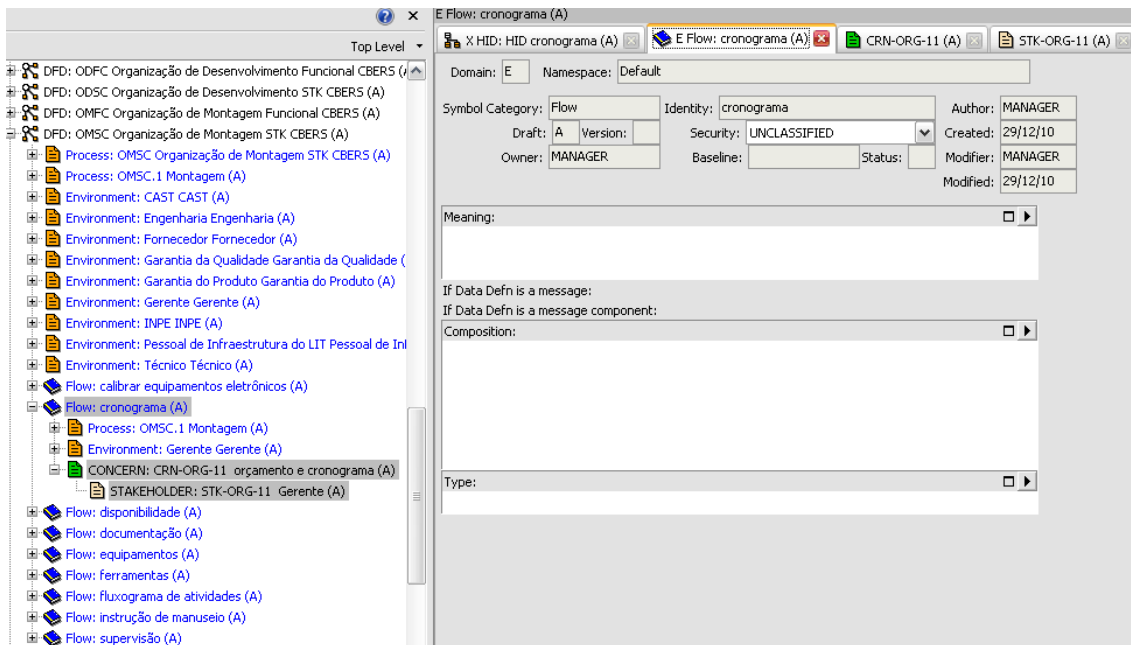


Figura 5.13 - Descrição de informação do fluxo.

Itens de informação são usados para capturar, as medidas de efetividade - *measures of effectiveness* (MoEs), que devem medir como o sistema atenderá os requisitos de *stakeholder*.

Para representar as MoEs elas são descritas no item de informação, que é associado ao interesse do *stakeholder*. Os itens de informação são associados por *links*. É possível manter a rastreabilidade e navegar por estes *links*.

Figura 5.14 ilustra o item de informação que representa as MoEs.

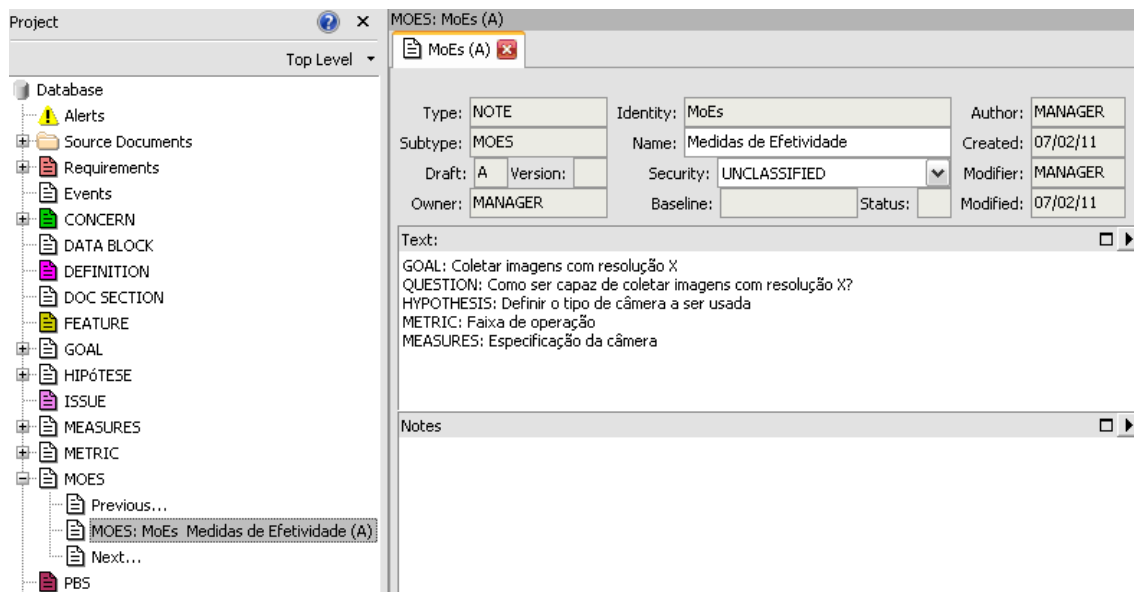


Figura 5.14 - Item de informação que representa as MoEs.

A Figura 5.15 representa a captura da declaração do interesse do *stakeholder* para o cenário do processo do ciclo de vida. Esse interesse é representado pelo item de informação chamado *concern*, que é associado ao item de informação que representa o *stakeholder*. O item de informação de *stakeholder* endereça o *stakeholder* capturado no diagrama que representa os interesses dos *stakeholders*, ilustrados nas Figuras 5.6 a 5.9 da análise de *stakeholder*. O interesse do *stakeholder* é traduzido para requisito de *stakeholder*. Do requisito de *stakeholder* são identificadas e traduzidas as MoEs.

Os *links* que associam os itens de informação permitem ser navegados e manter a rastreabilidade. A Figura 5.15 ilustra uma forma de visualizar e navegar pelos itens de informação associados por *links*.

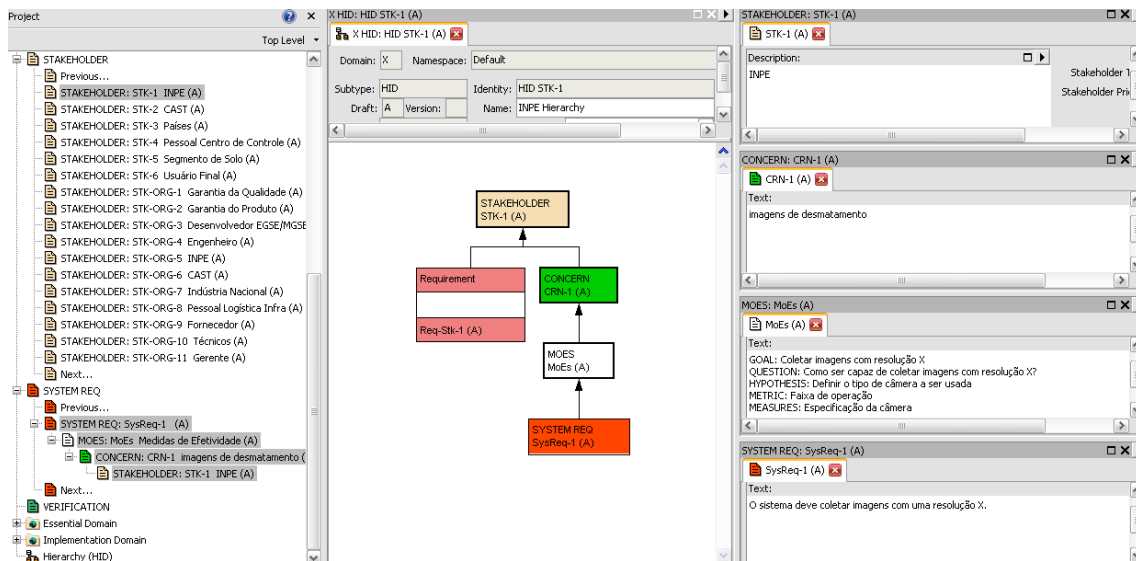


Figura 5.15 - Visualização dos itens de informação associados por *links*.

O *link* que associa os itens de informação permite receber informações e atributos, como a Figura 5.17 ilustra. O *link* pode ser criado para atender a um determinado tipo de associação de itens de informação. Por exemplo, um item de informação do tipo interesse pode ser associado ao item de informação do tipo *stakeholder* por um *link* chamado ‘declarado por’. Isso significa que o *stakeholder* declarou aquele interesse. Ou ainda, um *link* do tipo ‘verificado por’ significa que a associação entre esses elementos se dá por uma verificação.

Assim como o item de informação recebe atributos de gerenciamento e textos, os *links* que são gerados quando são associados itens de informação, também são uma forma de item de informação, que recebe atributos e textos que descrevem seu significado.

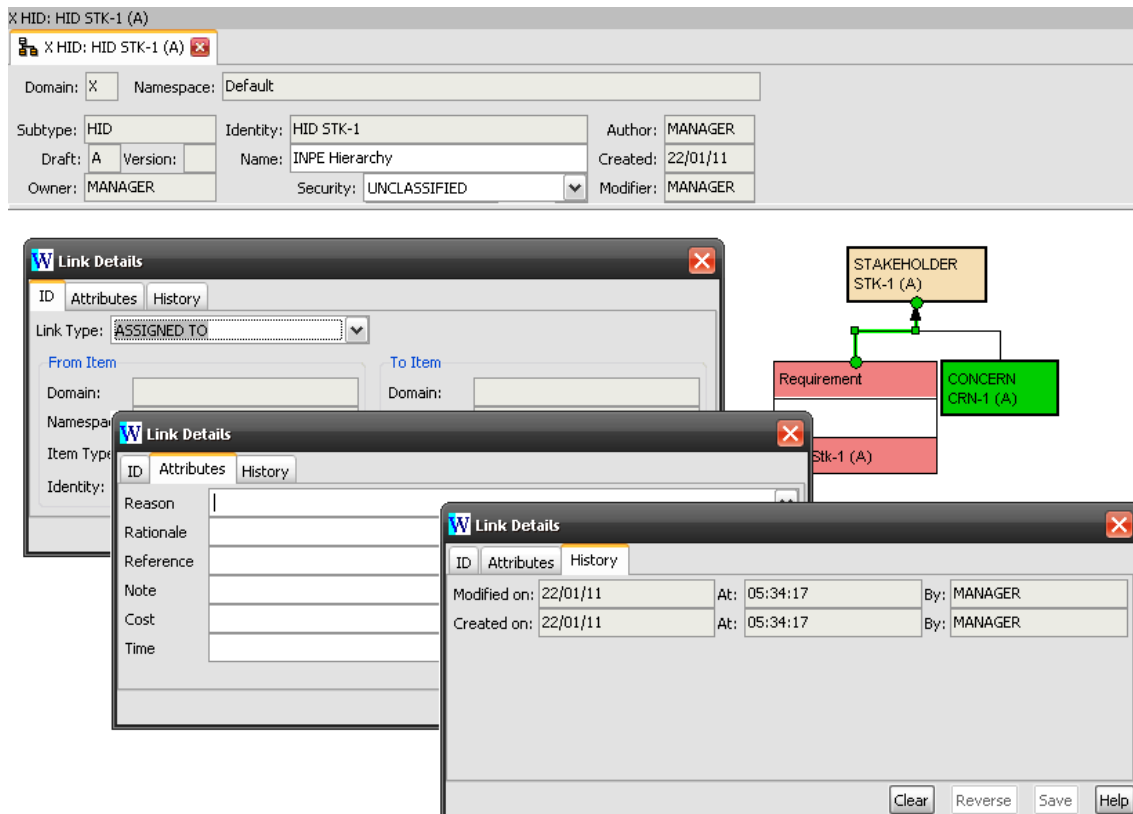


Figura 5.17 - Atributos dos *links* que associam os itens de informação.

A Figura 5.18 ilustra uma tabela contendo os itens de informação de *stakeholder* associados aos itens de informação dos interesses de *stakeholder*, e os requisitos de *stakeholder*. Isso permite outra maneira de representar os elementos da análise de *stakeholder*.

Project

Top Level

- METRIC
- PBS
- QUESTION
- REFERENCE
- RESULT
- RISK
- SBS
- STAKEHOLDER
 - Previous...
 - STAKEHOLDER: STK-1 INPE (A)
 - STAKEHOLDER: STK-2 CAST (A)
 - STAKEHOLDER: STK-3 Países (A)
 - STAKEHOLDER: STK-4 Pessoal Centro de Controle (A)
 - STAKEHOLDER: STK-5 Segmento de Solo (A)
 - STAKEHOLDER: STK-6 Usuário Final (A)
 - STAKEHOLDER: STK-ORG-1 Garantia da Qualidade (A)
 - STAKEHOLDER: STK-ORG-2 Garantia do Produto (A)
 - STAKEHOLDER: STK-ORG-3 Desenvolvedor EGSE/MGSE (A)
 - STAKEHOLDER: STK-ORG-4 Engenheiro (A)
 - STAKEHOLDER: STK-ORG-5 INPE (A)
 - STAKEHOLDER: STK-ORG-6 CAST (A)
 - STAKEHOLDER: STK-ORG-7 Indústria Nacional (A)
 - STAKEHOLDER: STK-ORG-8 Pessoal Logística Infra (A)
 - STAKEHOLDER: STK-ORG-9 Fornecedor (A)
 - STAKEHOLDER: STK-ORG-10 Técnicos (A)
 - STAKEHOLDER: STK-ORG-11 Gerente (A)
 - Next...
- SYSTEM REQ
- VERIFICATION
- Essential Domain
- Implementation Domain
- Hierarchy (HID)

Query: STAKEHOLDER - All

Identity	Ver	Dft	Name	All Linked Items			
				Identity	Name	Version	Draft
1		A	INPE	Req-Stk-1			A
				CRN-1	imagens de desmatamento		A
2		A	CAST	CRN-2	aplicações agricultura		A
3		A	Países	CRN-3	disponibilidade de imagens		A
4		A	Pessoal Centro de Controle	CRN-4	dados		A
5		A	Segmento de Solo	CRN-5	link		A
6		A	Usuário Final	CRN-6	imagens		A
7		A	Garantia da Qualidade	CRN-ORG-1	procedimento operacional		A
8		A	Garantia do Produto	CRN-ORG-2	documentação		A
9		A	Desenvolvedor EGSE/MGSE	CRN-ORG-3	compatibilidade elétrica mecânica		A
10		A	Engenheiro	CRN-ORG-4	aquisição de componentes		A
11		A	INPE	CRN-ORG-5	transporte		A
12		A	CAST	CRN-ORG-6	lançador		A
13		A	Indústria Nacional	CRN-ORG-7	contratos		A
14		A	Pessoal Logística Infra	CRN-ORG-8	disponibilidade		A
15		A	Fornecedor	CRN-ORG-9	reuniões		A
16		A	Técnicos	CRN-ORG-10	treinamento		A

Figura 5.18 - Itens de informação da análise de *stakeholders* listados em uma tabela.

A Figura 5.19 ilustra parte do documento gerado da tabela, contendo os itens de informação representados na Figura 5.19.

Exemplo de documento CBERS STK-REQ-CRN - Microsoft Word

Arquivo Editar Exibir Inserir Formatar Ferramentas Tabela Janela Ajuda Traduzir

100% Times New Roman 10

STAKEHOLDER - All

Identity	Ver	Dft	Name	All Linked Items			
				Identity	Name	Version	Draft
STK-1		A	INPE	Req-Stk-1			A
				CRN-1	imagens de desmatamento		A
STK-2		A	CAST	CRN-2	aplicações agricultura		A
STK-3		A	Países	CRN-3	disponibilidade de imagens		A
STK-4		A	Pessoal Centro de Controle	CRN-4	dados		A
STK-5		A	Segmento de Solo	CRN-5	link		A
STK-6		A	Usuário Final	CRN-6	imagens		A
STK-ORG-1		A	Garantia da Qualidade	CRN-ORG-1	procedimento operacional		A
STK-ORG-2		A	Garantia do Produto	CRN-ORG-2	documentação		A
STK-ORG-3		A	Desenvolvedor EGSE/MGSE	CRN-ORG-3	compatibilidade elétrica mecânica		A
STK-ORG-4		A	Engenheiro	CRN-ORG-4	aquisição de componentes		A
STK-ORG-5		A	INPE	CRN-ORG-5	transporte		A
STK-ORG-6		A	CAST	CRN-ORG-6	lançador		A
STK-ORG-7		A	Indústria Nacional	CRN-ORG-7	contratos		A
STK-ORG-8		A	Pessoal Logística Infra	CRN-ORG-8	disponibilidade		A
STK-ORG-9		A	Fornecedor	CRN-ORG-9	reuniões		A
STK-ORG-10		A	Técnicos	CRN-ORG-10	treinamento		A
STK-ORG-11		A	Gerente	CRN-ORG-11	orçamento e cronograma		A

Figura 5.19 - Documento gerado a partir da tabela.

Os exemplos apresentados até aqui, ilustram como a análise de *stakeholder* pode ser realizada, utilizando o *Cradle*.

5.1.4 Análise Funcional

Esta Subseção apresenta a análise funcional descrita nas Subseções de 3.3.11 a 3.3.18, utilizando o *Cradle* para o auxílio computacional do processo como na Subseção 4.2.3.

Os elementos do ambiente são capturados pelos terminadores do diagrama DFD. Os fluxos de material, energia e informação são capturados e descritos a partir das setas no diagrama. Os cenários são desdobrados para maiores níveis de detalhes, no qual atributos são derivados. Atributos de produto e organização derivados dos modelos na análise funcional são relacionados. Esse relacionamento de atributos no nível funcional permite descrever os elementos de produto e organização antes de sua implementação. Dessa forma é possível pensar não somente nos requisitos do produto, mas também os requisitos impostos pelas organizações que impactam no produto.

A partir dos fluxos é possível identificar esses atributos, para produto e organização. As Figuras 5.20 a 5.29 ilustram os cenários e o desdobramento desses cenários, para produto em 'Operação' e 'Teste', e para organização em 'Desenvolvimento do Projeto e Construção' e 'Montagem'. Os desdobramentos dos cenários ilustram com maiores detalhes as interações do produto e da organização com os elementos do ambiente. A partir dessas interações, surgem os atributos.

Por exemplo, a Figura 5.23 que é o desdobramento da Figura 5.22, ou seja, a captura do cenário que identifica os elementos que interagem com o produto sob teste, e troca energia, material e informação é desdobrada a um nível em

que é possível identificar os tipos de interação que o CBERS sofre. O modelo ilustra que o CBERS recebe uma determinada freqüência de vibração do equipamento de teste, e por essa interação é identificada a freqüência que o satélite pode receber para cumprir o requisito de vibração, e qual a capacidade de vibração que o equipamento de teste suporta. Dessa forma o atributo 'capacidade de vibração', impõe requisitos que impactam no equipamento que será utilizado quando o produto for passar pelo teste de vibração.

A Figura 5.20 ilustra os elementos do ambiente que trocam material, energia e informação com o produto durante a operação. Nas setas do diagrama são representados os tipos de fluxos entre os elementos do ambiente e o produto em operação.

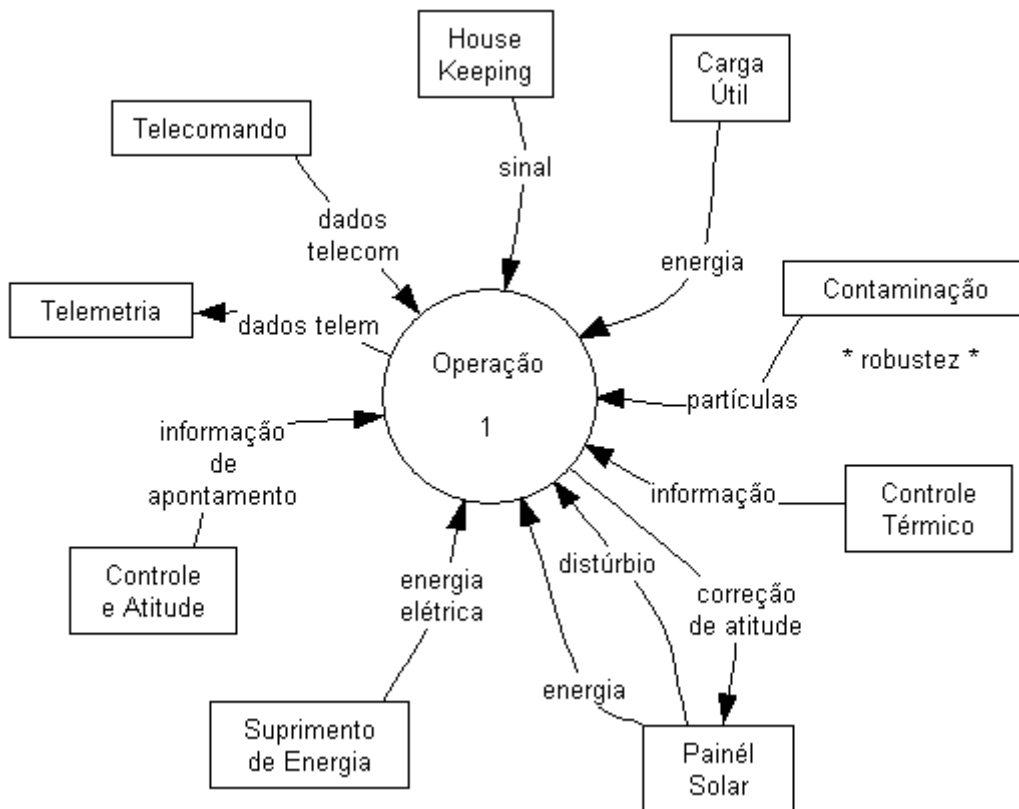


Figura 5.20 - Contexto funcional de produto para o cenário de operação.

A Figura 5.21 ilustra as funções que devem ser desempenhadas pelo produto, para realizar o desacoplamento da coifa até a sua estabilização e apontamento para a terra.

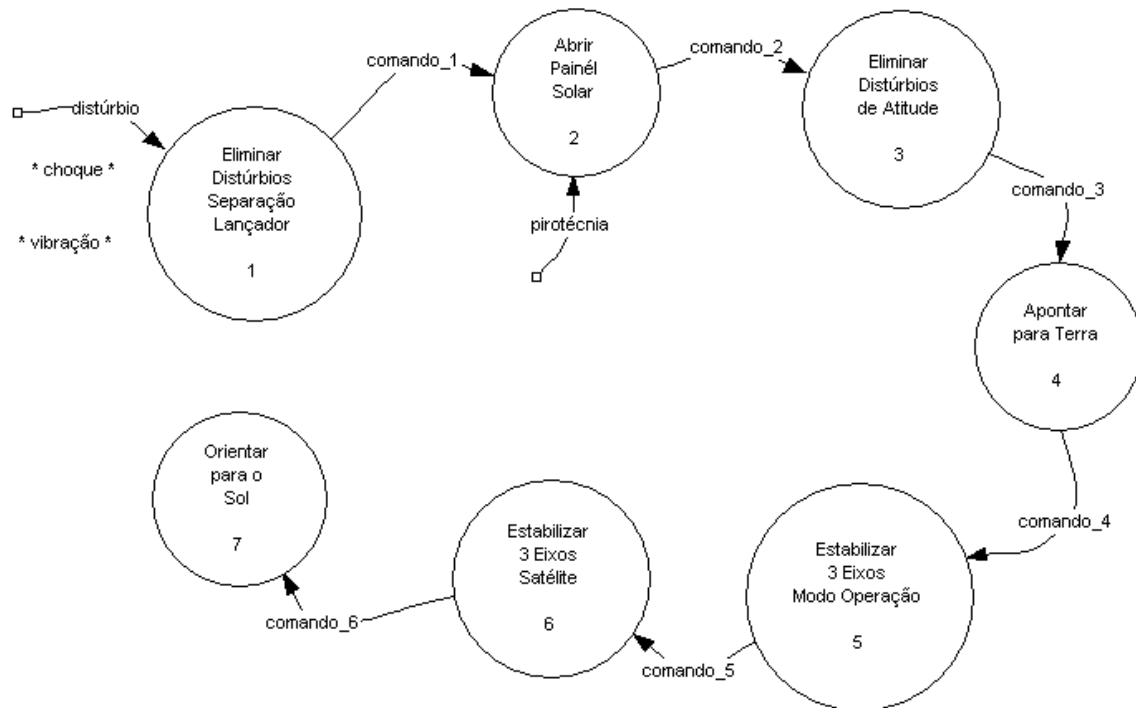


Figura 5.21 - Desdobramento do contexto funcional de produto para o cenário de operação.

A Figura 5.22 ilustra os elementos que interagem com o produto durante o processo de teste ambiental.

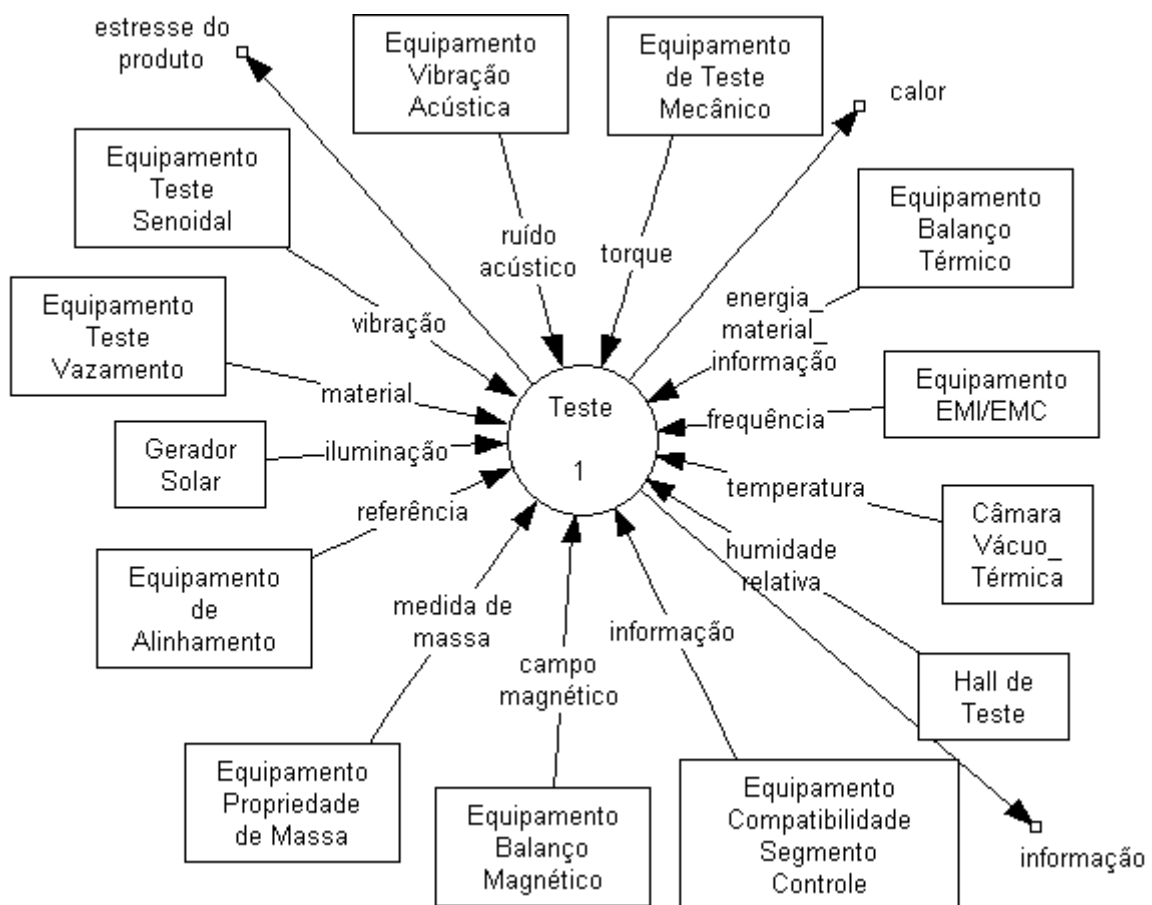


Figura 5.22 - Contexto funcional de produto para o cenário de teste.

A Figura 5.23 ilustra as interações do produto sob o teste de vibração. A partir dessas interações atributos são derivados.

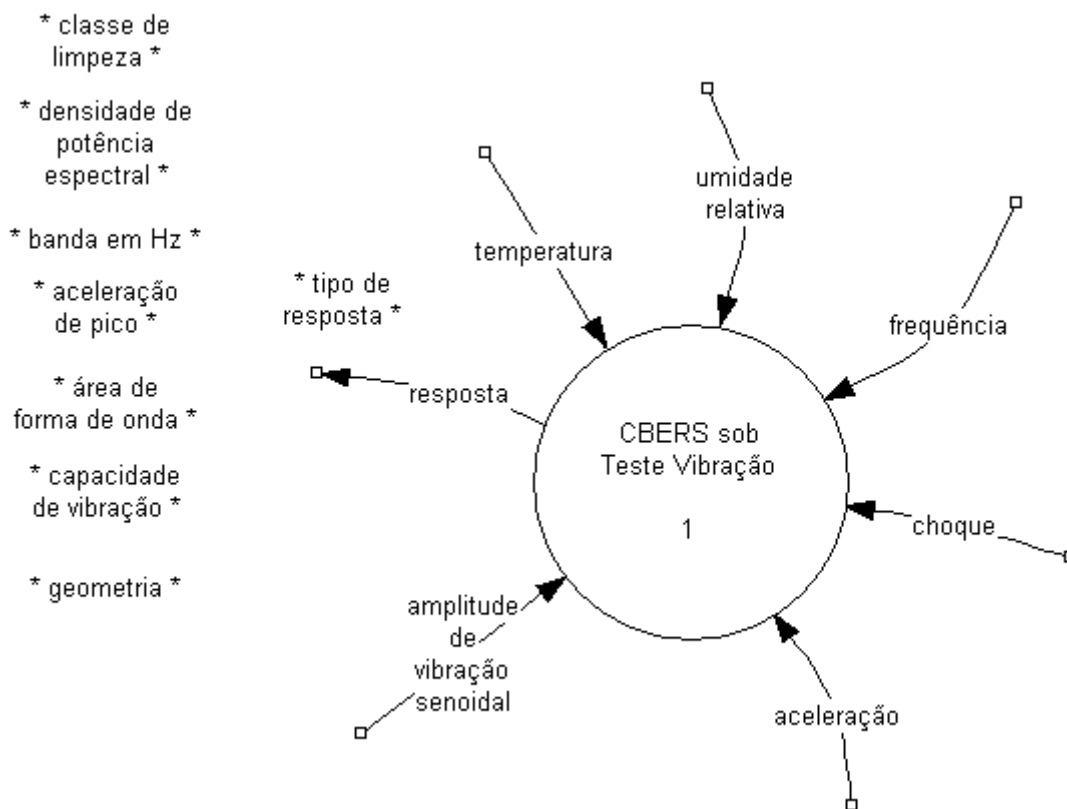


Figura 5.23 - Desdobramento do contexto funcional do produto para o cenário de teste.

Os modelos de organização são desdobrados em níveis que permitem identificar maiores detalhes, ilustrados pelos diagramas de comportamento. Cada elemento do diagrama de comportamento significa uma função no *Cradle*. Cada função representa uma atividade que será realizada no decorrer do tempo. Atributos são identificados para cada atividade a ser realizada pela organização. Por exemplo, atributos funcionais como ‘custo’ e ‘tempo’ descrevem o início de uma atividade e o seu tempo de execução. Isto impõe requisitos à organização de cronograma e de recursos que serão implementados para a realização de cada atividade.

A Figura 5.24 ilustra a captura da interação dos elementos que constituem a organização de desenvolvimento do projeto e construção. A execução das atividades é ilustrada na Figura 5.25.

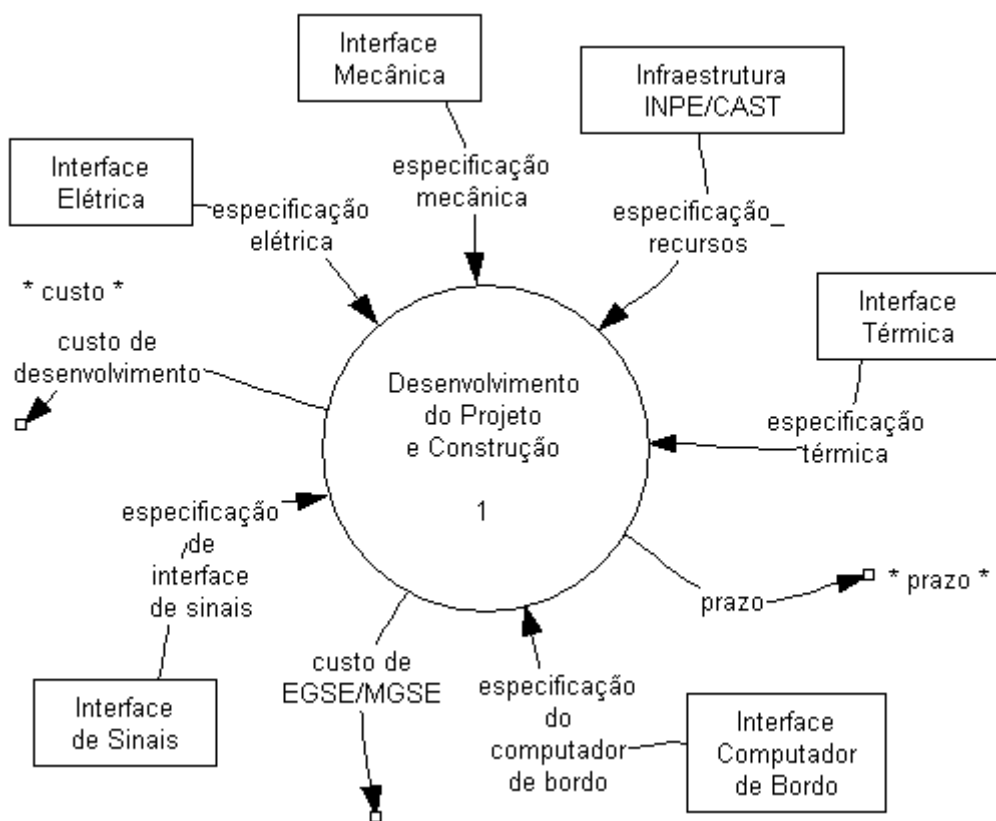


Figura 5.24 - Contexto funcional de organização para o contexto de desenvolvimento do projeto e construção.

A Figura 5.25 é ilustrada por um diagrama eFFBD para representar o desdobramento das atividades para desempenhar o desenvolvimento do projeto e construção.

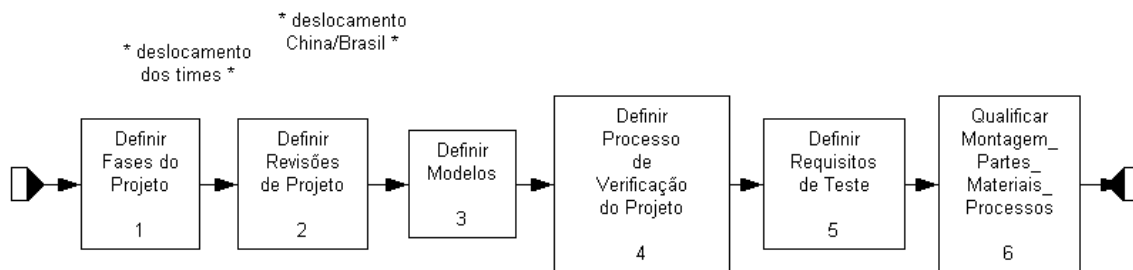


Figura 5.25 - Desdobramento do contexto funcional de organização para o contexto de desenvolvimento do projeto e construção.

Atributos que impactam no tempo de desenvolvimento do modelo do sistema e no custo são identificados na Figura 5.26 que é desdobrado da atividade de definir modelo, ilustrado na Figura 5.25.

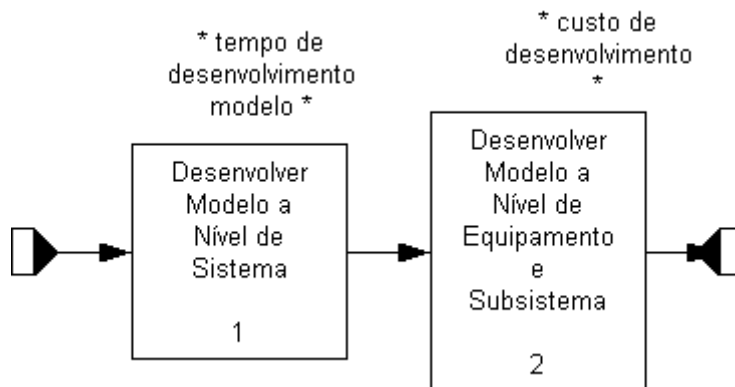


Figura 5.26 - Desdobramento de 'definir modelos'.

A Figura 5.27 ilustra custo e tempo como atributos funcionais que impactam na organização de desenvolvimento para testar equipamentos. Estas atividades são desdobradas da atividade de definir os requisitos de teste.

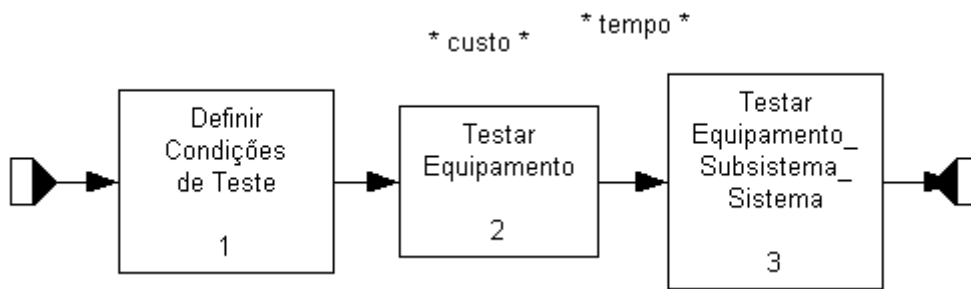


Figura 5.27 - Desdobramento de 'definir requisitos de teste'.

Atributo como 'disponibilidade de ferramentas' sugere que a organização deve pensar que durante o processo de montagem do satélite, não se deve faltar as ferramentas necessárias para a montagem, pois a falta de ferramenta impactaria no tempo de montagem do satélite.

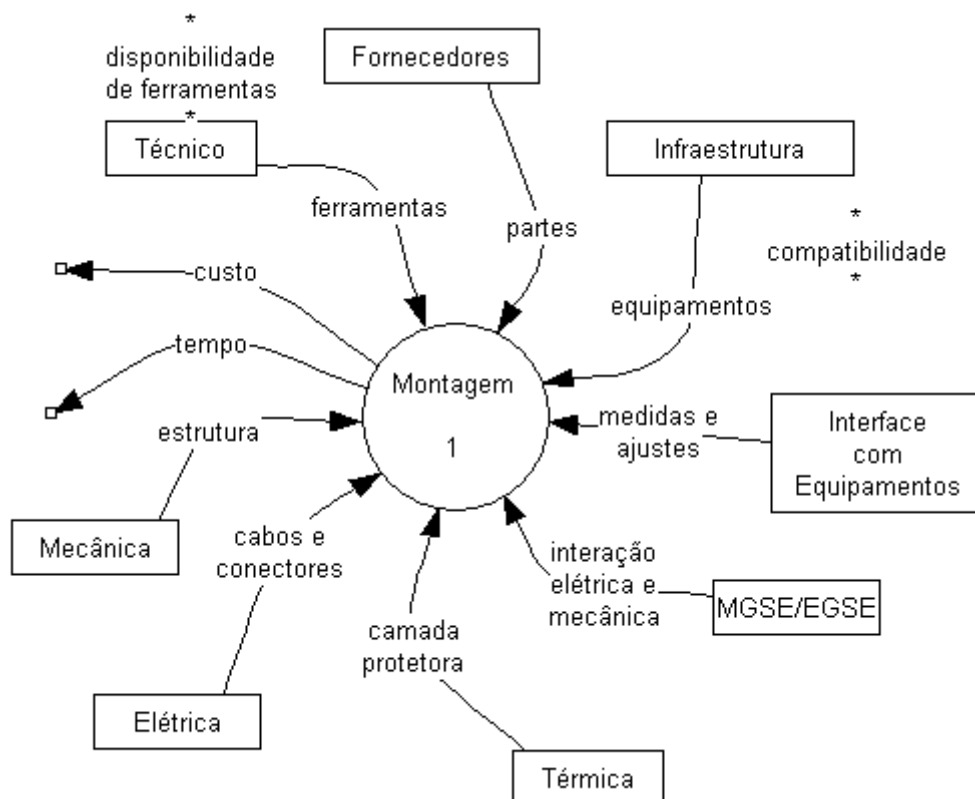


Figura 5.28 - Contexto funcional de organização para o cenário de montagem.

A Figura 5.29 ilustra as atividades para o cenário de montagem. Atributos são identificados a partir da descrição das funções que devem ser desempenhadas para a realização da montagem.

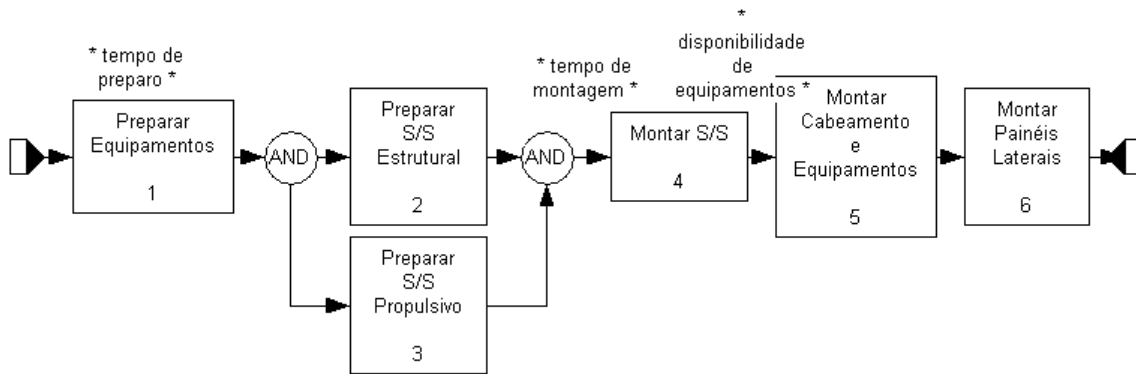


Figura 5.29 - Desdobramento do contexto funcional de organização para o cenário de montagem.

A Figura 5.30 ilustra o desdobramento do cenário de operação. A partir da referência cruzada é possível associar um item de informação de risco a cada elemento do diagrama. Desta maneira é feita a análise de risco sobre o tipo de função que, neste caso, o produto irá desempenhar durante um cenário de operação.

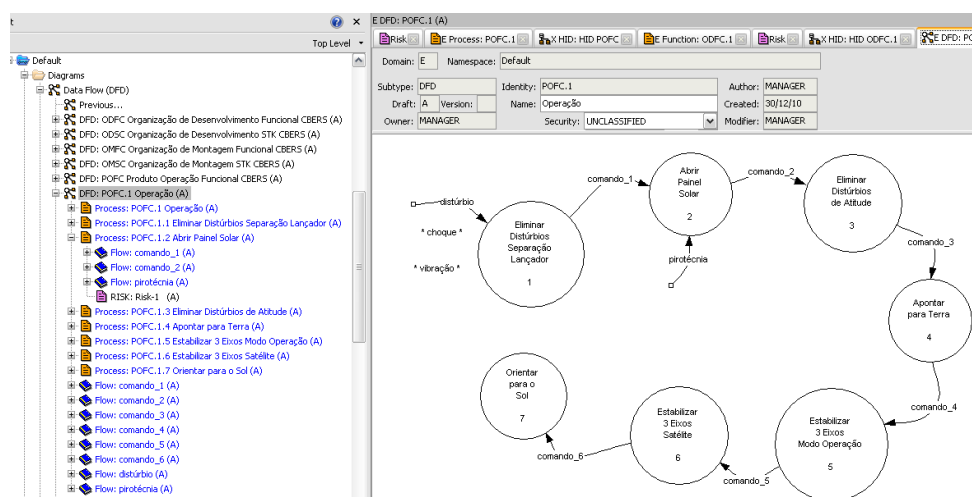


Figura 5.30 - Funções que devem ser desempenhadas pelo produto em operação.

A Figura 5.31 ilustra os campos a serem preenchidos com informação. Neste caso é descrita uma nota que o produto em operação irá desempenhar a função de 'abrir painel solar'.

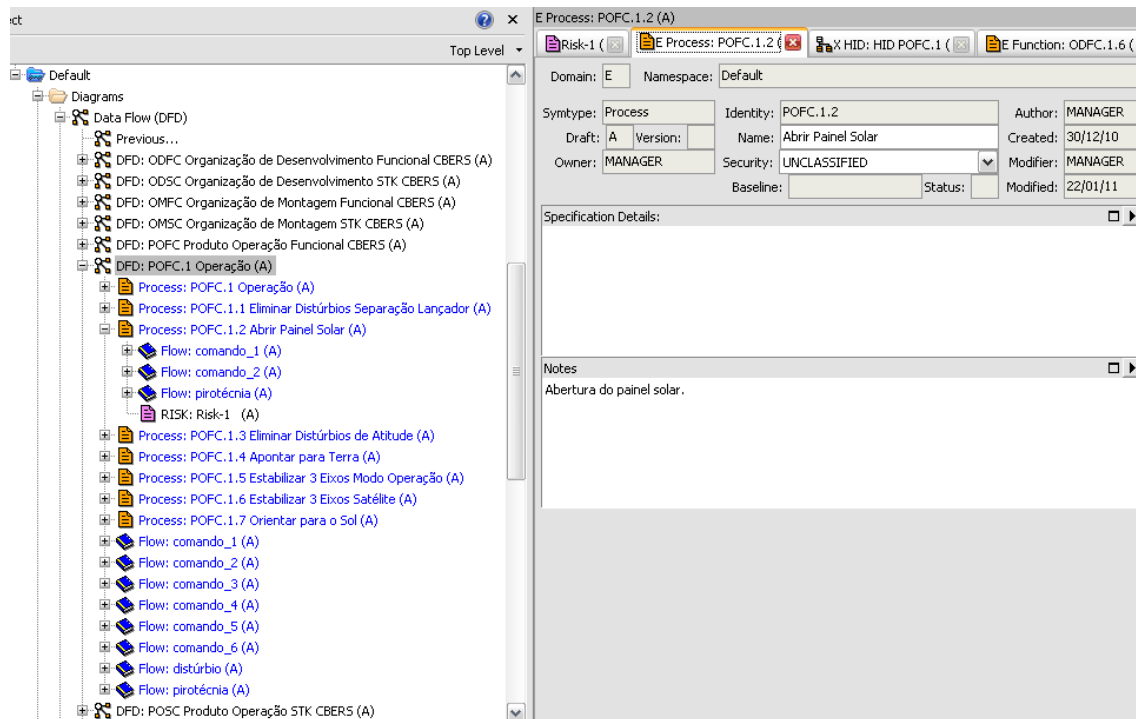


Figura 5.31 - Campo de informação da função de produto.

É para esta função de abertura do painel que um item de informação de risco é associado por referência cruzada. A Figura 5.32 ilustra a maneira de descrever a causa, o perigo, a falha e a consequência.

Atributos como probabilidade, impacto, prioridade e mitigação podem ser gerenciados no item de informação.

The image shows a software interface for managing risk information. At the top, there are four window tabs: 'Risk-1 (A)', 'X HID: HID POFC.1 (A)', 'E Function: ODFC.1.6 (A)', and 'Risk-2 (A)'. The main form is divided into several sections:

- Header Section:** Contains fields for 'Name', 'Identity' (set to 'Risk-1'), 'Draft' (set to 'A'), 'Owner' (set to 'MANAGER'), 'Risk Type', 'Status', 'Classification' (set to 'UNCLASSIFIED'), 'Author' (set to 'MANAGER'), 'Created on' (set to '22/01/11'), 'Last modifier' (set to 'MANAGER'), and 'Last modified' (set to '22/01/11').
- Risk Details Section:** A text area containing the following information in red text:
 - Causa: Não separar
 - Perigo: Não carregar as baterias
 - Falha: Não abrir o painel
 - Consequência: Morte do sistema
- Metadata Section:** Includes 'Date Raised', 'Review Date', and an 'ANNOTATIONS' button.
- Bottom Section:** Contains two expandable panels: 'Notes' and 'Mitigation Plan'.
- Additional Fields:** On the right side, there are dropdown menus for 'Responsibility', 'Risk Probability' (set to 'LOW'), 'Risk Impact' (set to 'HIGH'), 'Risk Priority' (set to 'CRITICAL'), and 'Mitigation' (set to 'ACCEPT').

Figura 5.32 - Item de informação de risco para produto.

O item de informação de risco fica associado ao elemento que representa a função do sistema que o produto deve desempenhar naquele cenário. Um exemplo de item de informação de risco associado à função é ilustrado na Figura 5.33.

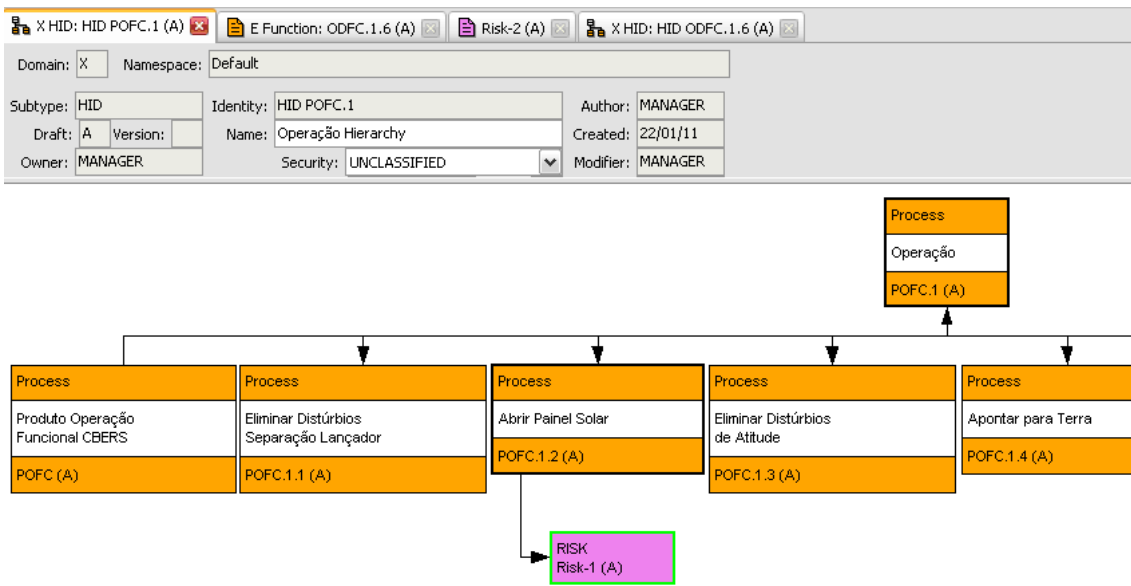


Figura 5.33 - Item de informação de risco associado à função 'abrir painel solar'.

A Figura 5.34 ilustra a descrição de uma função de organização.

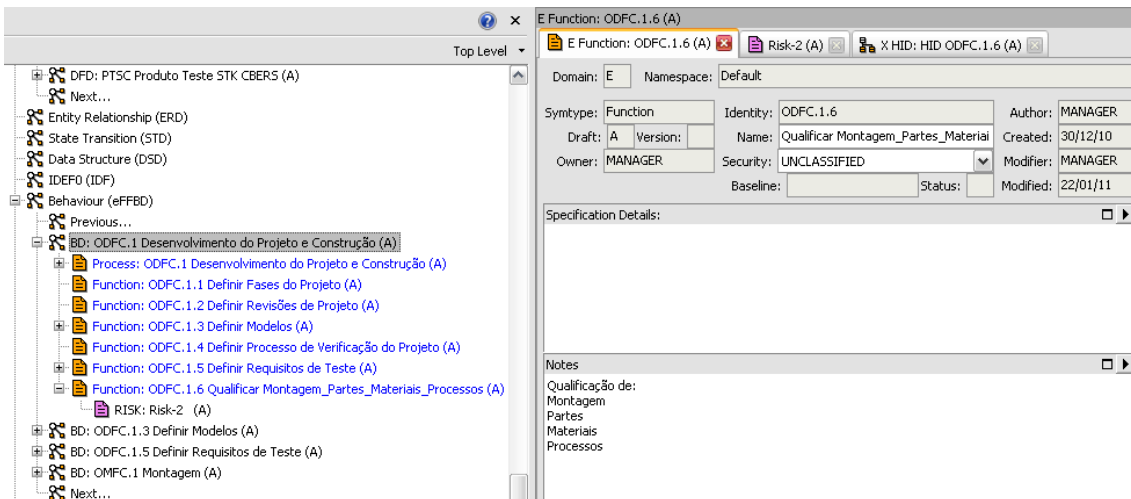


Figura 5.34 - Campos de informação da função de organização.

A análise de risco também é desempenhada para organização. A Figura 5.34 ilustra o item de informação de risco que será associado à função da organização.

The screenshot shows a software window titled 'Risk-2 (A)' with a sub-window 'X HID: HID ODFC.1.6 (A)'. The main form contains the following fields and sections:

- Name:** [Empty text box]
- Identity:** Risk-2
- Risk Type:** [Dropdown menu]
- Author:** MANAGER
- Draft:** A
- Version:** [Empty text box]
- Status:** [Dropdown menu]
- Created on:** 22/01/11
- Owner:** MANAGER
- Classification:** UNCLASSIFIED
- Last modifier:** MANAGER
- Baseline:** [Empty text box]
- Status:** [Empty text box]
- Last modified:** 22/01/11

Risk Details: [Expandable section]

- Causa:** Falta de recursos
- Perigo:** Equipamentos não funcionarem no ambiente espacial
- Falha:** Não funcionamento dos equipamentos
- Consequência:** Morte do sistema

Responsibility: [Dropdown menu]

Risk Probability: LOW

Risk Impact: HIGH

Risk Priority: CRITICAL

Mitigation: ACCEPT

Date Raised: [Dropdown menu]

Review Date: [Dropdown menu]

ANNOTATIONS [Button]

Notes: [Expandable section]

Mitigation Plan: [Expandable section]

Figura 5.35 - Item de informação de risco para organização.

A Figura 5.36 ilustra um exemplo do item de informação de risco associado a função da organização.

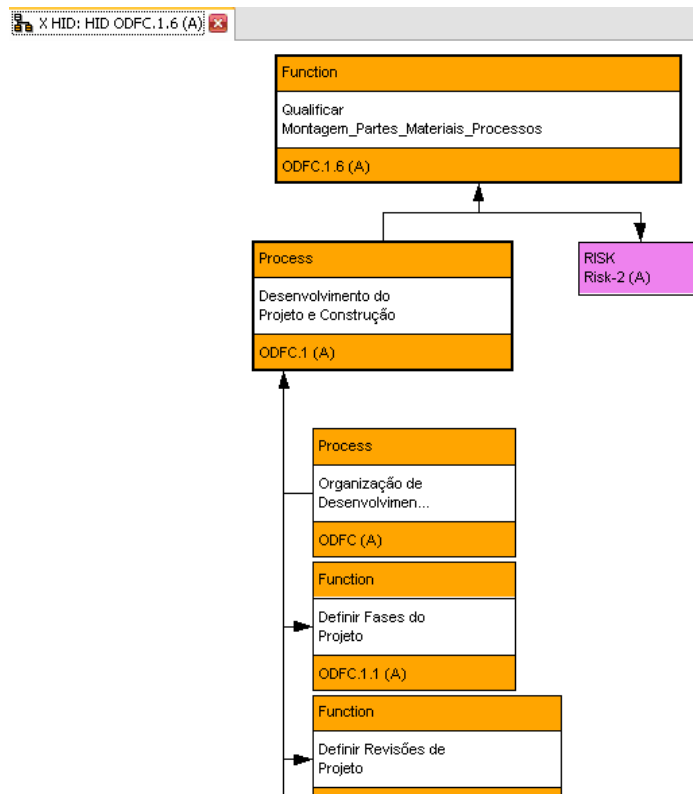


Figura 5.36 - Item de informação de risco associado à função da organização.

5.1.5 Análise de Implementação

Esta Subseção apresenta a análise de implementação descrita nas Subseções 3.3.19 a 3.3.27, utilizando o *Cradle* para o auxílio computacional do processo como na Subseção 4.2.4.

Aqui são usados diagramas do tipo PAD para as Figuras 5.37 a 5.39 e AID para a Figura 5.40 para ilustrar a captura dos elementos internos físicos, das interfaces físicas e dos fluxos de arquitetura para produto e organização. Atributos como geometria, diâmetro e peso são identificados a partir dos modelos.

A Figura 5.37 ilustra a captura dos instrumentos que compõem a carga útil. As interfaces físicas aqui representam quantidades de parafusos para fixar cada instrumento, a geometria do instrumento, a massa do instrumento. Cada instrumento possui uma característica física, que impacta no desenvolvimento do modelo da estrutura mecânica da carga útil.

E a estrutura da carga útil deve atender aos requisitos de vibração, requisitos de apontamento da câmera, e requisitos de taxa de transmissão dos barramentos de dados.

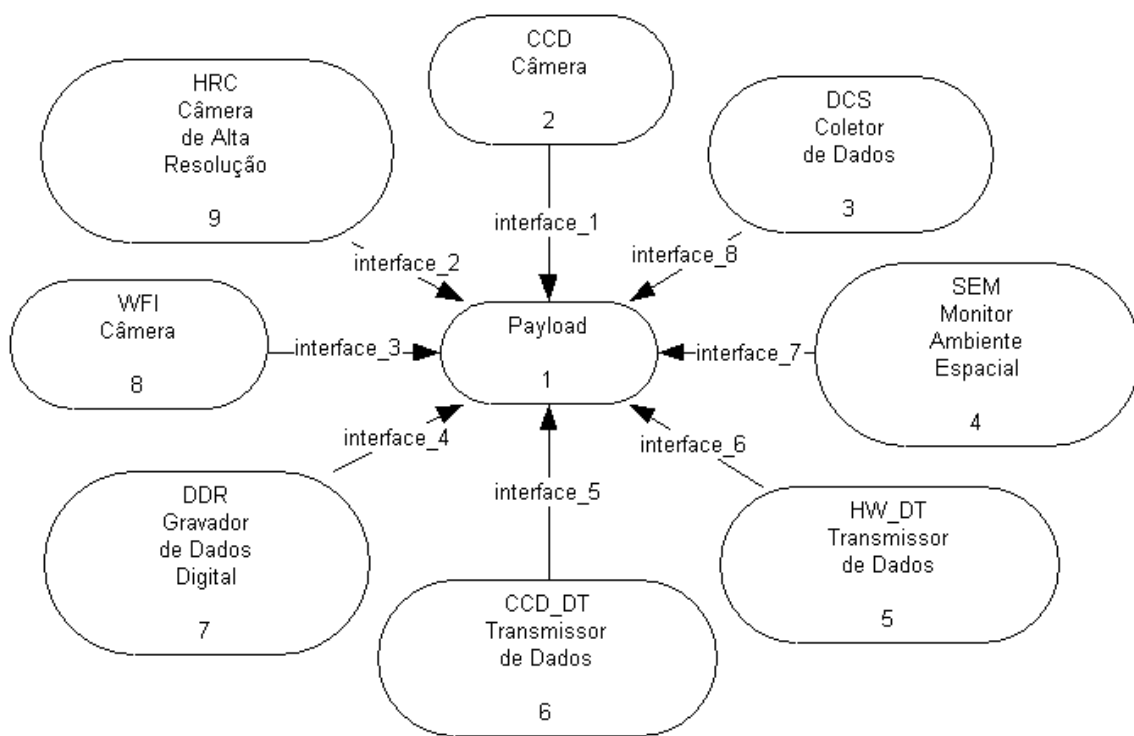


Figura 5.37 - Arquitetura de implementação de produto.

A Figura 5.38 ressalta que o satélite possui um atributo conforme a sua geometria, seu diâmetro e seu peso. O instrumento ao qual o satélite será submetido a teste também possui características, que se não forem compatíveis o produto terá que passar por modificações para passar a atender às características do equipamento. Portanto a interface entre satélite e

equipamento de teste deve ser identificada de forma a atender as especificações do equipamento.

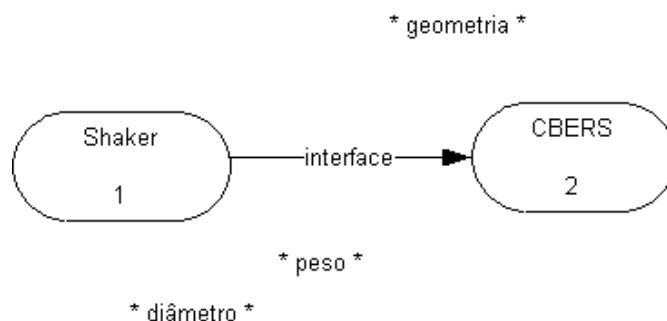


Figura 5.38 - Arquitetura de implementação de produto e interface.

A Figura 5.39 ilustra as interfaces entre equipamentos e produto. Para a execução dos testes são identificadas as interfaces elétricas e mecânicas dos equipamentos. Aqui a interface do laboratório representa a infraestrutura para manter os equipamentos que não podem ser alterados do lugar instalado ou modificados, caso não atendam a requisitos do produto. Estes são requisitos que impactam os recursos de implementação das atividades que solicitam os equipamentos do laboratório.

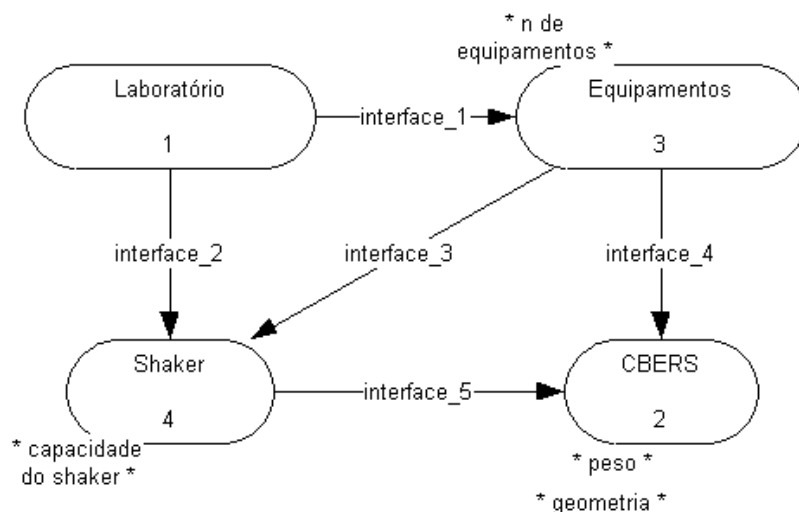


Figura 5.39 - Arquitetura de implementação de organização e interfaces.

A Figura 5.40 ilustra os equipamentos de testes e aponta para o desimpedimento dos *halls* para o deslocamento do satélite em um MGSE - *Mechanical Ground Support Equipment*. Para que a manobra de um satélite de algumas toneladas possa ser conduzida de maneira a não empenhar um esforço desnecessário, e o deslocamento da mão de obra também não seja desnecessário. Isso impacta em tempo, cronograma e custo.

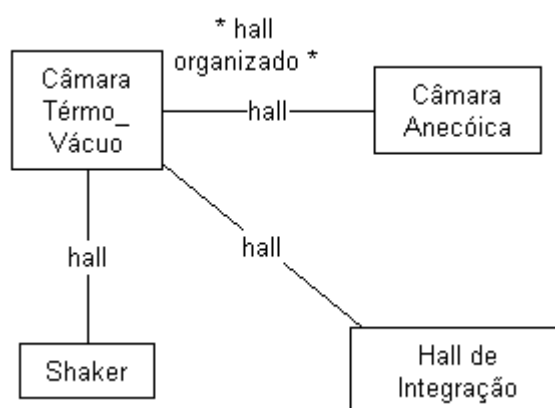


Figura 5.40 - Arquitetura de implementação de organização.

Os modelos gerados aqui para a análise de implementação de produto e organização, auxiliaram a identificar atributos que impactam na solução do produto e da organização. Essa solução deve ser balanceada levando em consideração os aspectos do projeto como por exemplo: se algum equipamento deve ser preparado pela organização, que realizará um teste sobre um produto fornecido pela indústria, só podem ser realizados os testes se os requisitos e atributos entre eles forem compatíveis.

5.1.6 Atributos

Esta Subseção apresenta os atributos derivados dos modelos funcionais, e dos modelos de implementação, conforme descritos na Subseção 3.3.28.

Os atributos derivados e capturados dos modelos da análise funcional e da análise de implementação, para produto e organização e o relacionamento entre eles são ilustrados na Tabela 5.1.

Tabela 5.1 - Atributos de produto e organização.

	Produto	Organização
Funcional x Funcional	<ul style="list-style-type: none"> - robustez - classe de limpeza - densidade de potência espectral - banda em Hz - aceleração de pico - área de forma de onda - capacidade de vibração - geometria - tipo de resposta - choque - vibração 	<ul style="list-style-type: none"> - custo - prazo - disponibilidade de ferramentas - compatibilidade - deslocamento times - deslocamento China/Brasil - tempo de desenvolvimento de modelo - custo de desenvolvimento - custo - tempo - tempo de preparo - tempo de montagem - disponibilidade de equipamentos
Funcional x Implementação	<ul style="list-style-type: none"> - capacidade de vibração - tipo de resposta - choque - peso 	<ul style="list-style-type: none"> - capacidade do shaker
Implementação x Funcional	<ul style="list-style-type: none"> - geometria - peso - diâmetro 	<ul style="list-style-type: none"> - tempo de desenvolvimento do modelo - custo de desenvolvimento - tempo de preparo - tempo de montagem
Implementação x Implementação	<ul style="list-style-type: none"> - geometria - peso - diâmetro 	<ul style="list-style-type: none"> - capacidade do shaker - n de equipamentos - peso - geometria - hall organizado

Os atributos podem ser gerenciados no *Cradle* pelos itens de informação, podendo ser cada atributo um item de informação. Esses atributos podem ser relacionados mostrando as associações entre eles, em uma matriz de relacionamento conforme Figura 5.41.

	ATB-PROD-1	ATB-PROD-2	ATB-PROD-3	ATB-PROD-4	ATB-PROD-5
ATB-ORG-1	↗				
ATB-ORG-2		↗			
ATB-ORG-3			↗		
ATB-ORG-4				↗	
ATB-ORG-5					↗

Figura 5.41 - Matriz de relacionamento de atributos de produto e organização.

Os atributos aqui descritos são associados a modelos de nível mais alto, chamado de arquitetura. Esta arquitetura é gerada para cada cenário do processo do ciclo de vida desempenhado para produto e organização. Por exemplo, para o cenário de operação do produto os atributos são associados a um modelo chamado de 'Arquitetura de Operação'. A captura dos elementos que constituem esse modelo, e a associação dos atributos é feita por um diagrama de arquitetura física - PAD.

Para o cenário de desenvolvimento desempenhado pela organização, os atributos são associados a um modelo chamado de 'Arquitetura de Organização de Desenvolvimento', que também é descrito por um diagrama de arquitetura física.

Às setas e aos elementos do diagrama são associados itens de informação dos atributos. Dessa forma ao derivar atributos dos modelos da análise funcional e dos modelos da análise de implementação, é possível em um nível de maior abstração criar uma arquitetura do sistema baseada em relacionamento de atributos. Para cada arquitetura do cenário do processo do ciclo de vida são associados os atributos, que impactam o produto e a organização para o desenvolvimento do sistema.

Isto leva a uma abordagem de modelos baseados em atributos, onde pode se analisar o impacto no sistema. Dessa forma, pode se levar a analisar os

atributos conflitantes de produto e organização, e fazer uma análise de custo, desempenho, cronograma e risco ao longo do ciclo de vida para a tomada de decisão para a implementação da arquitetura final.

As Figuras 5.41 a 5.44 ilustram os modelos de mais alto nível, que formam arquiteturas para cada cenário do processo do ciclo de vida para produto e organização.

A Figura 5.41 ilustra a arquitetura do produto em operação. Atributos aqui são associados aos fluxos entre os elementos do diagrama. Por exemplo, para o fluxo entre CBERS e Lançador podem ser associados atributos do modelo de implementação como 'geometria', e do modelo funcional como 'robustez' ou ainda as condições de deslocamento entre o Brasil e China. A descrição do relacionamento destes atributos identifica, desde o aspecto físico de interface mecânica entre satélite e lançador, até a da capacidade de o satélite de suportar o lançamento e de ser exposto em um ambiente dinâmico.

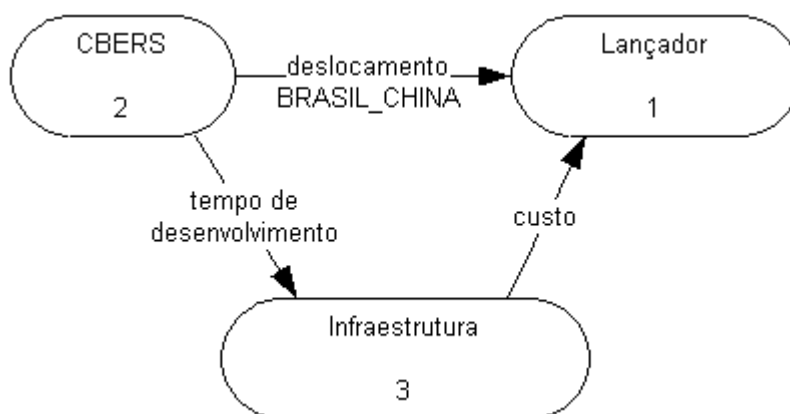


Figura 5.41 - Diagrama de arquitetura do produto em operação.

A Figura 5.42 ilustra a arquitetura do produto em teste.

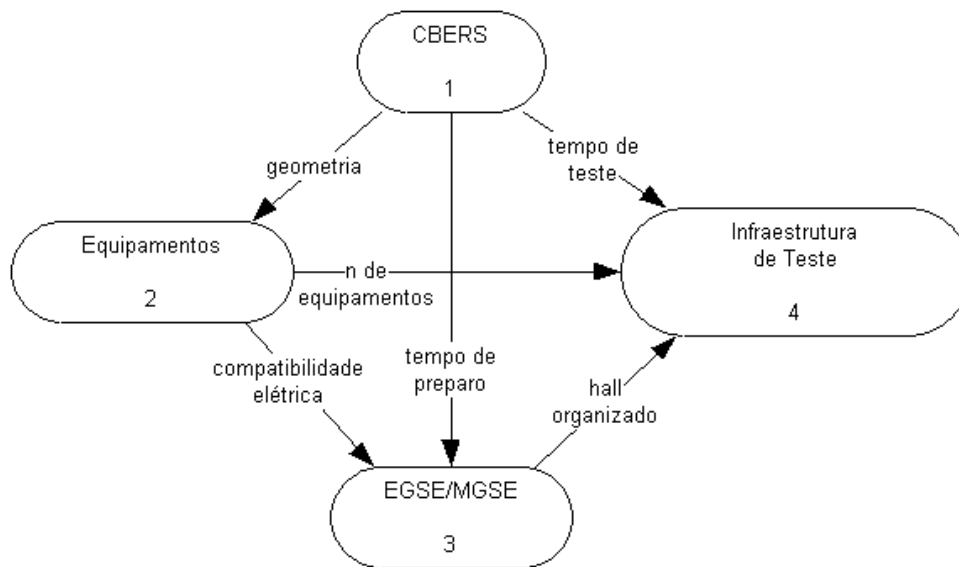


Figura 5.42 - Diagrama de arquitetura do produto em teste.

Figura 5.43 ilustra a arquitetura da organização de desenvolvimento.

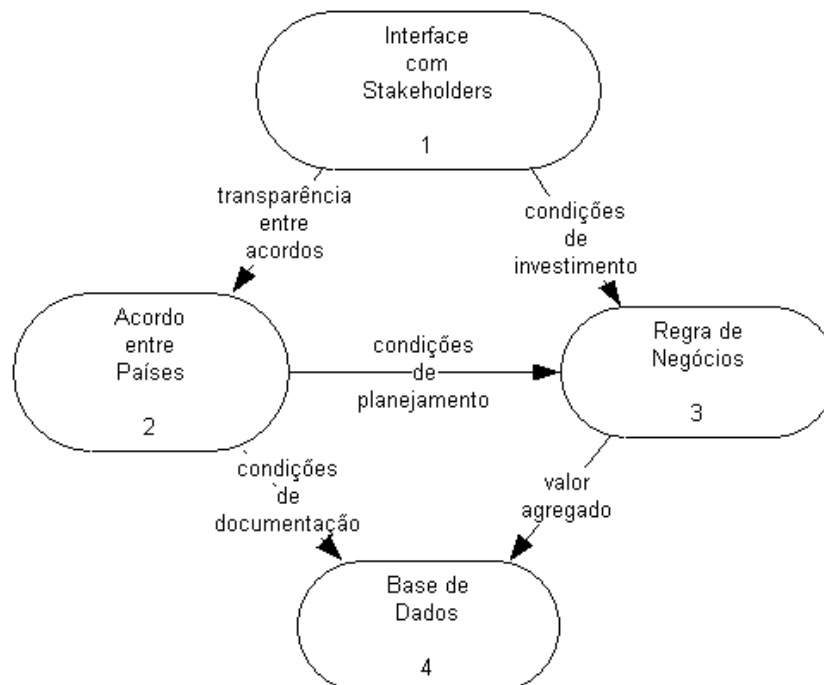


Figura 5.43 - Diagrama de arquitetura da organização de desenvolvimento.

Figura 5.44 ilustra a arquitetura da organização de montagem.

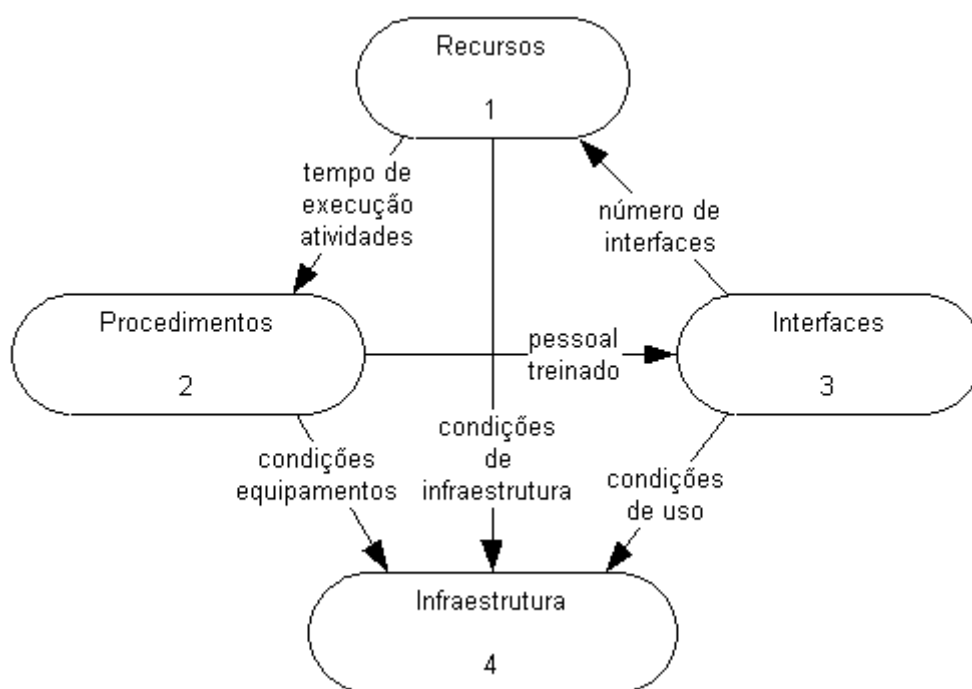


Figura 5.44 - Diagrama de arquitetura da organização de montagem.

Para a exemplificação do auxílio computacional ao processo de engenharia simultânea de sistemas aplicada ao CBERS foi necessário o levantamento de informações contidas em documentos que relacionavam os procedimentos para montagem, integração e testes do CBERS (INPE, 2006).

Esse exemplo mostra a aplicação do método a partir de informações contidas em documento, de um projeto já realizado. Quando o projeto do CBERS fora realizado não houve o auxílio computacional, por exemplo, no processo de captura de requisitos. O exemplo deste Capítulo, cobrindo desde a análise de *stakeholder* até os elementos físicos de implementação ilustram como seria o uso da ferramenta e do processo de engenharia simultânea de sistemas, para por exemplo, capturar os *stakeholders* e seus *concerns* focados em cada cenário dos processos do ciclo de vida.

Esses *stakeholders* seriam capturados pelos modelos e seriam associados a eles seus requisitos, dessa forma seria possível gerar um documento dos *stakeholders* e seus requisitos.

Ainda o auxílio computacional tornaria possível a rastreabilidade por cada elemento do produto e seu requisito associados por *links*. Todo o esforço colocado para preencher o diagrama seria aproveitado para os demais satélites desenvolvidos pra a família CBERS. Essas informações estariam em um ambiente unificado e de maneira integrada, podendo ser analisados os elemento do produto e da organização, ou seja, os recursos para realizar os processos do ciclo de vida do produto.

Conforme o plano de AIT (INPE, 2006) que ilustra processo de AIT para o CBERS foram utilizados diagramas para representar as atividades e as sequências a serem realizadas para montagem, integração e testes do produto. Esses diagramas seguem um formato livre, mas de forma que todos envolvidos para realizar o processo de AIT possam entender a sequência das atividades e os procedimentos.

Com o auxílio computacional ao processo de AIT poderiam ser capturados os processos do ciclo de vida, seus cenários, daí então desdobrados para as atividades e os procedimentos que já estão contidos em documentos. A ferramenta auxilia na captura utilizando os seus diagramas e capturando os itens de informação que descrevem cada etapa, atividade ou procedimento, o que não estão associados junto a cada descrição feita nos documentos. Feito isso uma vez, pode ser replicado por outros produtos que serão submetidos ao processo de AIT, somente mudando as informações e particularidades de cada produto e organização que implementará esse processo. Dessa forma é possível descrever um processo de AIT em alto nível de abstração que será desdobrado em um nível com maiores detalhes que auxilie no desenvolvimento para uma família de satélites de ordem de toneladas, mas também podem ser

reutilizados esses modelos e itens de informação para descrever ou até mesmo otimizar um processo de AIT para uma família de satélites da ordem de pouco mais de um quilograma.

Com o auxílio computacional, torna-se capaz de capturar diversas informações que a variedade de documentos e por sua vez gerado um grande volume, não capturam todos os elementos que descrevem o produto e a organização que implementam os processos do ciclo de vida, e muito menos permite uma reutilização dos modelos de produto e de organização nos quais foram posto muito esforço. Com o *Cradle* é possível manter uma base de dados com seus diversos modelos, requisitos, itens de informação e relacionamentos entre os elementos de produto e organização.

6 APLICAÇÃO DO MÉTODO PARA O VLM

Neste Capítulo é ilustrada a aplicação do processo descrito no Capítulo 3 sobre o exemplo do VLM, utilizando o método descrito no Capítulo 4.

6.1 VLM

O projeto do VLM foi declarado como um projeto orientado a custos (FULINDI et al, 2010). Para seu desenvolvimento foi formado um grupo de engenharia de sistemas, sendo esse grupo composto por diversos especialistas de diferentes áreas, todos com ampla experiência em desenvolvimento de veículos lançadores. Esse grupo alimenta o processo de engenharia simultânea de sistemas baseados na experiência de cada componente do grupo, e utiliza-se o *Cradle* para a captura das informações e para a implementação do projeto. Os exemplos aqui ilustrados foram capturados durante reuniões realizadas semanalmente pelo grupo de engenharia de sistemas. Ao utilizar o *Cradle* como ferramenta computacional no auxílio ao desenvolvimento do projeto, captura de informações baseada na experiência do grupo e gestão dos requisitos, mantém-se assim uma base de dados que pode ser aproveitada nos projetos futuros, e que não serão desperdiçadas com o passar do tempo com retrabalhos.

6.1.1 Descrição da Missão

Aqui é descrita a missão conforme a Subseção 3.3.1, e para o auxílio computacional do método usando *Cradle* descrito na Subseção 4.2.1.

Como o início do processo de engenharia simultânea de sistemas sugere, deve-se capturar a descrição da missão que o sistema irá desempenhar. Portanto a missão do VLM é injetar em órbita uma carga útil de 300 a 500 kg. Essa declaração impacta na tecnologia adotada para o desenvolvimento do lançador, e nas estratégias econômicas para o seu mercado. Por exemplo, uma faixa de capacidade de transporte de carga útil, impacta na quantidade do combustível a ser utilizado e na configuração dos motores a serem utilizados para propelir a quantidade de massa definida para a missão. O tipo de material utilizado para a construção dos propulsores também é um fator impactante, pois a construção de um propulsor que antes era desenvolvido em aço tem sua manipulação, desde a sua concepção até a sua integração no centro de lançamento, diferente de um propulsor feito de material composto. Sua manipulação requer MGSE's que atendam aos requisitos de transporte e manipulação.

Nota-se que a declaração da missão impacta em todo o processo do seu ciclo de vida. Estes processos do ciclo de vida para o produto, e o escopo do esforço de desenvolvimento para a organização são ilustrados a seguir.

6.1.2 Ciclo de Vida e Escopo do Esforço de Desenvolvimento

Aqui são descritos os processos do ciclo de vida, o escopo do esforço de desenvolvimento e os cenários, conforme Subseções 3.3.2 a 3.3.4, usando o *Cradle* para o auxílio computacional, como descrito na Subseção 4.2.1.

Os processos do ciclo de vida são capturados no diagrama eFFBD. Os processos são exemplificados de acordo com a norma da ECSS. Os cenários identificados aqui são para a exemplificação da aplicação da abordagem descrita no Capítulo 3. Na prática, porém, é necessário aplicar o método para todos os processos do ciclo de vida.

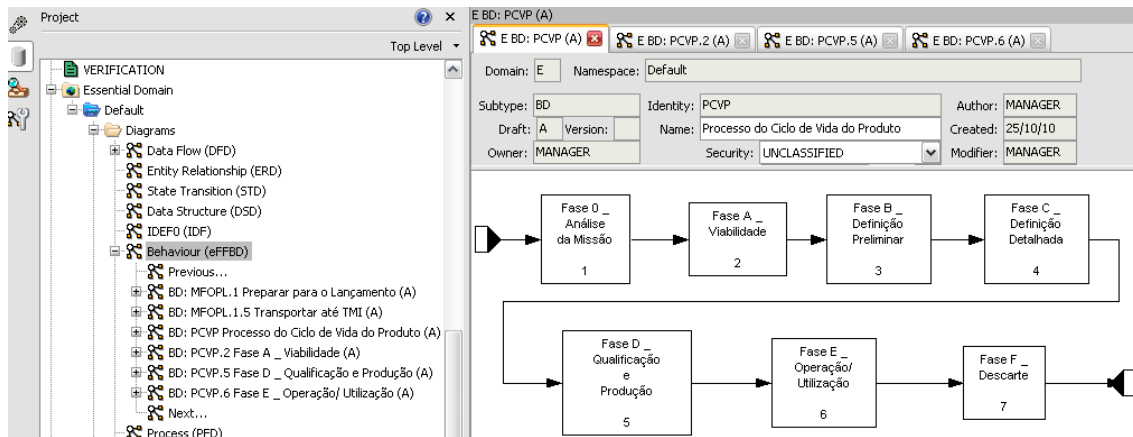


Figura 6.1 - Processos do ciclo de vida do produto.

A Figura 6.2 ilustra a organização para o cenário de desenvolvimento. A partir da Figura 6.2 serão identificados os interesses dos *stakeholders* de onde são derivados os requisitos que impactarão a organização de desenvolvimento do projeto.

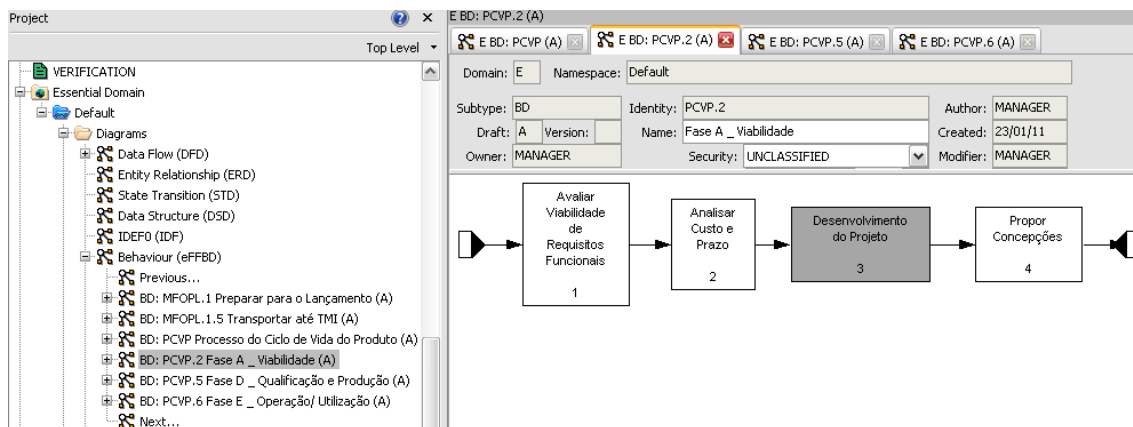


Figura 6.2 - Cenário de desenvolvimento do projeto.

A Figura 6.3 ilustra o cenário de integrar o modelo de voo para o produto. Para este cenário os interesses dos *stakeholders* são capturados, e serão derivados em requisitos que impactarão a integração do modelo de voo do VLM.

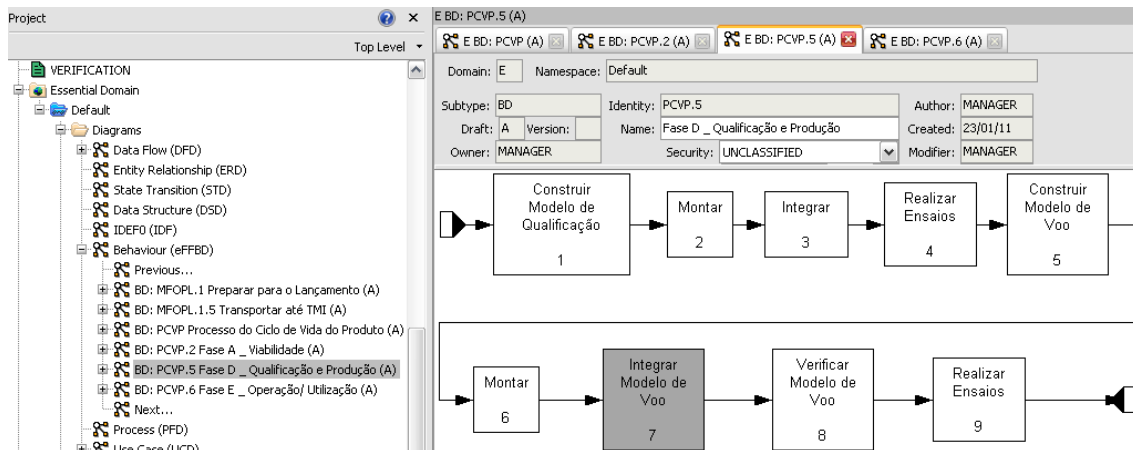


Figura 6.3 - Cenário 'integrar modelo de voo'.

A Figura 6.4 ilustra os cenários de 'preparar para o lançamento' e 'operar'. A organização irá impor os requisitos para o cenário de preparar para o lançamento, onde são identificados os MGSE's para atender os propulsores a serem transportados no centro de lançamento.

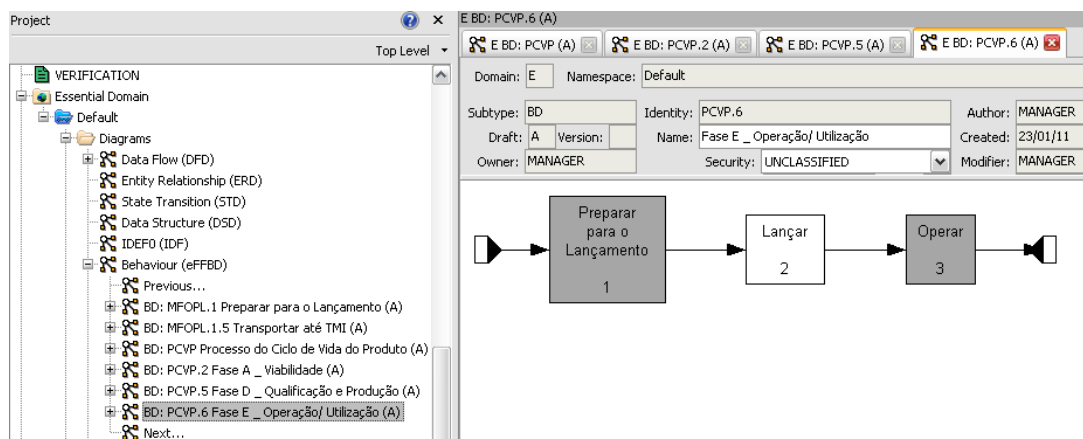


Figura 6.4 - Cenários de preparar para o lançamento e operar.

A Figura 6.5 ilustra o escopo do esforço de desenvolvimento. Os cenários identificados para a organização estão dentro do escopo do esforço de desenvolvimento.

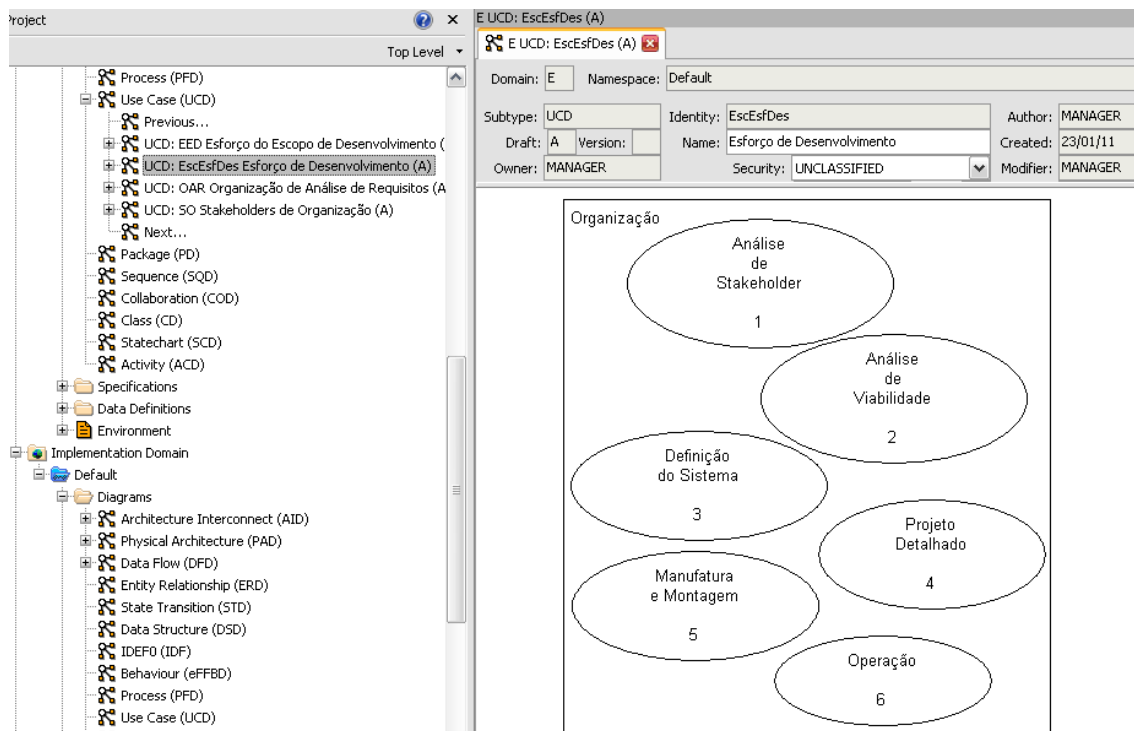


Figura 6.5 - Escopo do esforço de desenvolvimento.

6.1.3 Análise de *Stakeholder* e Análise de Requisitos

Esta Subseção apresenta a análise de *stakeholders* e a análise de requisitos descritas nas Subseções de 3.3.5 a 3.3.10, utilizando o *Cradle* para o auxílio computacional do processo como na Subseção 4.2.2.

A Figura 6.6 ilustra o diagrama DFD dos *stakeholders* e seus interesses no processo de integração do VLM. Esse processo de integração se dá quando o veículo está na torre de lançamento, no prazo da janela de lançamento. As declarações dos *stakeholders* são traduzidas em requisitos que impactarão a forma de executar a integração do veículo. A equipe que prepara a carga útil para ser integrada com o veículo preocupa-se com os procedimentos para a integração. Os meios de transporte são os interesses da equipe de

infraestrutura do centro de lançamento. O desenvolvimento de um novo veículo, com dimensões e propriedades diferentes, impõe requisitos de interface do produto com os MGSEs. Por exemplo, um propulsor construído de aço possui propriedades diferentes das de um propulsor construído de material composto. Isto impacta no seu transporte.

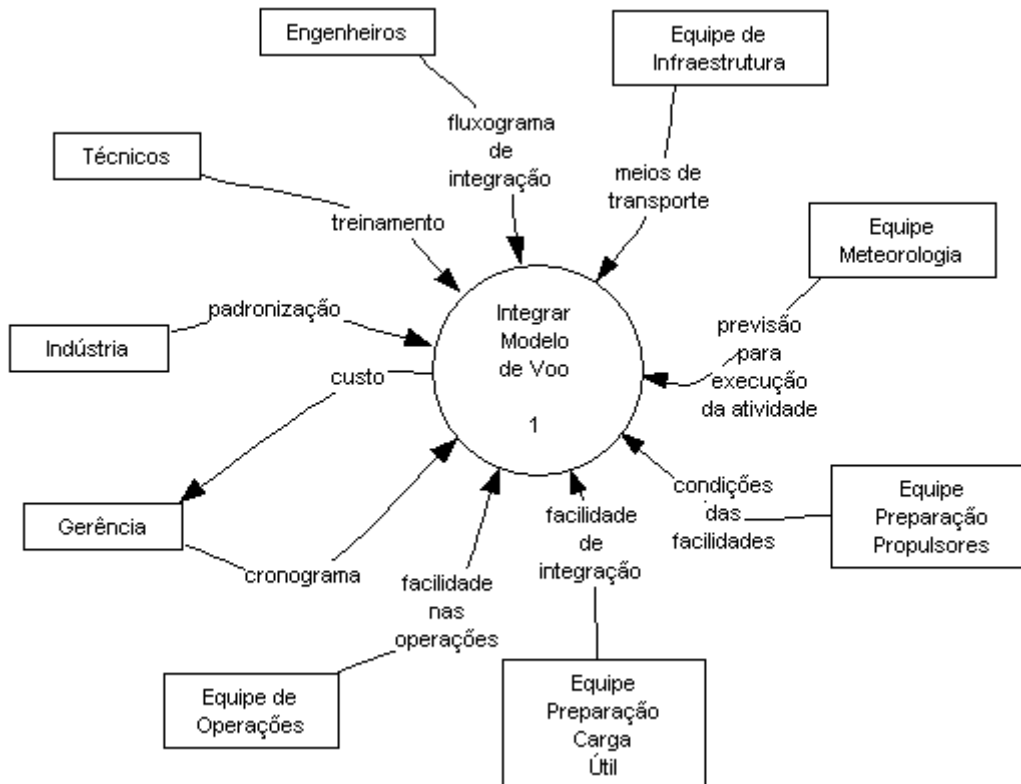


Figura 6.6 - Stakeholders de produto e seus interesses para o cenário de integrar modelo de voo.

A Figura 6.7 ilustra os *stakeholders* interessados na operação do VLM.

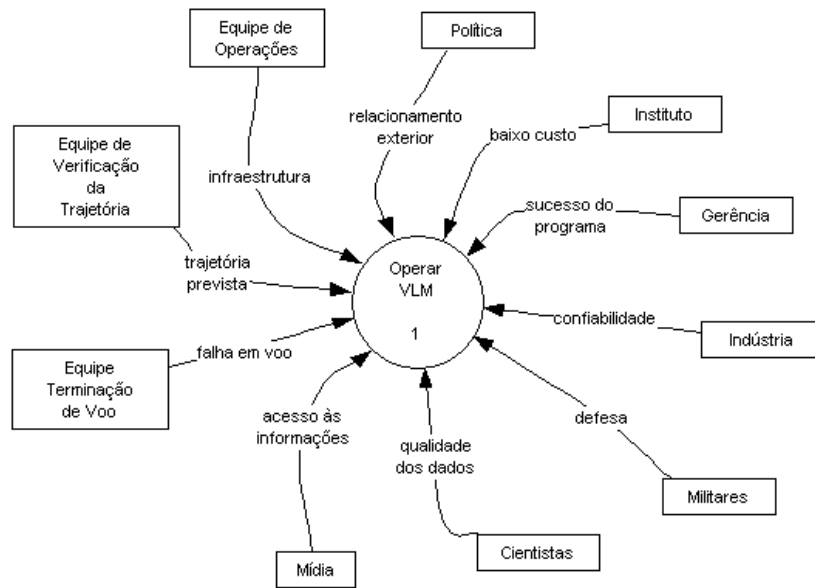


Figura 6.7 - *Stakeholders* de produto e seus interesses para o cenário de operação do VLM.

A Figura 6.8 ilustra os interesses dos *stakeholders* para o desenvolvimento do projeto.

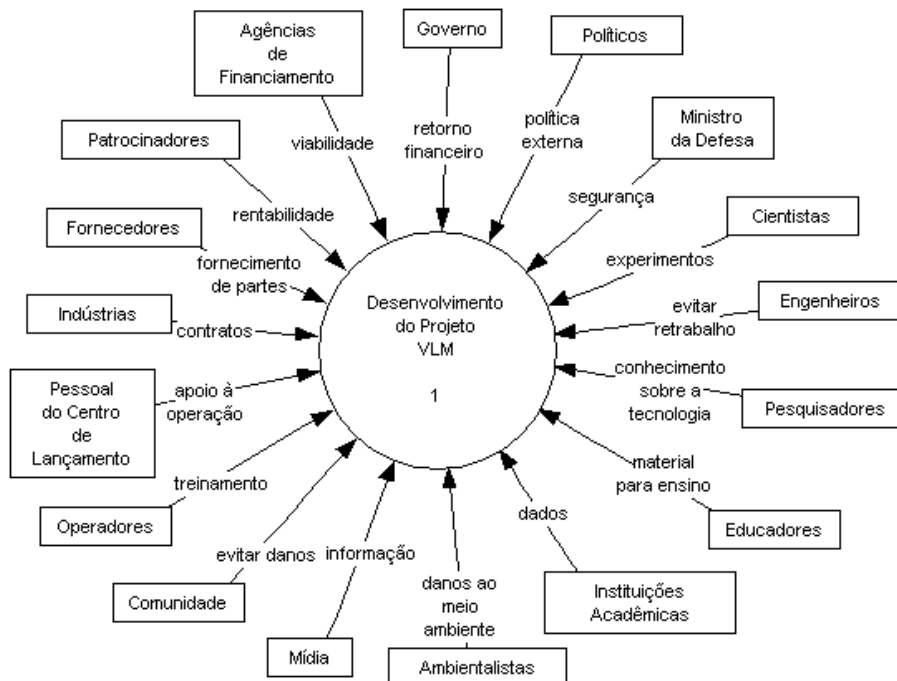


Figura 6.8 - *Stakeholders* de organização e seus interesses para o cenário de desenvolvimento.

A Figura 6.9 ilustra os *stakeholders* e seus interesses para preparar para lançar.

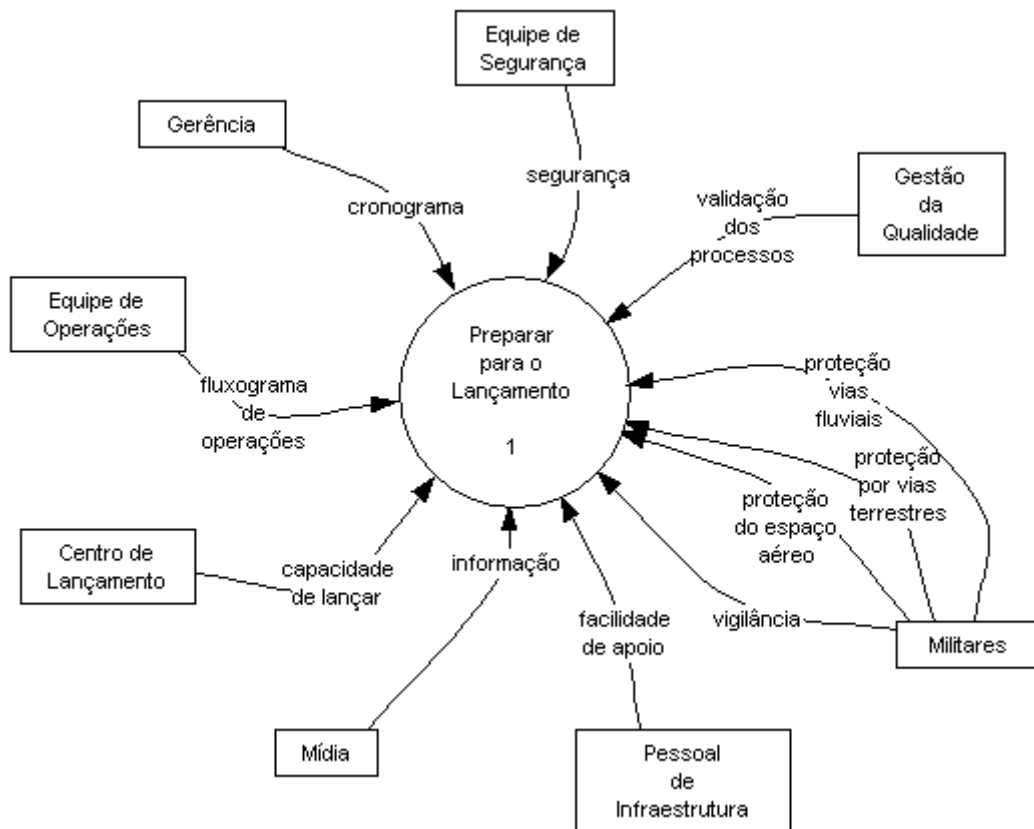


Figura 6.9 - *Stakeholders* de organização e seus interesses para o cenário de preparar para lançamento.

Os diversos *stakeholders* são capturados pelos modelos, e depois passam a ser representados pelos itens de informação. Desta maneira é possível associar os itens de informação *stakeholder* as suas declarações.

A Figura 6.10 ilustra a lista de itens de informação referentes aos *stakeholders*. Suas declarações também são ilustradas na Figura 6.11.

A partir dos itens de informação do tipo 'declarações de *stakeholders*', são associados os itens de informação do tipo 'medidas de efetividade', de onde são derivados os requisitos do sistema.

A Figura 6.12 ilustra um item de informação do tipo MoE.

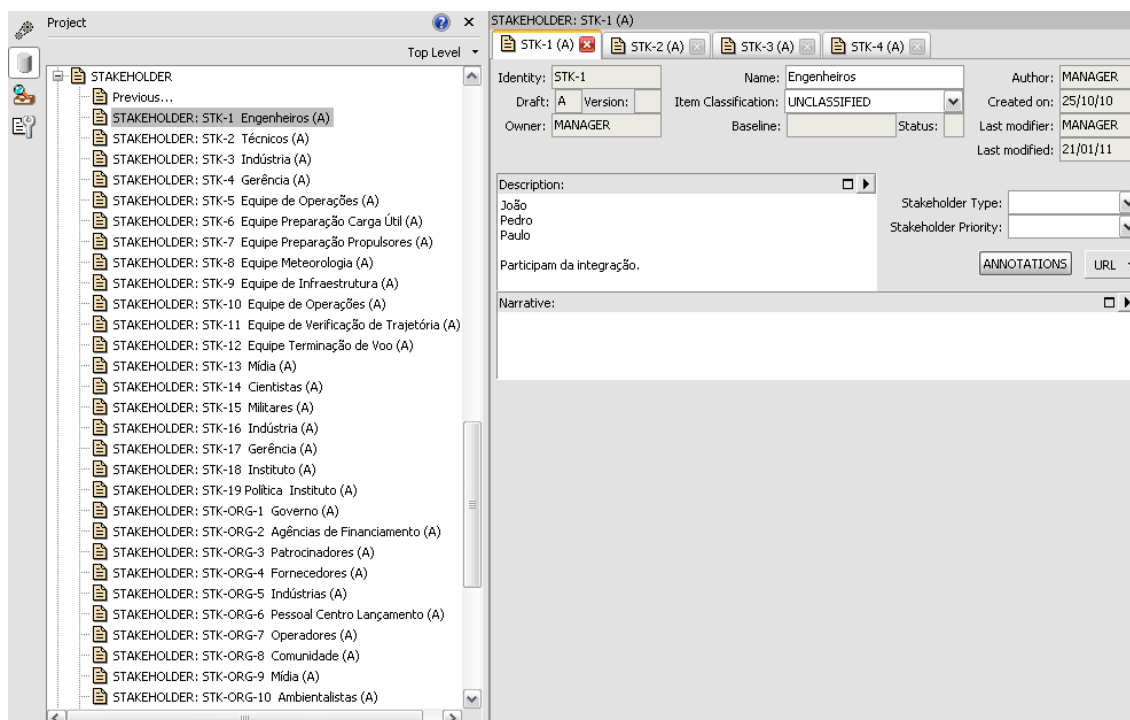


Figura 6.10 - Itens de informação de endereçamento de stakeholders.

O item de informação do tipo interesse de *stakeholder*, endereça as informações capturadas nas setas dos diagramas ilustradas nas Figuras 6.6 a 6.9. Dessa forma é possível gerenciar os interesses dos *stakeholders*, e associá-los com outros itens de informação.

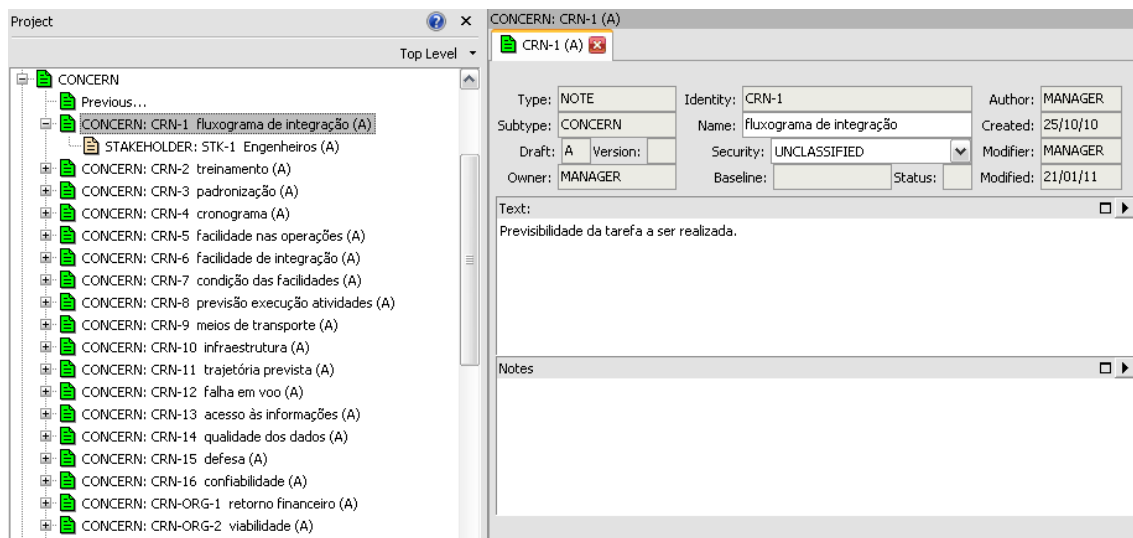


Figura 6.11 - Itens de informação de interesses de stakeholders.

A Figura 6.12 ilustra o item de informação MoEs. Os requisitos devem ser traduzidos em requisitos de sistemas, pelas MoEs.

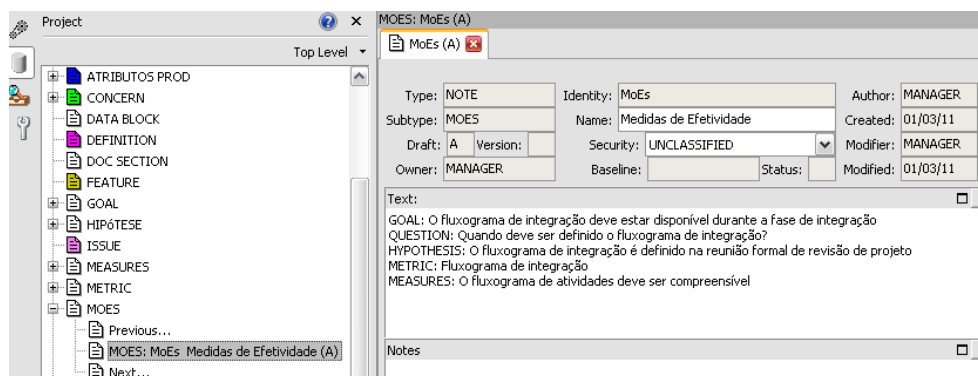


Figura 6.12 - Medidas de efetividade.

A Figura 6.13 ilustra os itens de informação do tipo *stakeholder*, declaração de *stakeholder*, requisito de *stakeholder*, MoE e requisito de sistema associados

por *link* de informação. Nesse *link* podem ser descritos os (*assumptions* e *rationale*).

A Figura 6.14 ilustra os campos de atributos dos *links* de informação.

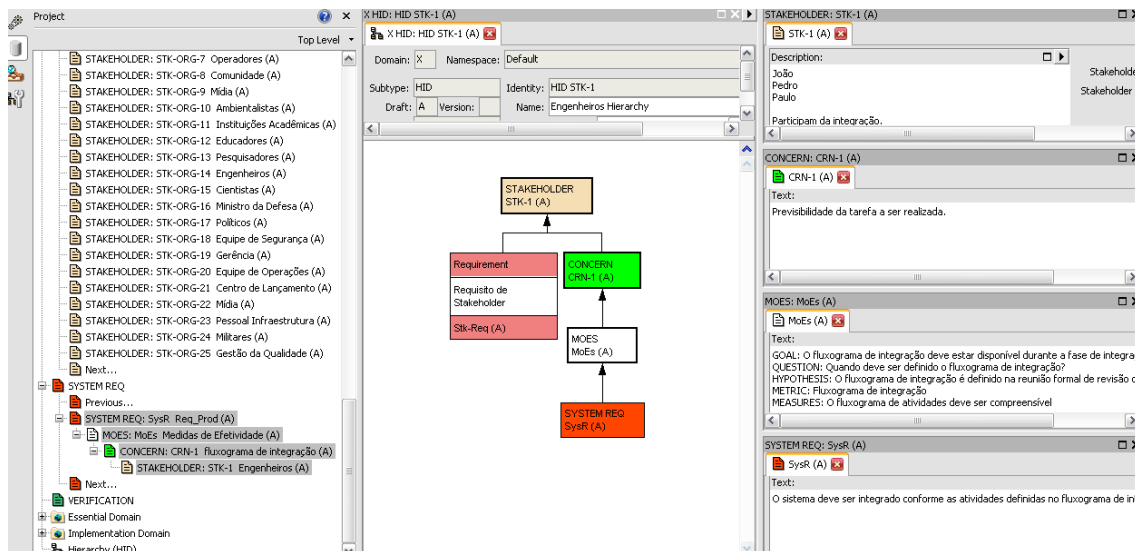


Figura 6.13 - Referências cruzadas entre itens de informação de MoE's.

A razão pela qual os itens de informação são associados pode ser descrita nos *links* de informação. Dessa forma é possível justificar os elementos associados. Cada especialista pode dar a sua contribuição, descrevendo o impacto que isso causará conforme sua experiência em outros projetos.

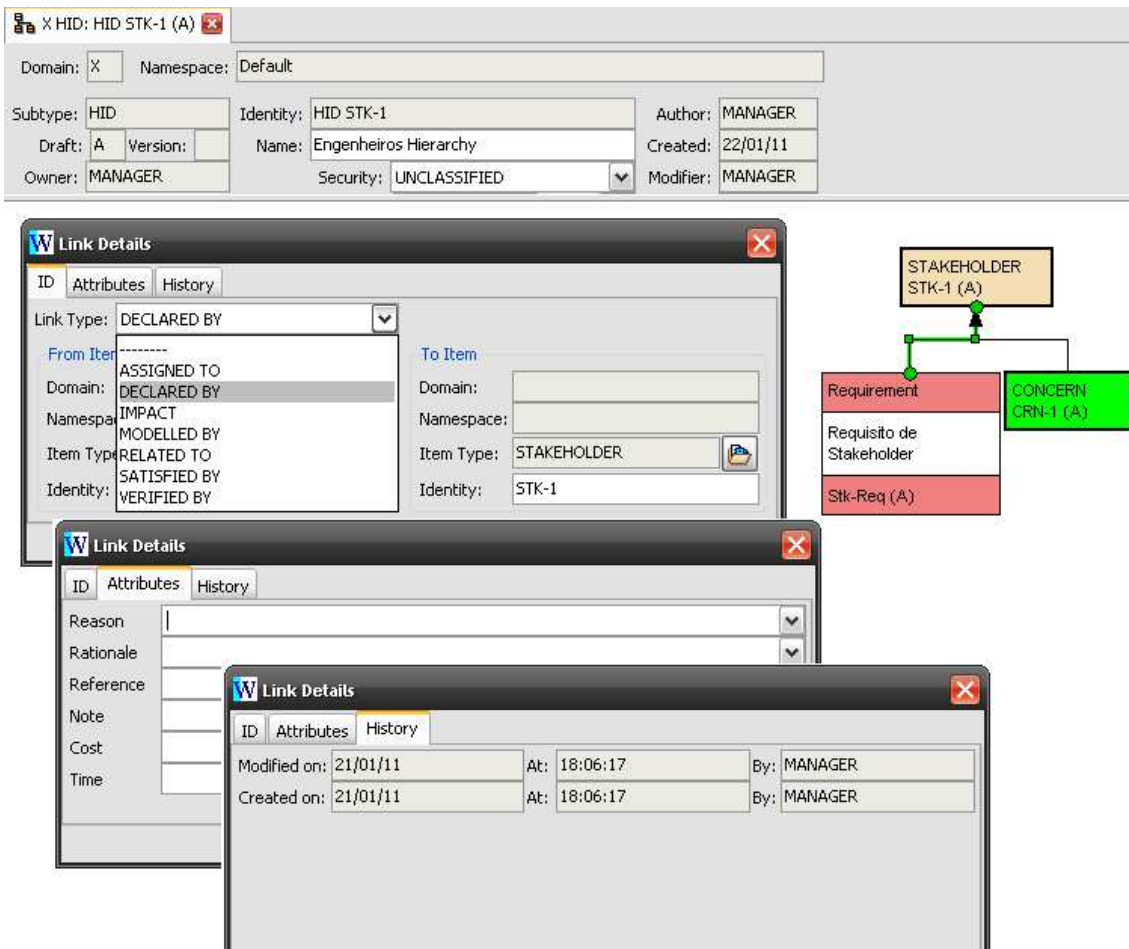


Figura 6.14 - Exemplo de detalhes de link.

Documentos são gerados de maneira a ilustrar a cobertura dos requisitos dos *stakeholders*. A Figura 6.16 ilustra a visualização no *Cradle*, de uma tabela gerada para conferir a cobertura dos requisitos. A Figura 6.17 ilustra o documento gerado a partir dessa tabela gerada pelo *Cradle*.

	Identity	Ver	Dft	Name	All Linked Items			
					Identity	Name	Version	Draft
Previous...								
1	STK-1		A	Engenheiros	Stk-Req	Requisito de Stakeholder		A
					CRN-1	fluxograma de integração		A
2	STK-2		A	Técnicos	CRN-2	treinamento		A
3	STK-3		A	Indústria	CRN-3	padronização		A
4	STK-4		A	Gerência	CRN-4	cronograma		A
5	STK-5		A	Equipe de Operações	CRN-5	facilidade nas operações		A
6	STK-6		A	Equipe Preparação Carga Útil	CRN-6	facilidade de integração		A
7	STK-7		A	Equipe Preparação Propulsores	CRN-7	condição das facilidades		A
8	STK-8		A	Equipe Meteorologia	CRN-8	previsão execução atividades		A
9	STK-9		A	Equipe de Infraestrutura	CRN-9	meios de transporte		A
10	STK-10		A	Equipe de Operações	CRN-10	infraestrutura		A
11	STK-11		A	Equipe de Verificação de Trajetória	CRN-11	trajetória prevista		A
12	STK-12		A	Equipe Terminação de Voo	CRN-12	falha em voo		A
13	STK-13		A	Mídia	CRN-13	acesso às informações		A
14	STK-14		A	Cientistas	CRN-14	qualidade dos dados		A
15	STK-15		A	Militares	CRN-15	defesa		A
16	STK-16		A	Indústria	CRN-16	confiabilidade		A

Figura 6.15 - Exibição de stakeholder, interesse e requisito de stakeholder.

As informações podem ser importadas do *Cradle* para compor um documento para a análise de uma revisão de requisitos de sistemas.

Exemplo de documento - Microsoft Word

Arquivo Editar Exibir Inserir Formatar Ferramentas Tabela Janela Ajuda Traduzir

100% Times New Roman 10

STAKEHOLDER - All

Identity	Ver	Dft	Name	All Linked Items			
				Identity	Name	Version	Draft
STK-1		A	Engenheiros	Stk-Req	Requisito de Stakeholder		A
				CRN-1	fluxograma de integração		A
STK-2		A	Técnicos	CRN-2	treinamento		A
STK-3		A	Indústria	CRN-3	padronização		A
STK-4		A	Gerência	CRN-4	cronograma		A
STK-5		A	Equipe de Operações	CRN-5	facilidade nas operações		A
STK-6		A	Equipe Preparação Carga Útil	CRN-6	facilidade de integração		A
STK-7		A	Equipe Preparação Propulsores	CRN-7	condição das facilidades		A
STK-8		A	Equipe Meteorologia	CRN-8	previsão execução atividades		A
STK-9		A	Equipe de Infraestrutura	CRN-9	meios de transporte		A
STK-10		A	Equipe de Operações	CRN-10	infraestrutura		A
STK-11		A	Equipe de Verificação de Trajetória	CRN-11	trajetória prevista		A
STK-12		A	Equipe Terminação de Voo	CRN-12	falha em voo		A
STK-13		A	Mídia	CRN-13	acesso às informações		A
STK-14		A	Cientistas	CRN-14	qualidade dos dados		A
STK-15		A	Militares	CRN-15	defesa		A
STK-16		A	Indústria	CRN-16	confiabilidade		A
STK-17		A	Gerência				
STK-18		A	Instituto				
STK-19		A	Instituto Política				
STK-ORG-1		A	Governo	CRN-ORG-1	retorno financeiro		A
STK-ORG-2		A	Agências de Financiamento	CRN-ORG-2	viabilidade		A
STK-ORG-3		A	Patrocinadores	CRN-ORG-3	rentabilidade		A
STK-ORG-4		A	Fornecedores	CRN-ORG-4	fornecimento de partes		A
STK-ORG-5		A	Indústrias	CRN-ORG-5	contratos		A
STK-ORG-6		A	Pessoal Centro Lançamento	CRN-ORG-6	apoio à operação		A

Figura 6.16 - Exemplo de uma tabela gerada em extensão RTF.

6.1.4 Análise Funcional

Esta Subseção apresenta a análise funcional descrita nas Subseções de 3.3.11 a 3.3.18, utilizando o *Cradle* para o auxílio computacional do processo como na Subseção 4.2.3.

O transporte dos propulsores ao chegar ao centro de lançamento, foi identificado como um fator de impacto nos MGSE's. Pois novos MGSE's necessitam ser construídos para atenderem aos requisitos de transporte dos propulsores. Essa oportunidade foi identificada ao realizar a análise funcional

para a organização de desenvolvimento do projeto. A Figura 6.21 ilustra esse caso.

O VLM durante a sua integração na torre de lançamento sofre diversos tipos de interações. A Figura 6.17 ilustra o modelo de contexto funcional dessa interação. Para serem capturados os elementos que impactarão o produto, nesse estágio de integração o modelo é desdobrado para melhores detalhes conforme a Figura 6.18 ilustra.

Os atributos derivados desses modelos têm impactos nos aspectos de capacidade de operação e na robustez do veículo.

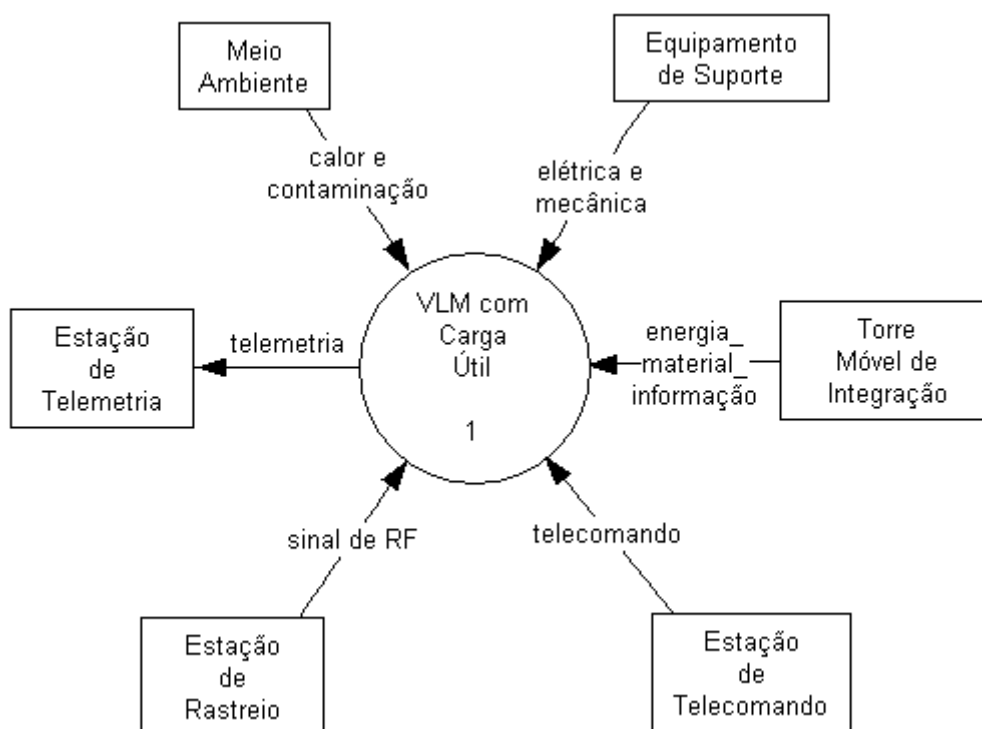


Figura 6.17 - Contexto funcional do produto no cenário 'integrar VLM' contando com a carga útil.

A Figura 6.18 auxilia a demonstrar os fluxos dos sinais que interagem com o sistema. Os atributos são capturados olhando para os fluxos de sinais, e para o impacto dessas interações. Por exemplo, o impacto no veículo ao receber diversos sinais resulta no atributo 'robustez'.

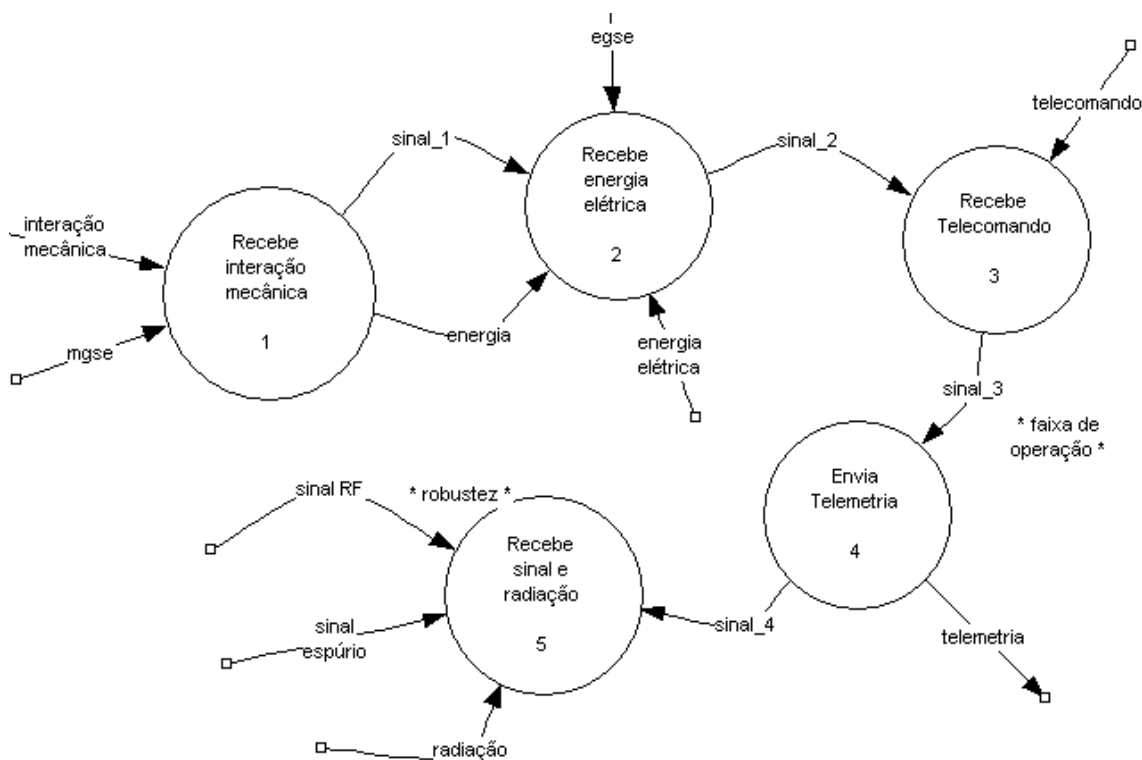


Figura 6.18 - Desdobramento do contexto do produto no cenário 'integrar VLM'.

Durante as separações de estágio, o veículo sofre diversas atuações de forças. Isso impacta na concepção de sua arquitetura e nos materiais a serem empregados na sua construção.

A Figura 6.19 ilustra as fontes das diversas interações que o sistema sofre nesse contexto do cenário do ciclo de vida do produto.

Funções que irão descrever o sistema são identificadas. Para cada função identificada é associado um item de informação do tipo risco. Essa maneira de associar risco à função é ilustrada nas Figuras 6.25 a 6.30, para produto e organização.

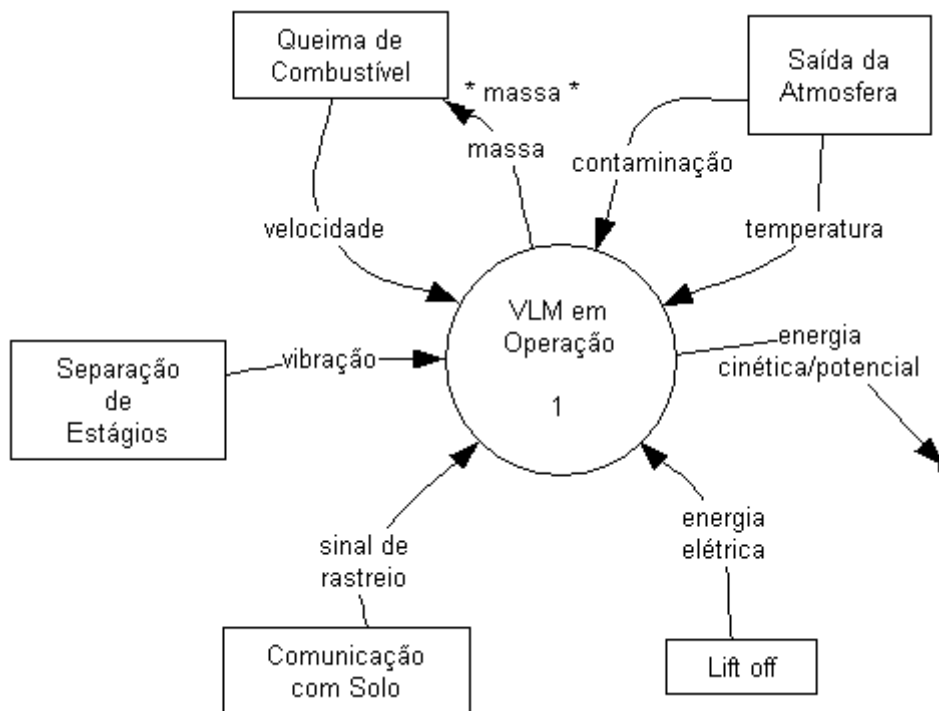


Figura 6.19 - Contexto funcional do produto em operação.

A Figura 6.20 auxilia a ilustrar de maneira simplificada os elementos que interagem com o sistema durante sua operação, desdobrados a partir do diagrama da Figura 6.19.

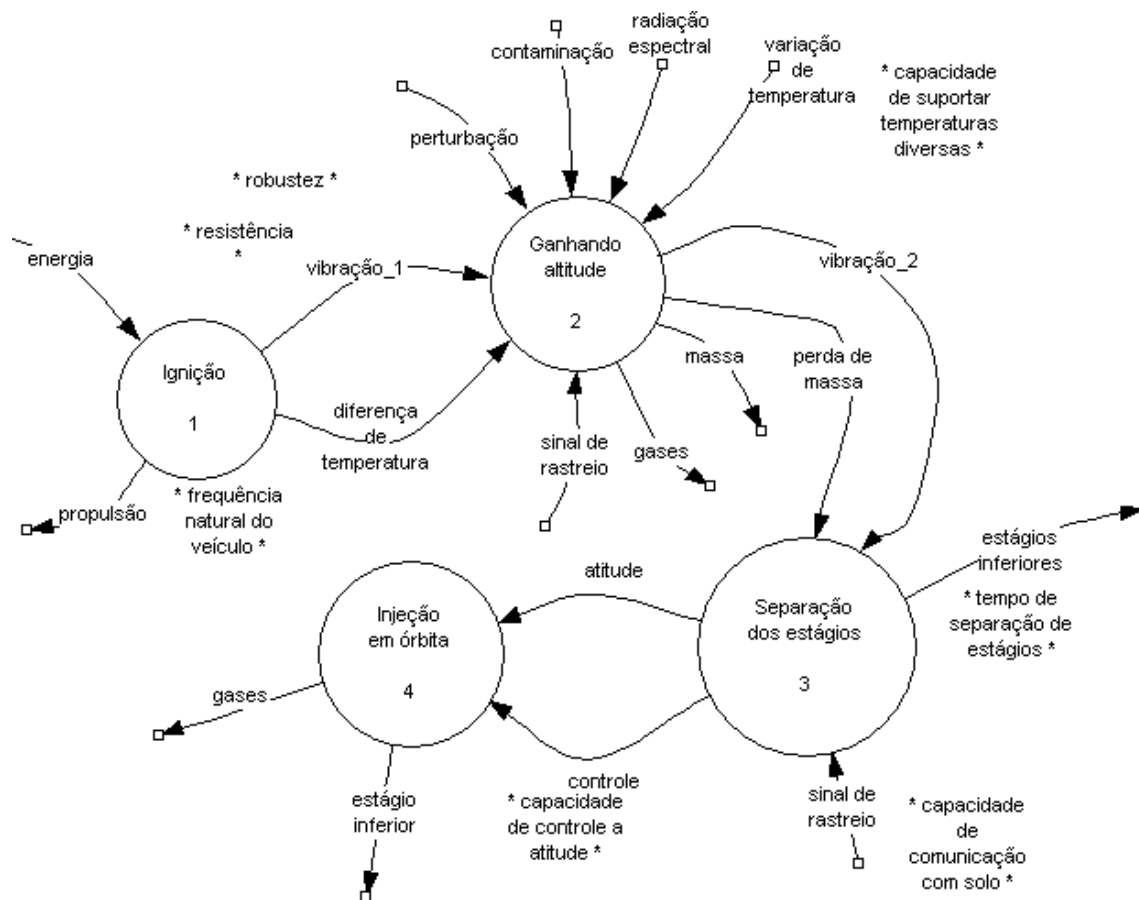


Figura 6.20 - Desdobramento do contexto do produto em operação.

Durante a realização da análise funcional para os cenários do ciclo de vida, dentro do escopo do esforço de desenvolvimento, a organização pôde focar nos elementos que impactariam para o desenvolvimento do veículo.

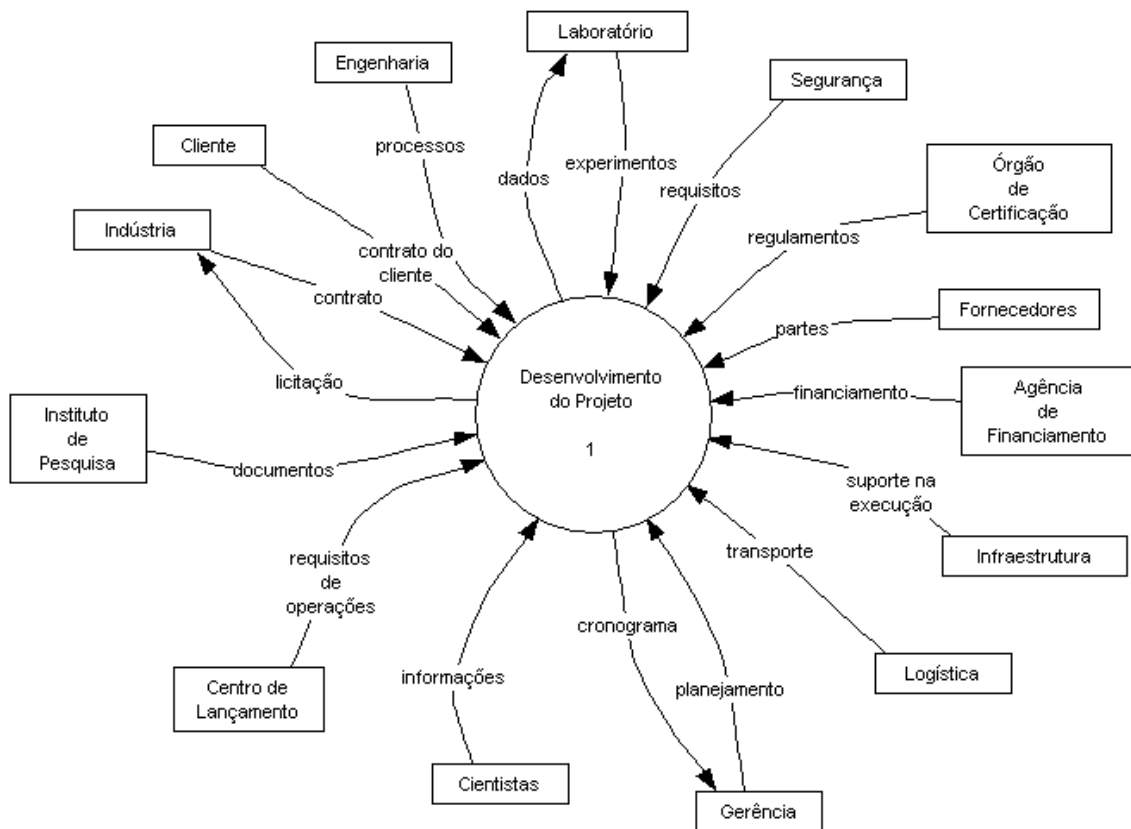


Figura 6.21 - Contexto funcional de organização de desenvolvimento.

Ao identificar os elementos de interação com o sistema no cenário de preparar para o lançamento, os meios de transportes impõem requisitos para o produto. Dessa forma atributos de produto e de organização, derivados destes modelos são relacionados e permitem identificar os fatores impactantes no projeto. Isso permite ser julgado pelo grupo de engenharia de sistemas em tomar decisões sobre o sistema.

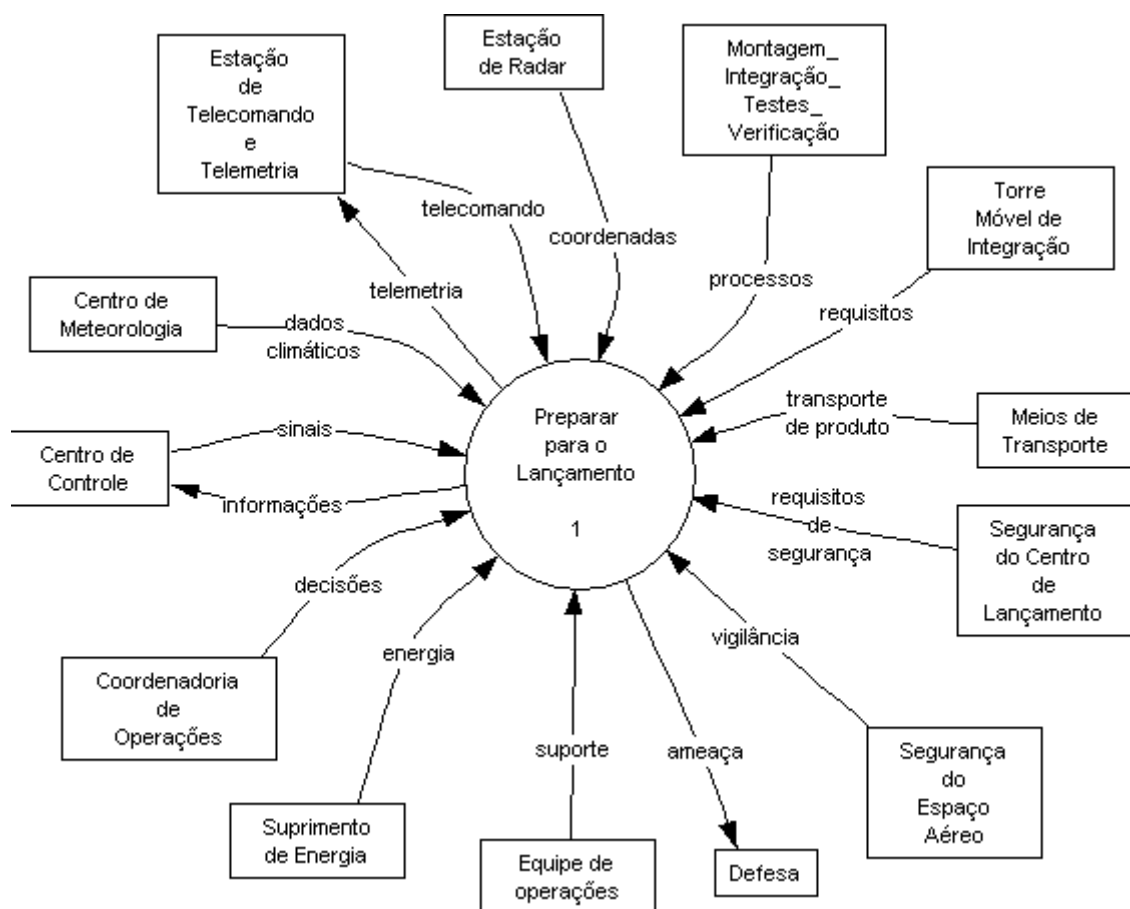


Figura 6.22 - Contexto funcional de organização de 'preparar para o lançamento'.

O desdobramento em diagramas de comportamento do tipo eFFBD, permitem analisar as atividades para a preparação para o lançamento. Esse desdobramento é realizado a partir da Figura 6.22.

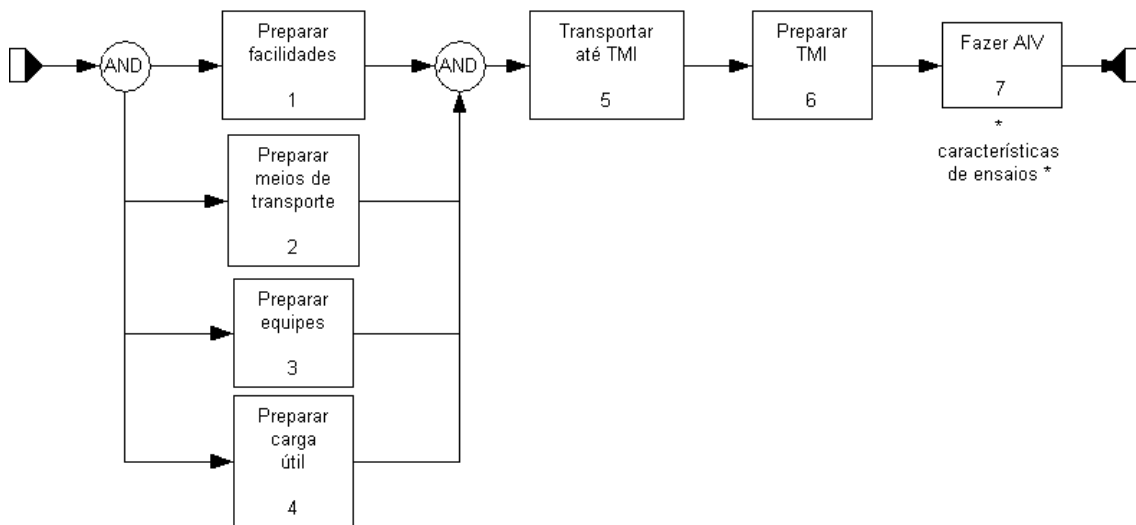


Figura 6.23 - Desdobramento do contexto funcional de organização de 'preparar para lançar'.

A Figura 6.24 é desdobrada a partir da atividade de transportar até a torre móvel de integração ilustrada na Figura 6.23.

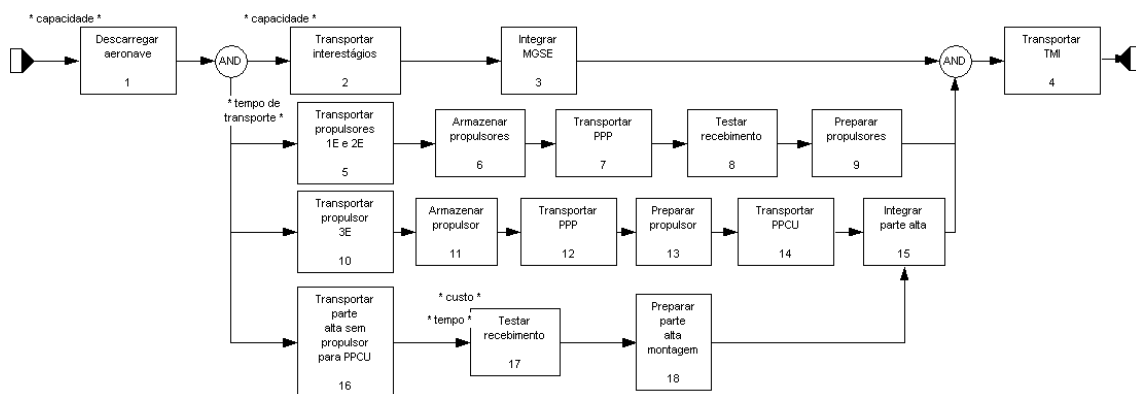


Figura 6.24 - Desdobramento da atividade 'transportar até torre móvel de integração - TMI'.

Uma análise de risco é realizada sobre as funções identificadas para o sistema. As Figuras de 6.25 a 6.30 ilustram um exemplo de cada, de como é feito

utilizando o *Cradle* para associar os itens de informação do tipo risco às funções realizadas para o produto e para a organização.

A Figura 6.25 ilustra o *layout* da ferramenta para a captura e gerenciamento das funções que são geradas pelos diagramas. Esse elemento permite ser associado a itens de informação. Neste caso, o elemento 'separação de estágios' será associado a um item de informação do tipo risco. Desta maneira é permitido representar que esse elemento desempenha uma função no sistema.

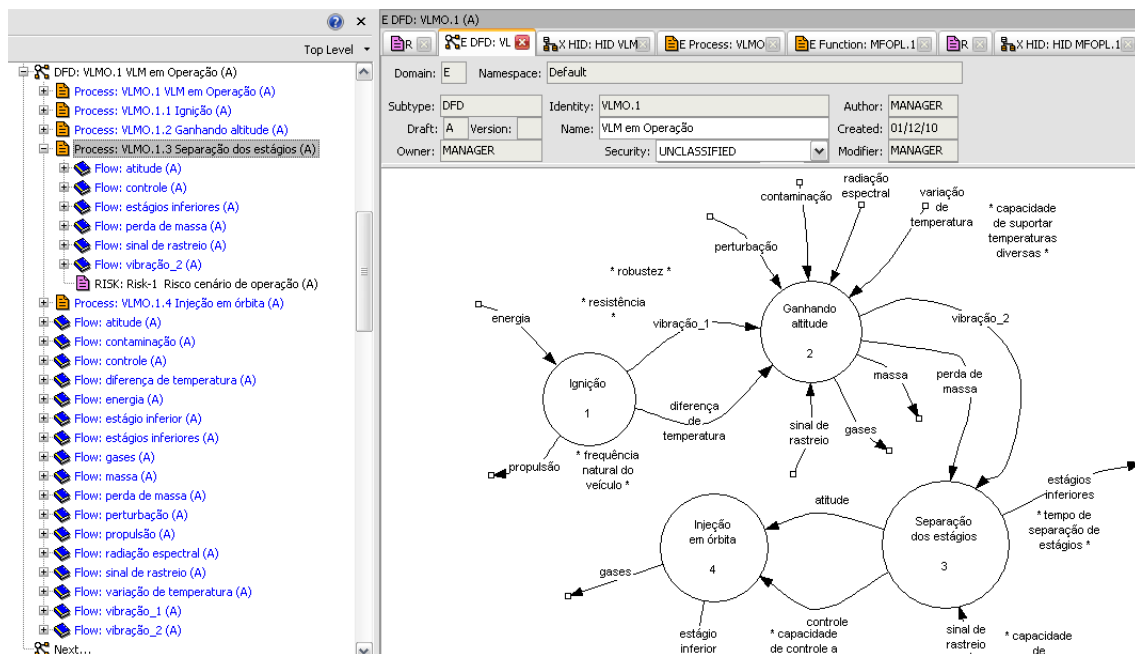


Figura 6.25 - Risco associado a uma função.

O elemento *process* também é um item de informação. A Figura 6.26 ilustra uma nota onde são inseridas informações concernentes a separação dos estágios.

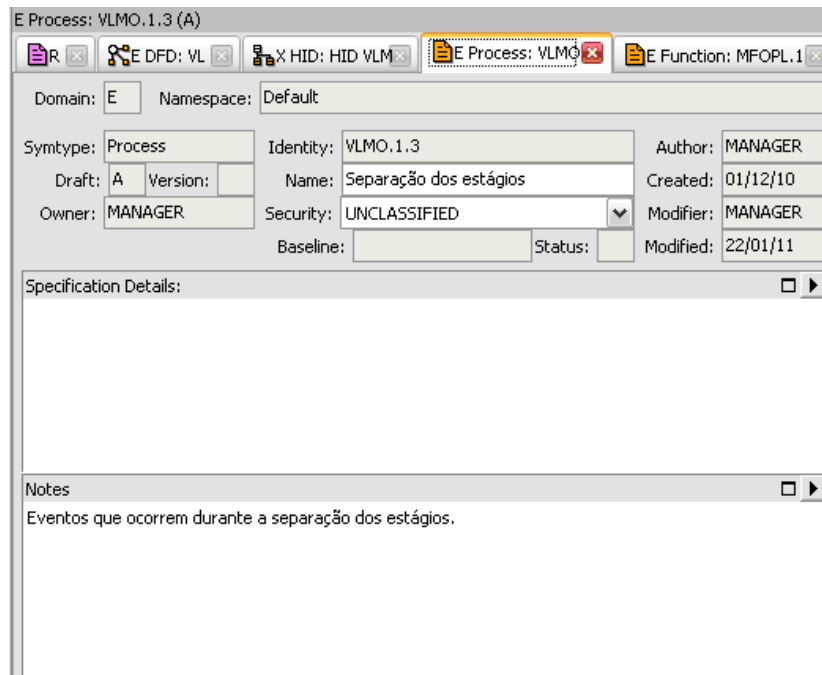


Figura 6.26 - Descrição da função modelo do produto.

A Figura 6.27 ilustra o item de informação do tipo risco. Este item de informação é padrão no *Cradle*. Assim cada especialista pode acessar no *Cradle* com sua conta de usuário e atribuir as informações a respeito do risco, de acordo com a sua experiência. Esse item ainda pode ser submetido durante uma reunião, onde um plano de mitigação pode ser elaborado por mais especialistas. As informações podem ser capturadas e descritas como uma nota no mesmo item.

RISK: Risk-1 (A)

R E DFD: VL X HID: HID VLM E Process: VLMO E Function: MFOPL.1

Name: Risco cenário de operação

Identity: Risk-1 Risk Type: Author: MANAGER
 Draft: A Version: Status: Created on: 22/01/11
 Owner: MANAGER Classification: UNCLASSIFIED Last modifier: MANAGER
 Baseline: Status: Last modified: 22/01/11

Risk Details: ▶
 Causa: Não separar estágios.
 Perigo: O veículo fica sem estabilidade e sem controle.
 Falha: Cinta de separação.
 Consequência: Não cumpre a missão.

Responsibility: ▶
 Risk Probability: MEDIUM ▶
 Risk Impact: HIGH ▶
 Risk Priority: CRITICAL ▶
 Mitigation: ACCEPT ▶

Date Raised: ▶ Review Date: ▶ ANNOTATIONS

Notes: ▶
 Mitigation Plan ▶
 Testes funcionais de separação de estágios.

Figura 6.27 - Item de informação do tipo 'risco'.

A Figura 6.28 ilustra um item de informação do tipo risco associado a uma função do produto.

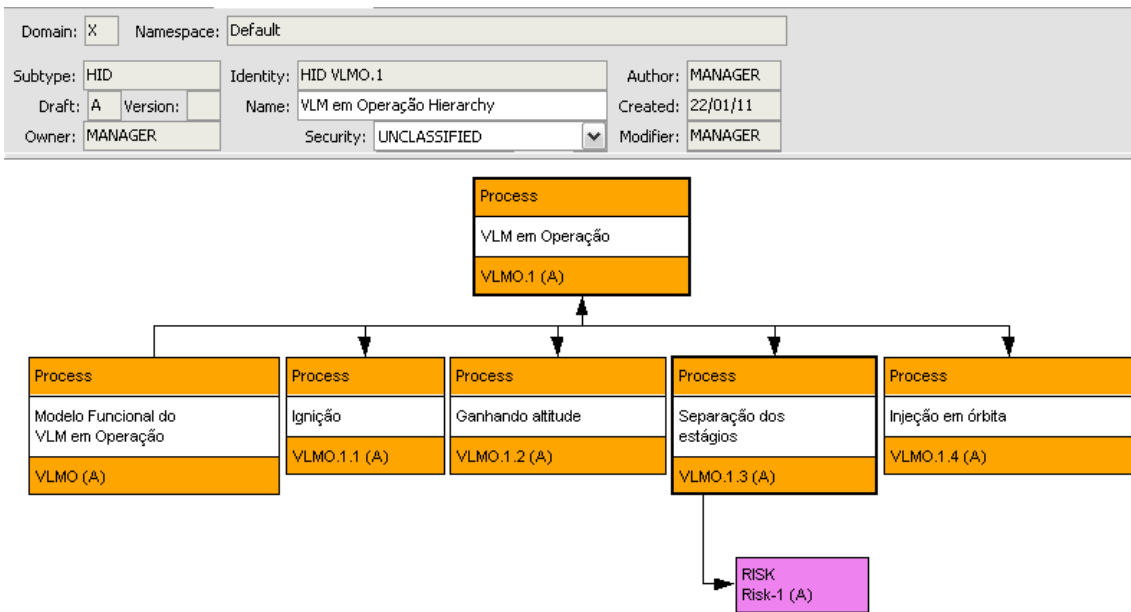


Figura 6.28 - Link de risco à função.

Para os elementos gerados pelos modelos de organização, também são capturados e associados os riscos. Dessa maneira é possível representar uma forma de análise de riscos associados às funções da organização. As Figuras 6.29 e 6.30 ilustram o risco associado a uma função da organização.

Figura 6.29 - Risco organização de preparar para lançar, transportar estágios.

A Figura 6.30 ilustra uma forma de ver em hierarquia os elementos gerados pelos diagramas, e os itens de informação. São exibidos os elementos dos diagramas e os itens de informação que são associados entre si. Usando os *links* pode ser mantida a rastreabilidade. A Figura 6.30 ilustra os elementos gerados pelo diagrama representado na Figura 6.24, e o risco associado à função de transportar propulsores do primeiro e segundo estágios.

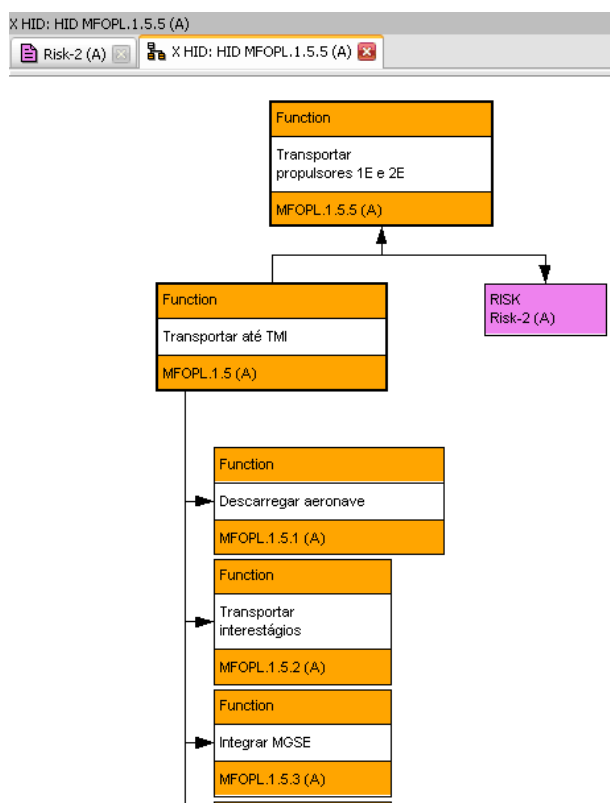


Figura 6.30 - Link à função de organização.

6.1.5 Análise de Implementação

Esta Subseção apresenta a análise de implementação descrita nas Subseções de 3.3.19 a 3.3.27, utilizando o *Cradle* para o auxílio computacional do processo como na Subseção 4.2.4.

Para a análise da implementação do VLM, foram identificados os fluxos da arquitetura e as interfaces do sistema. As imagens importadas no ambiente do *Cradle* auxiliam a descrever melhor as soluções a serem adotadas para o sistema. Por exemplo, as imagens focam no aspecto das interfaces entre o primeiro e segundo estágio. Os especialistas podem identificar, nas interações físicas os elementos impactantes que costumam gerar problemas durante a construção do sistema. Por exemplo, as interfaces possuem elementos de fixação que devem ser levados em consideração para a sua construção, atendendo o posicionamento dos furos e modelada a quantidade deles que serão necessárias.

A Figura 6.31 ilustra em mais alto nível de abstração as interfaces entre os estágios e interestágios. Para representar a interface e o interestágio, é possível inserir no *Cradle* figuras que ilustram essas interfaces.

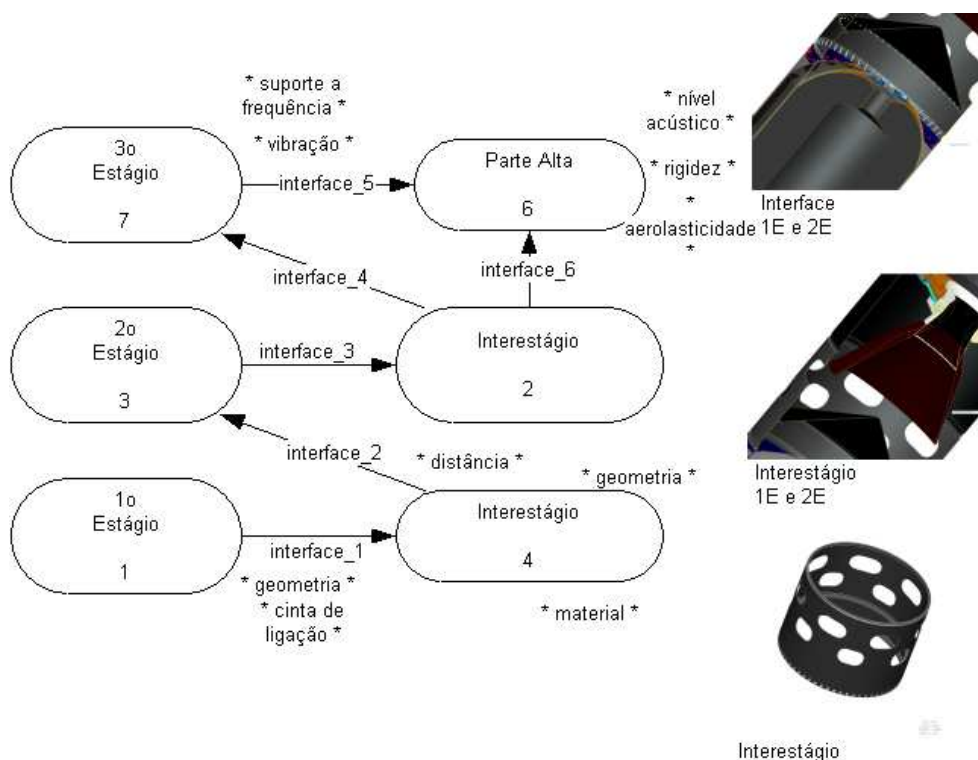


Figura 6.31 - Interconexão de arquitetura do produto.

A Figura 6.32 ilustra o fluxo da arquitetura. Isso auxilia o especialista que indicará o tipo da tecnologia empregada para implementar este tipo de arquitetura.

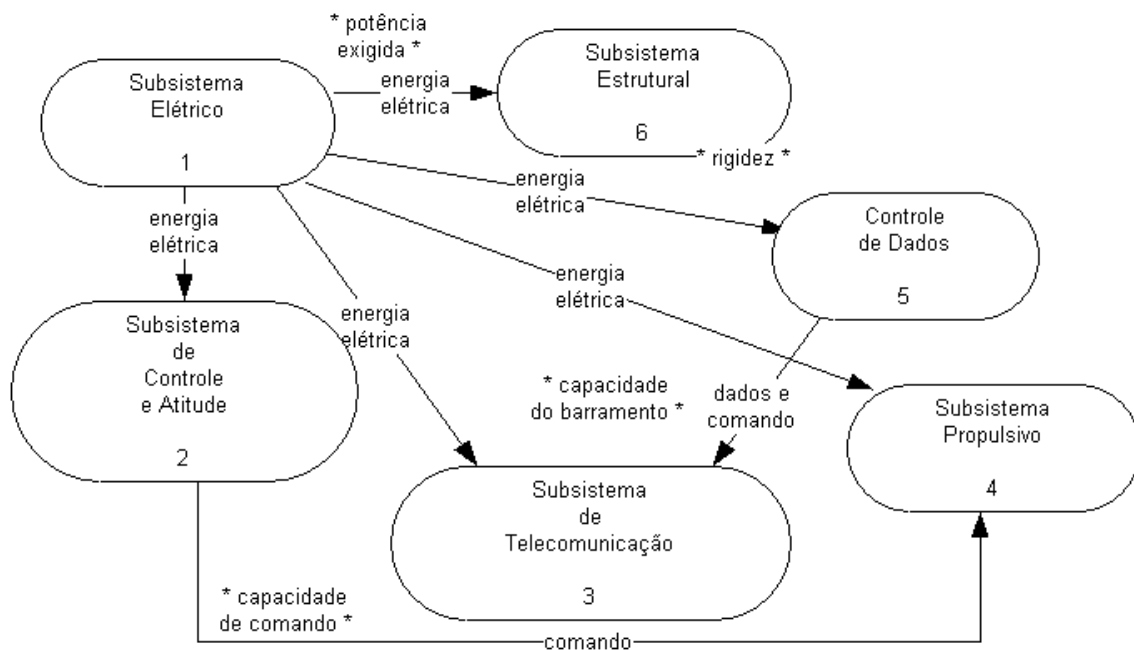


Figura 6.32 - Fluxo de arquitetura de produto.

A Figura 6.32 ilustra as interfaces físicas da organização, pelas quais interagem com o grupo de engenharia de sistemas. É identificando essas interfaces que se pode fazer um planejamento da alocação do time que irá compor o grupo de engenharia de sistemas. Assim podem-se observar quais as disciplinas que irão alimentar o processo de engenharia simultânea de sistemas para o desenvolvimento do veículo.

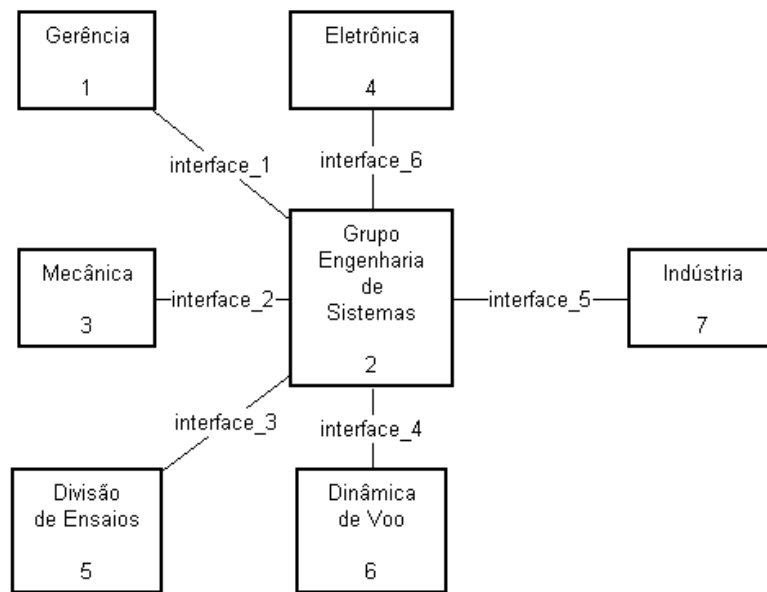


Figura 6.33 - Contexto de implementação de organização de engenharia de sistemas.

A Figura 6.34 ilustra o interrelacionamento dos blocos físicos para o cenário da organização de preparar para o lançamento. Dessa maneira, é possível identificar atributos que impactam na implementação dos recursos para o lançamento. A organização de preparar para o lançamento deve saber das condições a serem oferecidas para um tipo de veículo com, características distintas dos suportados pelas operações anteriores. Atributos de condição de pista de pouso do aeroporto, área dos prédios de armazenagem onde serão estocados os propulsores são atributos que impactam nos aspectos físicos da organização. Por exemplo, se um prédio de armazenagem não suporta um motor com características e propriedades diferentes do qual ele fora projetado, será necessário a construção de outro prédio que atenda às especificações do motor.

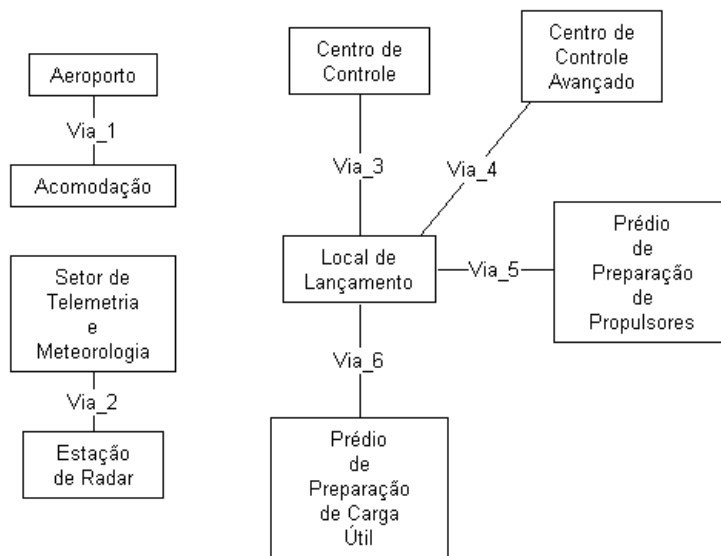


Figura 6.34 - Interconexão de arquitetura da organização de preparar para o lançamento.

Até aqui foram ilustrados os diagramas capturados para produto e organização, realizados para a análise de implementação. Nesse nível já se pode descrever soluções para o projeto. Essas soluções servem de base para a tomada de decisão sobre as tecnologias a serem empregadas, e sobre os recursos a serem empregados nos diversos setores da organização que implementaram os processos do ciclo de vida.

6.1.6 Atributos

Esta Subseção apresenta os atributos derivados dos modelos funcionais e dos modelos de implementação descritos na Subseção 3.3.28.

A Tabela 6.1 ilustra os atributos derivados dos modelos funcionais, e dos modelos de implementação.

Tabela 6.1 - Atributos de produto e organização.

	Produto	Organização
Funcional x Funcional	<ul style="list-style-type: none"> - faixa de operação - robustez - massa - frequência natural do veículo - resistência - capacidade de suportar diferenças de temperaturas - capacidade de controle e atitude - capacidade de comunicação com o solo - tempo de separação de estágios 	<ul style="list-style-type: none"> - características de ensaio - capacidade de transporte - tempo de transporte
Funcional x Implementação	<ul style="list-style-type: none"> - capacidade de comunicação com o solo 	<ul style="list-style-type: none"> - área do centro de lançamento
Implementação x Funcional	<ul style="list-style-type: none"> - tamanho dos estágios - número de estágios - peso dos estágios - geometria - nível acústico - distância (interface, interestágio) - aerolasticidade 	<ul style="list-style-type: none"> - volume do estoque - custo - tempo de transporte - capacidade de descarregar - capacidade de transportar da aeronave - características de ensaio
Implementação x Implementação	<ul style="list-style-type: none"> - suporte a frequência - vibração - nível acústico - rigidez - aerolasticidade - distância - geometria - cinta de ligação - material - potência exigida - capacidade do barramento - capacidade de comando 	<ul style="list-style-type: none"> - condição da pista de aterrissagem - condição das vias para transporte - vias desobstruídas - tamanho (largura, altura dos prédios) - área do centro de lançamento (operacional)

Esses atributos servem para descrever os componentes que formam as arquiteturas derivadas para cada cenário do processo do ciclo de vida.

A Figura 6.35 ilustra a arquitetura de mais alto nível para o produto em operação.

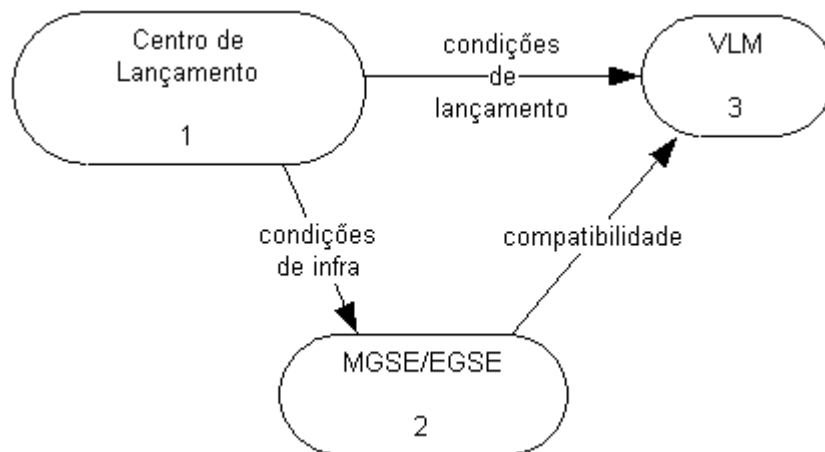


Figura 6.35 - Arquitetura de produto em operação.

A Figura 6.36 ilustra a arquitetura do produto em integração.

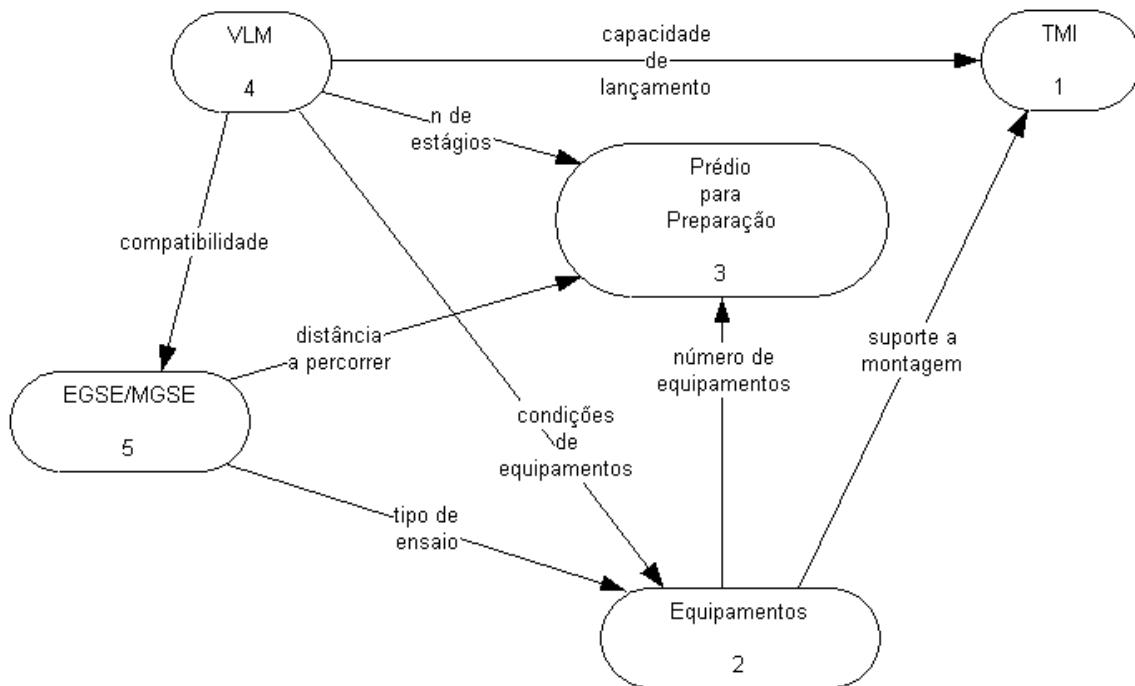


Figura 6.36 - Arquitetura do produto em integração.

A Figura 6.37 ilustra a arquitetura da organização de desenvolvimento do projeto.

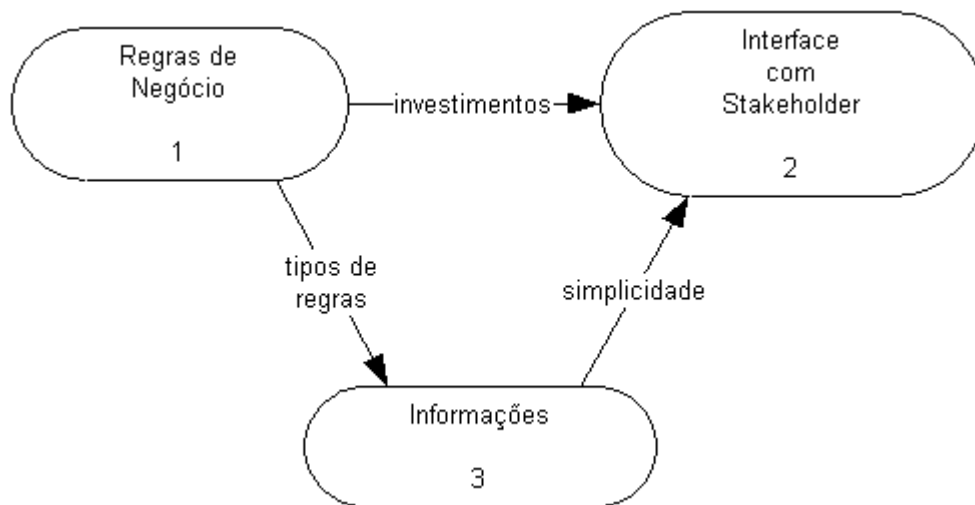


Figura 6.37 - Arquitetura da organização de desenvolvimento do projeto e construção.

A Figura 6.38 ilustra a arquitetura da organização de preparar para o lançamento.

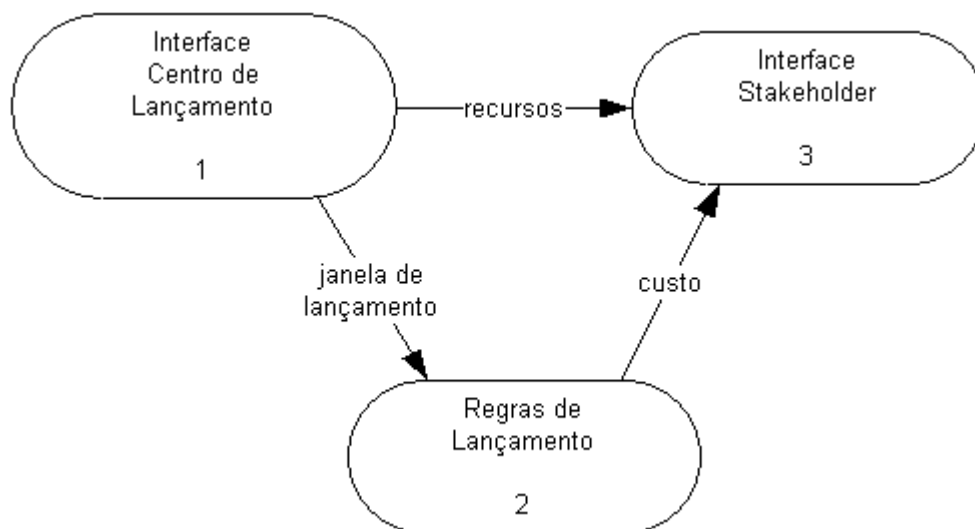


Figura 6.38 - Arquitetura da organização de preparar para lançar.

Aqui foram ilustradas as arquiteturas, derivadas dos cenários do processo de ciclo de vida e do escopo do esforço de desenvolvimento. Essas arquiteturas abrem a oportunidade de serem avaliadas no ponto de vista de integrar critérios de custo, prazo, desempenho e risco.

O exemplo do VLM foi obtido com a participação durante as reuniões do grupo de engenharia de sistemas do IAE. Esse grupo coopera para a engenharia de sistemas do VLM, e a oportunidade de utilizar uma ferramenta para o auxílio computacional para a concepção do veículo, para a captura das informações, para a criação dos requisitos e das funções auxiliou a metodologia da aplicação da engenharia de sistemas.

No caso da aplicação do *Cradle* para identificar os elementos de produto e organização durante o transporte dos propulsores no centro de lançamento, foram necessárias reuniões com o grupo para identificar e modelar as essas atividades. Sem o auxílio computacional não seria possível uma manipulação e captura de informação durante a reunião, e ao passo que cada idéia ia surgindo ou algum membro do grupo lembrava de uma situação, logo era capturada e inserida no contexto do modelo aquela informação, que mais tarde viria a ser requisitos para os MGSE's ou para a organização de desenvolvimento. Sem o auxílio computacional que atendesse a dinâmica das reuniões seria difícil capturar a diversidade de informações que são representadas por diversos tipos de modelos, que ajudam e auxiliam a todos para uma visão de todo o sistema e de seu impacto.

7 DISCUSSÃO

Este Capítulo ressalta as contribuições do trabalho comparando-o com a revisão bibliográfica e com a prática vigente.

7.1 Evolução da Engenharia Simultânea de Sistemas com e sem Auxílio computacional

Para a engenharia simultânea de sistemas são muitos os modelos de produto, por exemplo: um modelo de produto por cenário para *stakeholder*, um modelo funcional e um modelo de implementação por modo. O mesmo pode-se dizer para os modelos de organização. Neste caso são somente para as organizações dentro do escopo do esforço de desenvolvimento. A engenharia de sistemas tradicional foca, por exemplo, no produto em operação e na organização de desenvolvimento. A evolução da engenharia simultânea de sistemas expande esse conceito ao serem modelados todos os processos do ciclo de vida e seus cenários para o produto e organização.

Uma vez obtidos cada um desses modelos, atributos são derivados. Atributos funcionais para os modelos funcionais, e atributos de implementação para os modelos de implementação. Como são muitos os modelos para descrever o sistema e seu desdobramento para maiores níveis de detalhe, maior ainda é a quantidade desses atributos.

Os relacionamentos entre os atributos é que propiciam a compreensão da influência mútua entre produto e organização, viabilizando a engenharia simultânea de produto e organização. Esses relacionamentos podem ser capturados em forma de *links* de informação ou matrizes, que dado o grande número de elementos somente poder ser manipuladas com auxílio

computacional. Sem o auxílio computacional não seria possível manter a consistência entre os modelos e sua rastreabilidade desde os desdobramentos do sistema e seus itens de informação na engenharia simultânea de sistemas.

7.2 Melhoria no Desenvolvimento Integrado de Produto e Engenharia de Sistemas

Na engenharia de sistemas tradicional e na engenharia simultânea para o desenvolvimento de produtos já é necessária a engenharia e gestão de requisitos, por exemplo: para gerenciar a quantidade de requisitos de um produto complexo, rastrear requisitos de *stakeholders* a requisitos de sistema, manter a consistência entre modelos em diferentes níveis de abstração.

Dado o número de requisitos, funções e elementos de implementação, a rastreabilidade entre esses elementos somente pode ser feita por uma ferramenta computacional. Essa rastreabilidade é importante por exemplo para saber se todos os requisitos foram atendidos ou para indicar por exemplo, se uma determinada parte possui requisitos para os quais ela foi concebida.

Nos projetos com muitas pessoas podendo fazer modificações ao mesmo tempo, somente com auxílio computacional pode gerenciar as configurações resultantes.

7.3 Team X, CDF e a Engenharia Simultânea de Sistemas

As iniciativas da NASA e da ESA como o *Team x* e o CDF, realizam em mais alto nível o projeto, a partir da definição da missão, no nível de definir as trajetórias e as órbitas. Usam ferramentas para simular essas questões e seus impactos no projeto em tempo real. Eles disponibilizam de infraestrutura para a realização dessas atividades, ao passo que o processo de engenharia simultânea de sistemas está no nível de desenvolvimento do sistema.

7.4 Melhoria na Modelagem de Produto e Organização

A engenharia simultânea de sistemas usa as notações da análise estruturada e do UML para capturar os elementos do produto e da organização e descrever o sistema de forma integrada, pelos atributos derivados dos modelos funcionais e de implementação. Esses modelos são adaptados para produto e para a organização. Desta forma a engenharia simultânea de sistemas melhora a descrição dos elementos de organização que trocam energia, material e informação, suas interfaces funcionais e suas interconexões físicas.

7.5 BPMN, SysML e a Engenharia Simultânea de Sistemas

Em relação ao BPMN que cobre a visão funcional, e o SysML como UML estendido para hardware com diagrama de requisitos e diagrama paramétrico, com foco no produto, ambos tem sua cobertura com cada um focando parte de um desenvolvimento. A engenharia simultânea de sistemas assume a organização como parte da solução sistema. A engenharia simultânea de sistemas cobre a visão funcional que é a da organização e também foca no produto.

O SysML modela o sistema, porém seu foco ainda é no produto, e o BPMN está restrito somente ao conceito de modelagem aplicado a processos de negócios. Isso significa que outros tipos de modelagens realizadas por organizações para negócios estão fora do escopo do BPMN. Portanto ficam fora aspectos do tipo:

- definição de modelos de organização e recursos;
- modelagem de desdobramento funcional;
- modelos de dados e informação;
- modelagem de estratégia;
- modelos de regras de negócios.

7.6 Auxílio Computacional ao Processo de Engenharia Simultânea de Sistemas

A abordagem adotada para o auxílio computacional ao processo de engenharia simultânea de sistemas tem como contribuição a captura dos elementos do processo, desde a declaração da missão, seus processos do ciclo de vida e os processos dentro do escopo do esforço de desenvolvimento, e os elementos da análise de *stakeholders*, da análise de requisitos, da análise funcional e da análise de implementação.

O auxílio computacional ao processo de engenharia simultânea de sistemas permite ser uma ferramenta para a engenharia de atributos. Isto impacta no gerenciamento de complexidade do produto durante seu ciclo de vida. Os atributos capturados descrevem a solução do produto. Isto permite uma análise em todas as camadas do sistema, levando em consideração os atributos que precisam ser melhorados, ou os que serão impactantes ou ainda fazer um balanço deles do que precisa ser implementado no produto, ou na organização, como por exemplo, seus recursos disponíveis para desempenhar um processo de desenvolvimento.

7.7 Vantagens do *Cradle*

O *Cradle* possui a vantagem de utilizar os diagramas dentro do domínio essencial e do domínio de implementação e por ser possível criar os *links* de informação entre os elementos de produto e de organização, adaptando alguns diagramas como por exemplo, os diagramas DFD para a captura dos *stakeholders*, e os diagramas que capturam os elementos de organização. Já as ferramentas de engenharia simultânea e de engenharia de sistemas disponibilizadas no Capítulo 2 frente ao *Cradle* não permitem essa adaptação para atender a necessidade da engenharia simultânea de sistemas. Isso foi

levado em consideração ao escolher essa ferramenta para a aplicação no projeto do VLM e para o desenvolvimento deste trabalho.

- O auxílio computacional usando o *Cradle* impacta no desenvolvimento de produtos complexos. A ferramenta auxilia em ilustrar os elementos do processo. Os níveis de detalhes ao desdobrar os modelos do sistema são capturados e ficam armazenados na base de dados do *Cradle*. Desta maneira, informações do projeto podem alimentar outros projetos vindouros, e as informações não serão perdidas.
- A rastreabilidade entre os elementos de produto e de organização pode ser feita pelos *links* de informação, e os atributos associados a cada elemento pode ser monitorado. O *Cradle* permite monitorar esses itens mantendo a rastreabilidade, assim pode-se evitar retrabalho no futuro. Ao invés de usar uma planilha Excel para a captura dos atributos e para o relacionamento entre eles, no *Cradle* já ficam capturados os atributos e já podem ser associados aos elementos de produto e organização de forma a facilitar o seu monitoramento. E isto em um único ambiente, onde estão os modelos e seus elementos.
- Os exemplos do CBERS e do VLM abordados neste trabalho foram submetidos para análise de alguns especialistas, com o intuito de levantar a opinião sobre trabalho e seu impacto. As respostas obtidas foram de que “o aperfeiçoamento e detalhamento futuro do uso da ferramenta será útil no desenvolvimento de projetos espaciais”. E ainda “a ferramenta auxilia a tornar clara a captura dos elementos do processo, e a manter a rastreabilidade entre eles”.

7.8 Auxílio Computacional ao Processo de Engenharia Simultânea de Sistemas - CBERS

Para um programa como o CBERS não se pode deixar de ressaltar as potenciais contribuições que o auxílio computacional traz em seus vários níveis de aplicação. Por exemplo: o auxílio computacional permite que itens de informação possuam rastreabilidade, ou seja, requisitos podem ser rastreáveis por *links* entre os requisitos e seus métodos de verificação em uma matriz de verificação. O *link* entre os requisitos passam a possuir atributos que descrevem a razão de ser daquele requisito e porque o relacionamento com outro determinado item de informação.

O auxílio computacional ainda expande a maneira de modelar o produto e as organizações que implementam os processos do ciclo de vida dentro do escopo de desenvolvimento. Ao passo que identificam-se os recursos necessários para as atividades de AIT (*assembly, integration and tests*) montagem, integração e testes esses elementos de organização uma vez capturados e modelados por auxílio computacional passam a tornar-se claro e em um nível de linguagem que possa fluir um diálogo entre técnicos e *stakeholders*. Os modelos que representam o produto para as atividades de AIT podem servir de base para diversos projetos com várias classes de satélites por exemplo. São esses modelos reutilizáveis e podem ser interpretados em vários níveis de abstração.

Os modelos de *stakeholders* passam a ser parte fundamental do desenvolvimento do projeto, uma vez identificados e atribuídos a cada um deles as informações que servem para mais tarde serem desdobrados nas suas satisfações.

7.9 Auxílio Computacional ao Processo de Engenharia Simultânea de Sistemas - VLM

Para o projeto do VLM o auxílio computacional teve suas vantagens ao utilizar o processo de engenharia simultânea de sistemas para a sua concepção, onde foi possível identificar diversos fatores impactantes no desenvolvimento do sistema e de seu ciclo de vida. Por exemplo, o transporte de novos propulsores de material composto não poderia ser manipulado por MGSE's da mesma maneira em que se manipulam os motores de aço. A modelagem do 'transporte até TMI' permitiu a identificação dos atributos de produto e organização que foram base para um balanço entre o que se deveria fazer em relação aos recursos já existentes para atender ao transporte, e evitar retrabalho, impacto no cronograma e custos adicionais com imprevistos no futuro desenvolvimento.

Ainda durante as reuniões de projeto, foram capturados fatores impactantes do projeto em forma de itens de informação e imediatamente associados às atividades da organização, ou ao funcionamento de um equipamento específico para o desenvolvimento do produto. Ao longo da análise o diagrama era revisitado e a informação servia de realimentação para o desenvolvimento naquele instante, alterando os elementos necessários de acordo com a sugestão de cada especialista.

8 CONCLUSÃO

Este Capítulo confirma que o trabalho realizado atende os objetivos apresentados no Capítulo 1.

8.1 Objetivos Atingidos

Este trabalho apresentou como descrito no objetivo geral um software ambiente de engenharia de sistemas para o auxílio computacional a um processo de engenharia simultânea de sistemas. Sua demonstração foi sobre o exemplo do CBERS II descrito no Capítulo 5, e para o VLM descrito no Capítulo 6.

O auxílio computacional mostrou-se possível com o uso da ferramenta comercial *Cradle* para cobrir os elementos do processo de engenharia simultânea de sistemas aplicada a exemplos da área espacial, no qual este trabalho tem seu escopo.

O auxílio computacional usando o software comercial *Cradle* traz os benefícios de possuir em um único ambiente a solução sistema, cobrindo todos os elementos do processo de engenharia simultânea de sistemas. Permitiu capturar os *stakeholders* e seus interesses assim como os elementos não só de produto, mas também os de organização que também são parte da solução sistema. Utilizando a ferramenta foram ilustrados os modelos de produto e de organização dentro do domínio essencial e do domínio de implementação, que podem ser relacionados por *links* de informação, o que permite manter a rastreabilidade ao longo do desenvolvimento dos processos do ciclo de vida.

O auxílio computacional utilizando o software comercial *Cradle* do processo de engenharia simultânea de sistemas trouxe os benefícios citados pelos especialistas sobre a aplicação nos exemplos do CBERS e do VLM.

8.2 Contribuições

O processo de engenharia simultânea de sistemas possui muitos modelos e sem o auxílio computacional a esse processo não seria possível manter a integridade e a rastreabilidade entre eles.

A engenharia simultânea e a engenharia de sistemas para o desenvolvimento integrado de produtos complexos também necessitam do auxílio computacional para gerenciar os diversos requisitos, funções e para gestão da configuração.

Em comparação com as iniciativas do *Team X* e o CDF a engenharia simultânea de sistemas auxiliada por computador concebe uma solução sistema em que a organização é parte dessa solução. Enquanto *Team X* e CDF focam na missão.

Esse processo de engenharia simultânea de sistemas melhora a descrição dos elementos de organização, e inova ao adaptar os diagramas para a captura dos *stakeholders* e seus interesses, no qual vem a ser parte da concepção do sistema e da solução.

A engenharia simultânea de sistemas cobre a modelagem de organização e de produto.

O auxílio computacional para o processo de engenharia simultânea de sistemas cobre a análise de *stakeholders*, a análise de requisitos, a análise funcional e a análise de implementação.

O *Cradle* possui a vantagem de ter suas notações dentro do domínio essencial e do domínio de implementação que permitem cobrir todo o processo de engenharia de sistemas.

A aplicação da engenharia simultânea de sistemas auxiliada por computador para um programa como o do CBERS permite manter a consistência entre os muitos itens de informação que são feitos de forma manual, e a comunicação entre os especialistas nos diversos níveis da estrutura do trabalho.

A aplicação do auxílio computacional durante as reuniões de projeto do grupo de engenharia de sistemas para o desenvolvimento do VLM foi de maneira imensamente proveitosa para a captura das informações dos diversos especialistas, que propuseram soluções e idéias dado a ampla experiência de cada um no desenvolvimento de sistemas espaciais. Tal oportunidade de desenvolver um sistema utilizando o processo de engenharia simultânea de sistemas, auxiliado por computador utilizando um software ambiente de engenharia de sistemas como o *Cradle* inova a maneira de conceber um sistema, seus processos do ciclo de vida e seus GSE's (*Ground Support Equipments*) no âmbito espacial.

A realização deste trabalho tornou possível gerar um modelo de referência para o desenvolvimento integrado de produtos. Este modelo pode ser aplicado em laboratório de engenharia de sistemas para o ensino e a aplicação em exemplos da indústria.

A aplicação da abordagem de engenharia simultânea de sistemas nos exemplos do CBERS e do VLM permitiu derivar o que foi chamado de arquiteturas, realizadas para cada cenário do processo do ciclo de vida. Nelas são associados os atributos derivados dos modelos de produto e organização. Nisto foi visto que surge a oportunidade de compor a essas arquiteturas, realizadas para todos os processos do ciclo de vida, critérios de custo, prazo, desempenho e risco.

Este trabalho abre a oportunidade para as instituições desenvolvedoras dos satélites e foguetes, para levarem em conta no processo de desenvolvimento

os todos os cenários dos processos do ciclo de vida, e dos cenários dentro do escopo do esforço de desenvolvimento.

Ainda este trabalho contribui para a descrição textual do processo de engenharia simultânea de sistemas. Esse processo vem evoluindo e sendo ensinado pelo Professor Geilson Loureiro mas sem um texto descritivo. Essa dissertação oferece essa descrição.

8.3 Sugestões para Trabalhos Futuros

Como sugestões de trabalhos futuros:

- Integração de ferramentas de modelagem descritiva com modelos computáveis (e.g. MATLAB/SIMULINK);
- Aplicação do DSM sobre as matrizes de relacionamento entre atributos gerados pelo método;
- Uso do SysML como notação única de modelagem de produto e de organização;
- Geração automática de documentos;
- Aplicação do auxílio computacional ao processo de engenharia simultânea de sistemas em outras áreas;
- O uso em uma organização da engenharia simultânea de sistemas auxiliado por computador.

REFERÊNCIAS BIBLIOGRÁFICAS

ANDREASEN, M. M.; HEIN, L. **Integrated product development**. Bedford, UK: Springer-Verlag, 1987.

BOOTHROYD DEWHURST. **Design for Manufacture and Assembly (DFMA)**: 2011. Disponível em: <<http://www.dfma.com/>>. Acesso em: 23 mai. 2011.

BOUJUT, J. F.; LAUREILLARD, P. A co-operation framework for product-process integration in engineering design. **Design Studies**, v.23, p.497-513, 2002.

CALVEZ, J. P. **Embedded real-time systems: a specification and design methodology**. Chichester, England: John Wiley Sons, 1993.

COMPANY, P. et al. Computer-aided sketching as a tool to promote innovation in the new product development process. **Computers in Industry**, v.60, p. 592-603, 2009.

ELECTRONIC INDUSTRIES ALLIANCE. **EIA-632: process for engineering a system**. Arlington. 1999.

EUROPEAN COOPERATION FOR SPACE STANDARDIZATION. **ECSS-E-ST-10C: space engineering: system engineering general requirements**: Noordwijk: ESA, 2009.

EUROPEAN SPACE AGENCY – ESA. **Concurrent design facility: what is the CDF?** Disponível em: <http://www.esa.int/esaMI/CDF/SEMQOF1P4HD_0.html>. Acesso em: 18 jan. 2011.

FRIEDENTHAL, S. et al. **A practical guide to sysml: the systems modeling language**. Burlington: Elsevier, 2009.

FULINDI, J. B.; LOUREIRO. G.; COSTA, L, E, L. Concurrent systems engineering of a microsatellite launch vehicle. In: INTERNATIONAL

ASTRONAUTICAL CONGRESS, 61, Praga, 2010. **Proceedings...** Praga: IAF, 2010.

GIL, A. C. **Como elaborar projetos de pesquisa**. 3 ed. São Paulo. Atlas, 1991.

HAQUE, B.; PAWAR, K. S.; BARSON, R. J. The application of business process modelling to organisational analysis of concurrent engineering environments. **Technovation**, v.23, p.147-162, 2003.

HORVÁTH, L.; RUDAS, J. I.; HANCKE, G. P. Content orientation in integrated product modeling. In: IEEE INTERNATIONAL CONFERENCE ON INTELLIGENT ENGINEERING SYSTEMS, 11., 2007, Lahore, Pakistan. **Proceedings...** [S.l.] IEEE, 2007. p.167-172.

HUANG, G. Q. **Design for X: concurrent engineering imperatives**. London: Chapman Hall, 1996.

HUBKA, V.; EDER, W. E. **Theory of technical systems: a total concept theory for engineering design**. Berlin: Springer-Verlag, 1988.

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. **Systems engineering: application and management of the systems engineering process**: New York: IEEE, 2005.

INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING - INCOSE. **International council on systems engineering**. 2010a. Disponível em: <<http://www.incose.org/>>. Acesso em: 28 jan. 2011.

INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING - INCOSE. **Systems architecture tools survey**. Disponível em: <<http://www.incose.org/ProductsPubs/products/sysarchtools/read.php>>. Acesso em: 28 jan. 2011.

INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING - INCOSE. **Tools Database Working Group (TDWG)**. 2010b. Disponível em: <<http://www.incose.org/practice/techactivities/wg/tools/>>. Acesso em: 28 jan. 2011.

INSTITUTO DE AERONÁUTICA E ESPAÇO. **IAE**. São José dos Campos. Disponível em: <<http://www.iae.cta.br>>. Acesso em: 18 jan. 2011.

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS. **CBERS 2B ait plan**. São José dos Campos: INPE, 2006. (CB-AIT-026-2B/02).

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS. **Satélite Sino-Brasileiro de Recursos Terrestres**: histórico. Disponível em: <<http://www.cbbers.inpe.br/?content=historico>>. Acesso em: 18 jan. 2011a.

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS. **Satélite Sino-Brasileiro de Recursos Terrestres**: introdução. Disponível em: <<http://www.cbbers.inpe.br/?content=introducao>>. Acesso em: 18 jan. 2011b.

ISPE INTERNATIONAL CONFERENCE ON CONCURRENT ENGINEERING, 17, 2010, Cracow, Poland. Disponível em: <<http://ce2010.pl>>. Acesso em: 24 jan. 2011.

JET PROPULSION LABORATORY. **Team X**. Disponível em: <<http://jplteamx.jpl.nasa.gov/>>. Acesso em: 18 jan. 2011.

JURAN, M. J.; GODFREY, A. B. **Juran's quality handbook**. 5.ed. New York: McGraw Hill, 1998.

LAI, Y. L. A constrained-based system for product design and manufacturing. **Robotics and Computer-Integrated Manufacturing**, v. 25, p. 246-258, 2009.

LEON, N. The future of computer-aided innovation. **Computers in Industry**. V. 60, p. 539-550, 2009.

LOUREIRO, G. **A systems engineering and concurrent engineering framework for the integrated development of complex products**. 1999. Tese (Doutorado em Manufacturing Engineering) - Loughborough University, England, 1999.

LOUREIRO, G. Lessons learned in 12 years of space systems concurrent engineering. In: INTERNATIONAL ASTRONAUTICAL CONGRESS, 61, Praga, 2010. **Proceedings...** Praga: IAF, 2010b.

LOUREIRO, G.; LEANEY, P. G. A systems and concurrent engineering framework for the integrated development of space products. **Acta Astronautica**, p.945-961, 2003.

LOUREIRO, G. et al. Systems concurrent engineering of an electrical ground support equipment for an on-board computer. In: ISPE INTERNATIONAL CONFERENCE ON CONCURRENT ENGINEERING, 17, Cracóvia, 2010. **Proceedings...** Cracóvia: Springer-Verlag, 2010a.

LOUREIRO, G. et al. Systems concurrent engineering of a turbogenerator. In: ISPE INTERNATIONAL CONFERENCE ON CONCURRENT ENGINEERING, 17, Cracóvia, 2010. **Proceedings...** Cracóvia: Springer-Verlag, 2010b.

LOUREIRO, G. et al. Systems concurrent engineering of an electric bike. In: ISPE INTERNATIONAL CONFERENCE ON CONCURRENT ENGINEERING, 17, Cracóvia, 2010. **Proceedings...** Cracóvia: Springer-Verlag, 2010c.

LOUREIRO, G. et al. Systems concurrent engineering for the conception of a hybrid vehicle. In: ISPE INTERNATIONAL CONFERENCE ON CONCURRENT ENGINEERING, 17, Cracóvia, 2010. **Proceedings...** Cracóvia: Springer-Verlag, 2010d.

LOUREIRO, G. et al. Systems concurrent engineering to develop a green car. In: ISPE INTERNATIONAL CONFERENCE ON CONCURRENT ENGINEERING, 17, Cracóvia, 2010. **Proceedings...** Cracóvia: Springer-Verlag, 2010e.

LOUREIRO, G. **Introdução à engenharia de sistemas espaciais**. São José dos Campos: INPE, abr. 2009. Notas de aula.

LOUREIRO, G. **Introdução à engenharia de sistemas espaciais**. São José dos Campos: INPE, 2010a. Material de Aula.

MARTINS, G. A. **Manual para elaboração de monografias e dissertações**. 2 ed. São Paulo: Atlas, 2000.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION. **NASA systems engineering handbook**. Washington, 2007.

OBJECT MANAGEMENT GROUP (OMG). **Business Process Model Notation (BPMN)**: version 2.0. jan. 2011. Disponível em: <<http://www.omg.org/spec/BPMN/2.0>>. Acesso em: 28 jan. 2011a.

OBJECT MANAGEMENT GROUP (OMG). **Unified Modeling Language (UML)**: 2011. Disponível em: <<http://www.uml.org/>>. Acesso em: 28 jan. 2011b.

PRASAD, B. **Concurrent engineering fundamentals: integrated product and process organisation**. New Jersey, USA: Prentice Hall, 1996. v. 1.

SILVA, E. L.; MENEZES, E. M. **Metodologia da pesquisa e elaboração de dissertação**. 3 ed. Florianópolis, Brasil. Universidade Federal de Santa Catarina, 2001.

SOLINGEN, V. R.; BERGHOUT, E. **The goal/question/metric method: a practical guide for quality improvement of software development**. England, UK: McGraw-Hill, 1999.

STEELE, J.; SON, Y. J.; WYSK, R. A. Resource modeling for the integration of the manufacturing enterprise. **Journal of manufacturing systems**, v.19, n.6, p.407-427, 2001.

STEVENS, R. et al. **Systems engineering: coping with complexity**. London: Prentice Hall Europe, 1998.

STRUCTURED SOFTWARE SYSTEMS (3SL). **Cradle overview**. Disponível em: <<http://www.threesl.com/pages/Evaluators/startup/overview.pdf>>. Acesso em: 28 jan. 2011a.

STRUCTURED SOFTWARE SYSTEMS (3SL). **Requirements management software with Cradle**. 2011. Disponível em: <<http://www.threesl.com/index.php>>. Acesso em: 28 jan. 2011b.

STRUCTURED SOFTWARE SYSTEMS (3SL). **WorkBench user guide**. Cumbria. United Kingdom, 2009.

VERNADAT, F. B. Enterprise modeling and integration (EMI): current status and research perspectives. **Annual Reviews in Control**, v.26, p. 15-25, 2002.

VERNADAT, F. B. **Enterprise modeling and integration: principles and applications**. London: Chapman Hall, 1996.

WINNER, R. I. et al. **The role of concurrent engineering in weapon systems acquisition**. Alexandria: Institute for Defense Analysis, IDA, 1988. (Report R-338).

YOURDON, E. **Análise estruturada moderna**. Rio de Janeiro: Campus, 1990.

APÊNDICE A - FUNCIONALIDADES DO CRADLE

Neste apêndice é descrita a cobertura do ciclo de vida, o escopo da versão 6.0 do *Cradle* e seus módulos.

Cradle é um ambiente de engenharia de sistemas e de gestão de requisitos

- suporta todo o ciclo de vida de engenharia de sistemas;
- provê acesso simultâneo por vários usuários;
- provê acesso distribuído, web e não-web;
- possui escalabilidade ilimitada para qualquer tamanho de projeto;
- provê interface aberta e flexível e compatibilidade de padrão;
- integra com ferramentas existentes;
- pode ser personalizado pelo usuário sem *scripts/macros/programação*.

A.1 Cobertura do ciclo de vida

A Figura A.1 ilustra o ciclo de vida no nível de sistema suportado pelo Cradle, e os elementos que ele provê rastreabilidade.

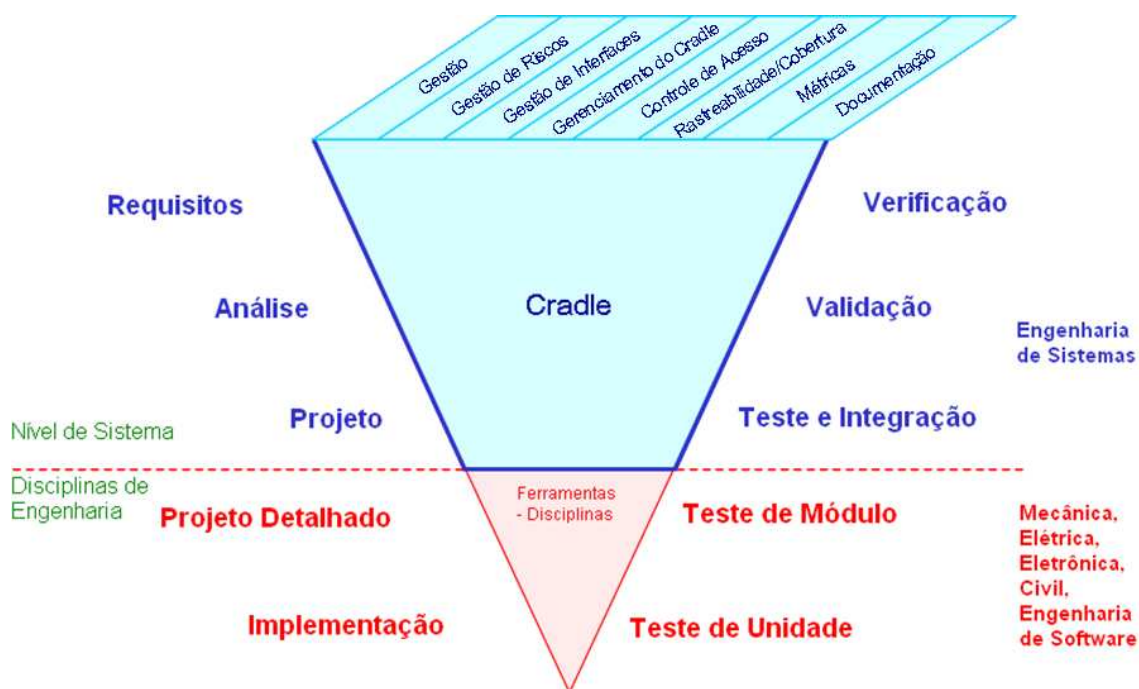


Figura A.1 - Cobertura do ciclo de vida pelo *Cradle*.

A.2 Escopo do *Cradle* versão 6.0

Estão no escopo do *Cradle* versão 6.0:

- captura e gerenciamento de requisitos, *assumptions*, justificações, etc;
- cria modelos de caso de uso, funções, classes, relacionamentos de dados;
- constrói sistema-de-sistemas, sistema, ou arquitetura de subsistemas;
- identifica e gerencia sistema / interface de sistemas;
- aloca funções para subsistemas e equipamentos;
- define desempenho, disponibilidade, confiabilidade, manutenibilidade e efetividade;
- gerencia estrutura de desdobramento de sistema, produto, trabalho e custo e associa por *links* os dados do projeto;
- desempenha a gestão de teste em todos os níveis;
- documentação de projeto automática;
- cria rastreabilidade ao longo de todo o ciclo de vida;
- captura métricas ao longo de todo o ciclo de vida;
- gerencia a configuração e o controle do projeto;
- rastreia todas mudanças formais e/ou edições individuais;
- integra ferramentas *desktop* e especialista;
- provê personalização de ambiente *web* para grupos do projeto.

A.3 Módulos do *Cradle*

O *Cradle* possui os seguintes módulos:

- gestão de dados do projeto - *Project Data Management* (PDM);
- gestão de requisitos - *Requirements Management* (REQ);
- métricas - *Metrics* (MET);
- modelagem de sistema - *System Modelling* (SYS);
- modelagem de desempenho - *Performance Modelling* (PERF);
- engenharia de software - *Software Engineering* (SWE);
- geração de documento - *Document Generator* (DOC);
- publicação web - *Web Publishing* (WEBP);
- acesso web - *Web Access* (WEBA).

A Figura A.2 ilustra os módulos do *Cradle*.

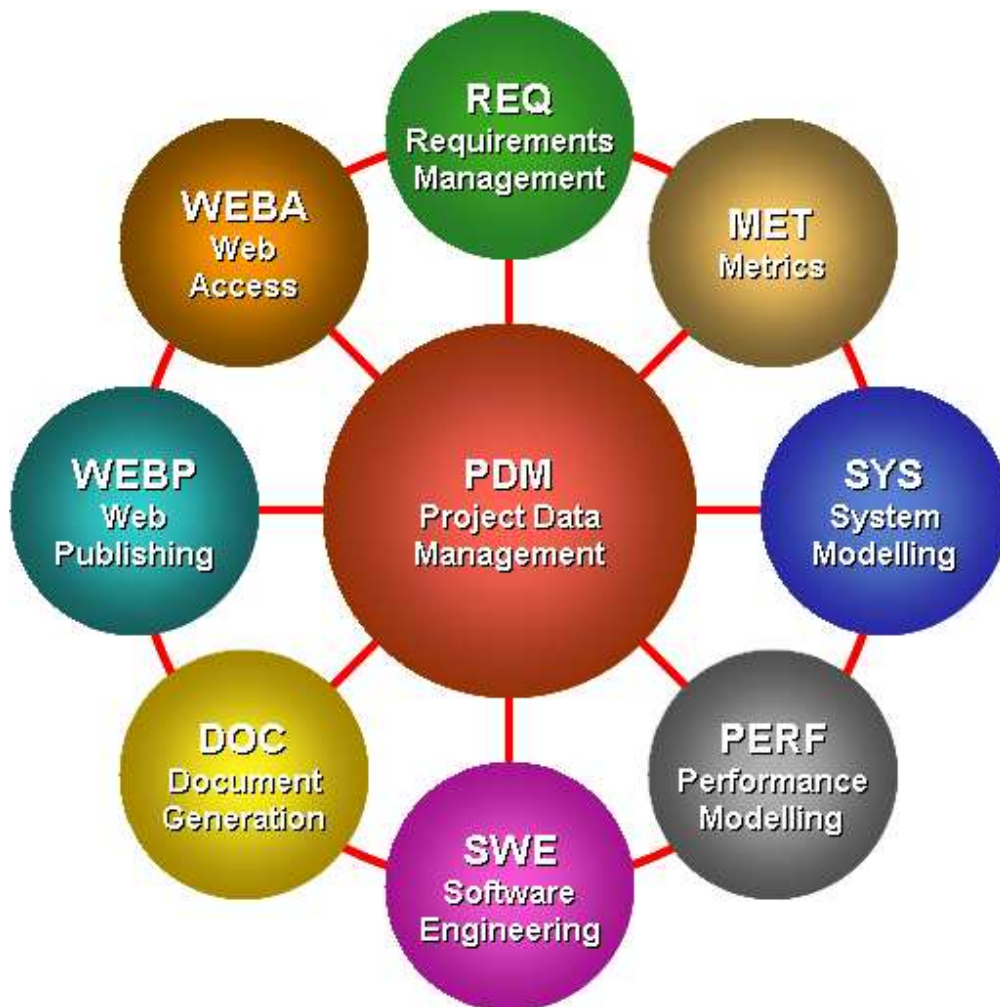


Figura A.2 - Módulos do *Cradle*.

A.3.1 PDM

Esse módulo provê a infraestrutura do projeto, todas as ferramentas usam o PDM.

Esse módulo provê:

- gerenciamento do projeto;
- ajuda web;
- referência cruzada;
- controle de acesso;
- gestão de configuração;
- alertas;
- anotações;
- interfaces e importa/exporta;
- hierarquia de fases.

A.3.1.1 Referência Cruzada

Pelas referências cruzadas é possível associar itens de informação, e manter a rastreabilidade. As referências cruzadas possuem dependências entre *links* para a associação dos itens de informação.

As referências cruzadas possuem:

- tipos de *links* para diferentes tipos de dependências;
- grupo de *links*;
- atributos de *links*, para parametrizar, justificar, caracterizar e explicar as referências cruzadas (*links*);
- regras de *links* definem o que pode ser associado, quem pode criar/modificar/deletar os *links*;
- associação de *links* de maneira unidirecional, bidirecional e indireta;
- associação de itens 1:1, 1:N ou M:N.

A Figura A.3 ilustra um exemplo de itens associados por *links*.

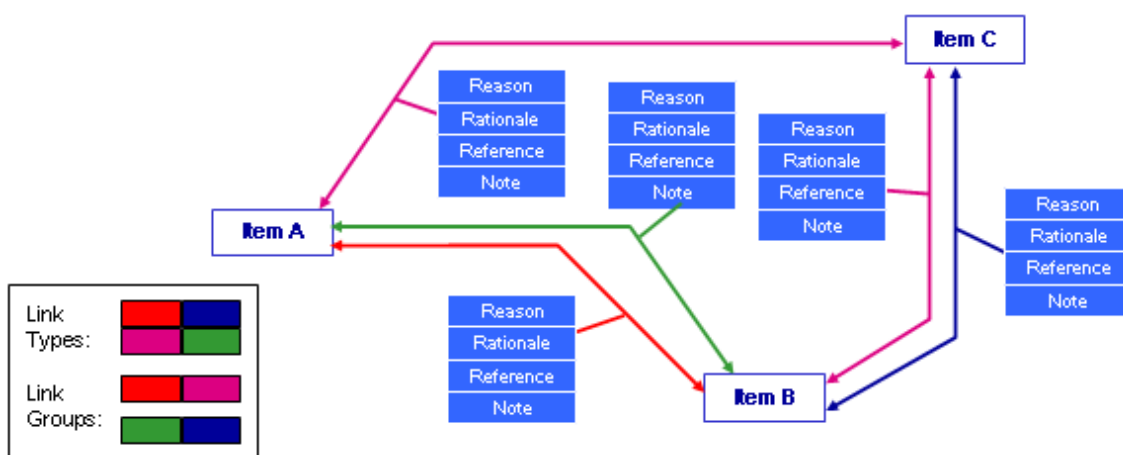


Figura A.3 - Itens associados por *links*.

As referências cruzadas (*links*) são usadas para:

- análise de impacto;
- análise de rastreabilidade;
- análise de cobertura;
- propagação de mudanças.

Isso é disponível no *Cradle* em forma de:

- listas;
- tabelas;

- árvores (hierarquias de exploração);
- matrizes;
- diagramas de hierarquia.

A Figura A.4 ilustra um exemplo de árvore, diagramas de hierarquia e de tabelas.

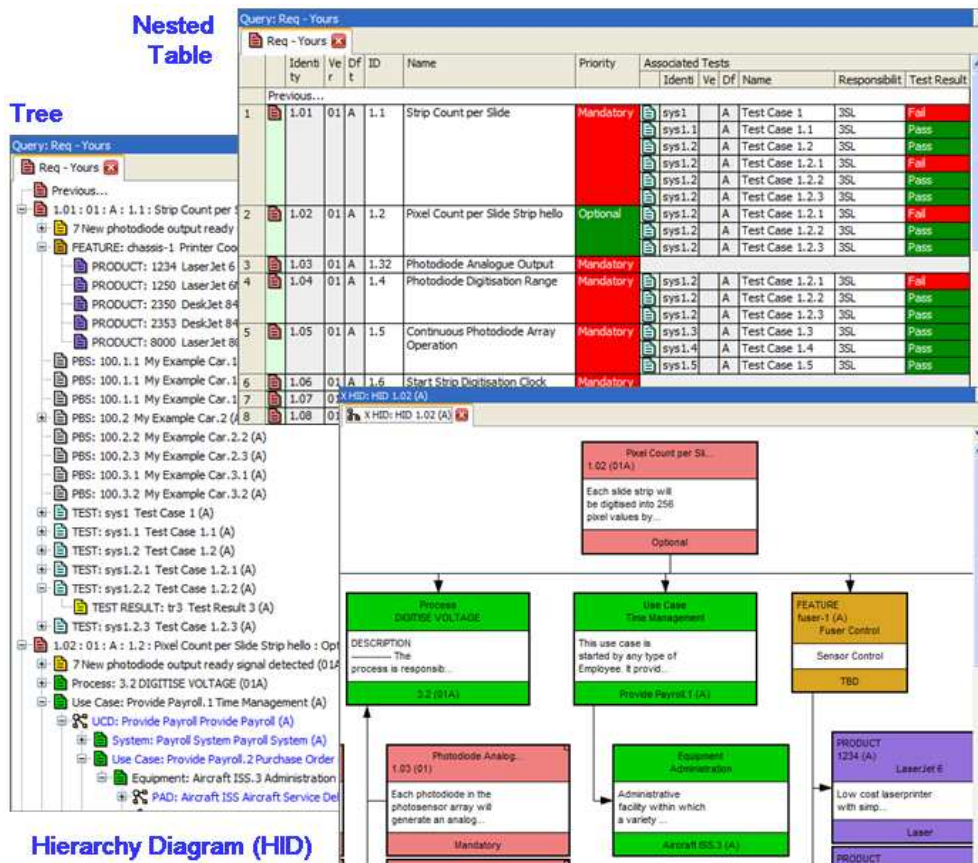


Figura A.4 - Exemplo de árvore, diagramas de hierarquia e de tabelas.

A.3.2 REQ

O módulo REQ permite a captura e a engenharia de requisitos, assim como seu gerenciamento.

A gestão de requisitos no *Cradle* permite:

- definir grupos (de negócios, operacional, usuário, sistema ...);
- capturar os requisitos nesses grupos;

- formatar requisitos com atributos (tipos, prioridade ...);
- prover argumentos para satisfação e justificação;
- fazer a engenharia e eliminar problemas;
- evoluir para o próximo grupo de requisitos (stakeholder requisitos para requisitos de sistema);
- associar *links* dos grupos;
- justificar a evolução;
- mapear o resto do processo;

A Figura A.5 ilustra a captura de requisitos pelo *Cradle*:

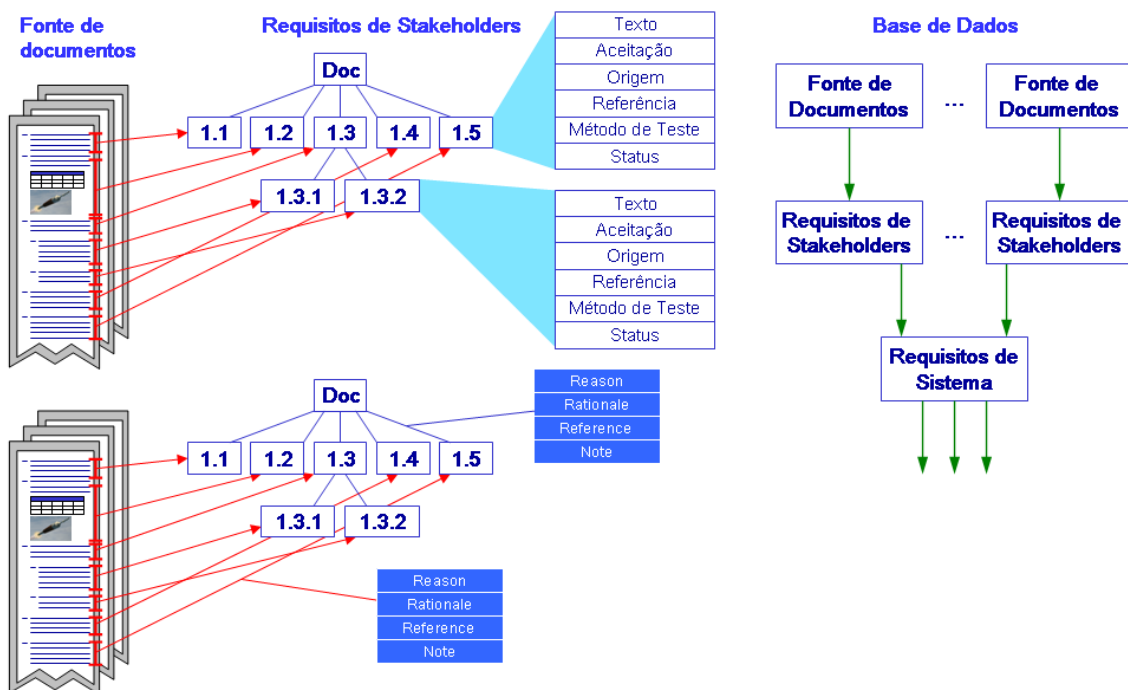


Figura A.5 - Captura de requisitos usando *Cradle*.

O *Cradle* permite a rastreabilidade ao longo do ciclo de vida por:

- modelos de análise (caso de uso, classes, funções, processos ...);
- arquitetura;
- comportamento/alocação de função;

Do desenvolvimento do projeto a rastreabilidade se estende por:

- configurações de sistema;
- descrição dos subsistemas;
- descrição dos equipamentos;
- descrição das interfaces;
- especificações de software e hardware;
- descrições de teste e especificações;
- fontes de software, módulos de códigos;

- requisitos funcionais associados por *link* ao modelo de análise;
- modelos de análise são alocados nas arquiteturas.

A Figura A.6 ilustra elementos rastreáveis desde requisitos aos modelos do projeto.

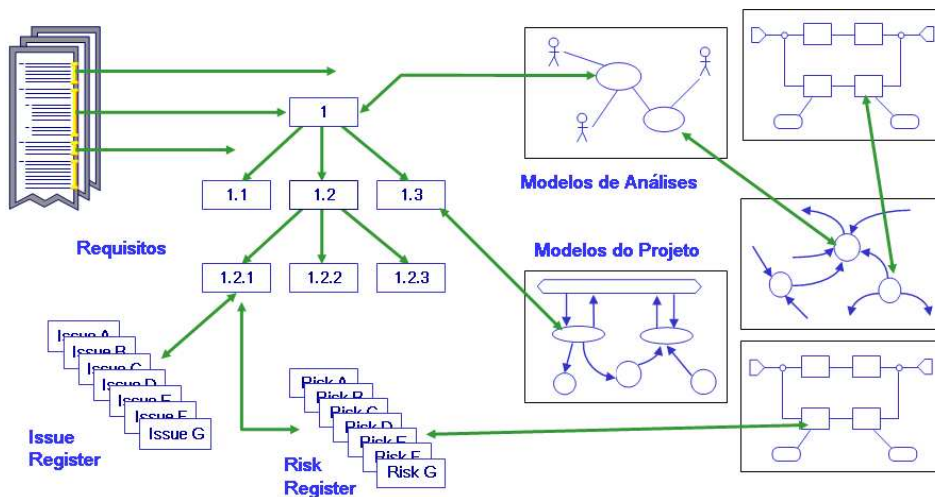


Figura A.6 - Rastreabilidade desde requisitos ao modelo de implementação

A.3.3 MET

Esse módulo permite criar e executar métricas ao longo do ciclo de vida.

O usuário define métricas do tipo:

- números de elementos;
- e escolhe uma tabela para habilitar os cálculos.

O módulo de métricas suporta:

- contas;
- análises de cobertura;

Exporta tabelas em:

- Word;
- Excel;
- HTML;
- CSV.

A.3.4 SYS

Esse módulo permite criar as análises, arquitetura e os modelos de implementação.

As características desse módulo são:

- metodologia independente;
- notações integradas de arquitetura, processo, funcional e UML;
- domínios de análise distinta e modelagem de projeto
 - essencial - análise independente de implementação
 - implementação - projeto dependente de implementação
- reuso e compartilhamento entre modelos.

A.3.4.1 Modelagem de análise

A Figura A.7 ilustra que é possível modelar diversas análises em diferentes níveis de abstração. O pacote de notações para modelagem do *Cradle* possui vários diagramas para a modelagem de análises.

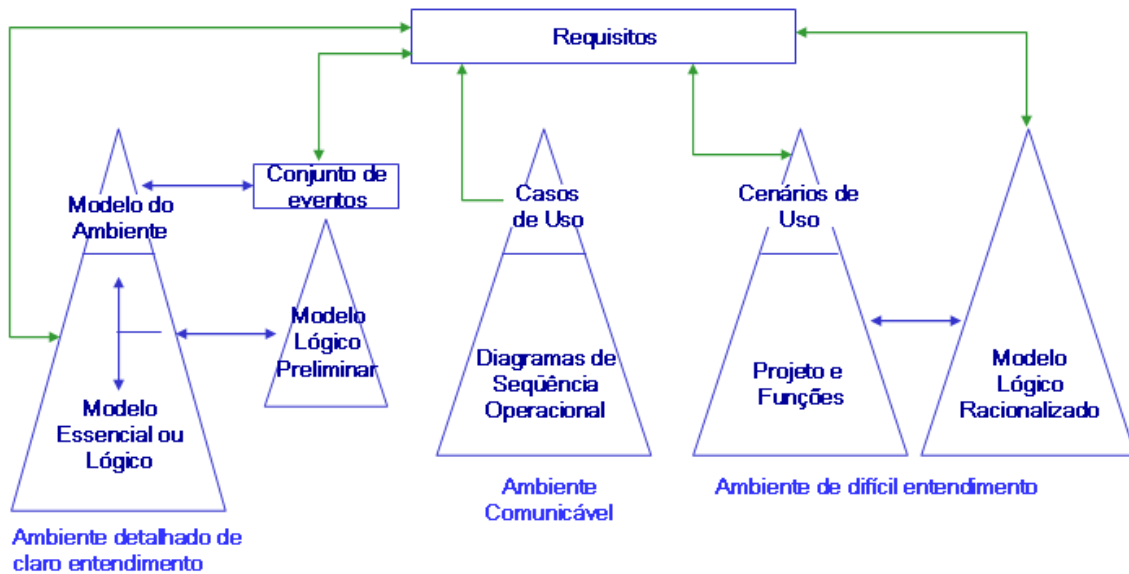


Figura A.7 - Modelagem de análise em diferentes níveis de abstração.

A.3.4.2 Contexto de Modelagem de Projeto

A Figura A.8 ilustra que também é possível modelar o contexto do projeto, e com isso alocar os elementos dos modelos de análise aos modelos de arquitetura.

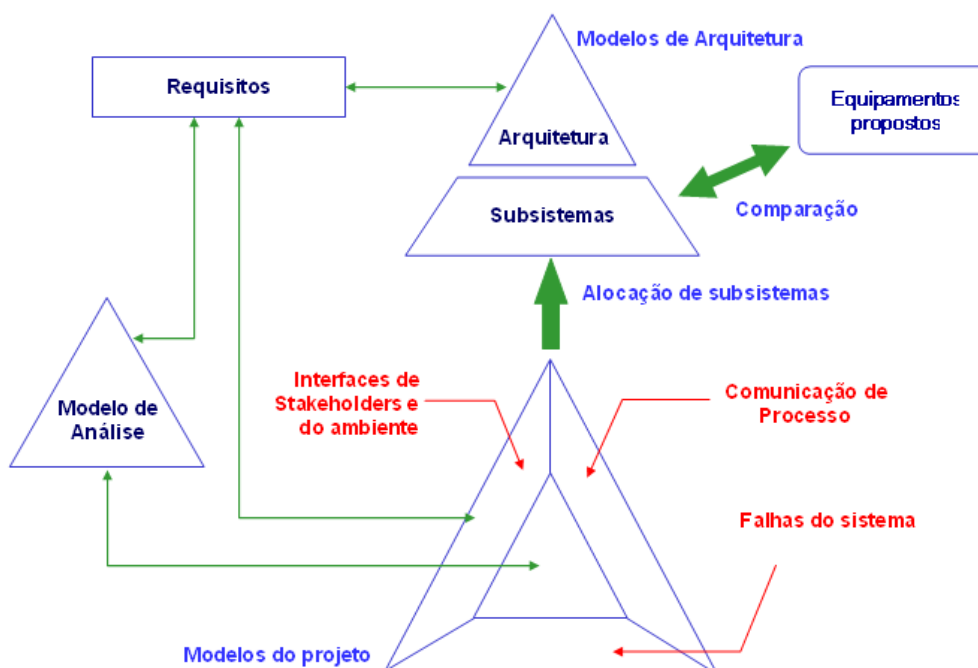


Figura A.8 - Modelagem do contexto do projeto.

A.3.5 PERF

Esse módulo permite calcular as restrições de desempenho para os vários níveis do modelo do projeto.

Esse módulo se aplica para:

- validar arquitetura de sistema;
- validar arquitetura de subsistema;
- identificar limites de desempenho do sistema;
- investigar condições de falhas.

A Análise de alto nível da arquitetura, ou dos modelos do projeto para definir as restrições para o próximo nível do projeto é:

- aplicável no nível de sistema para subsistema;
- do nível de subsistema para equipamentos.

Os modelos de desempenho medem:

- quão rápido / preciso
 - quantos dados
 - quanta energia / espaço / peso
 - quanto dinheiro
- e definem os parâmetros de desempenho.

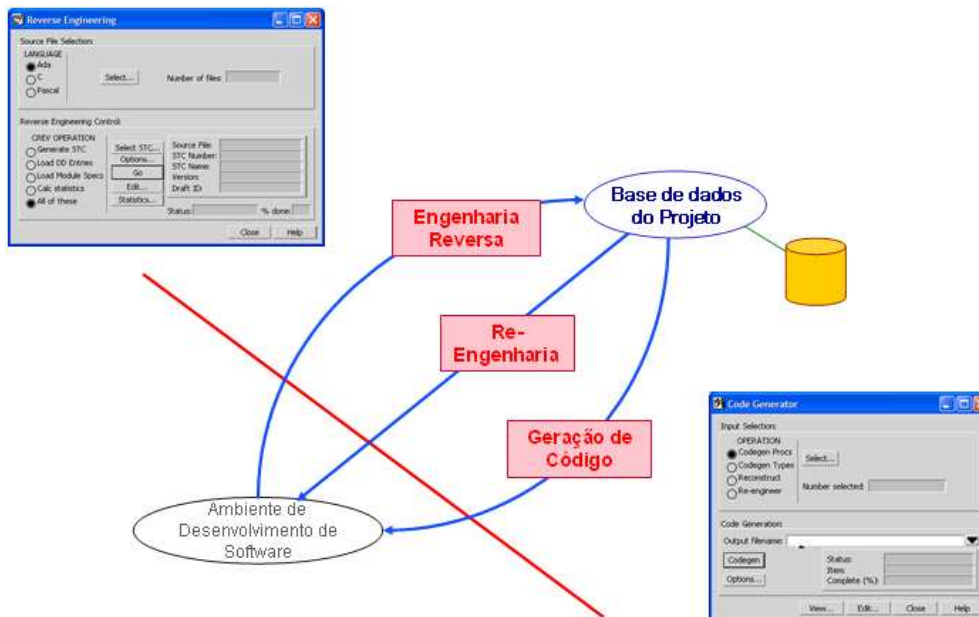
Parâmetros de desempenho possuem características como:

- atraso, latência, utilização
- consumo de energia, custo

E podem ser associados por *links* de requisitos de desempenho para o projeto, do tipo restrição em parâmetros de desempenho. Isso auxilia na análise para avaliar a viabilidade do sistema de acordo com as restrições do projeto.

A.3.6 SWE

O módulo SWE permite manter consistência entre o projeto detalhado e o software.



Características desse módulo:

Geração de código:

- suporte à C, Ada e Pascal;
- gera código procedural;
- reconstrói arquivo fonte a partir de engenharia reversa;

- incorpora mudanças de projeto em novos arquivos fontes;
- manutenção de código automatizada.

Engenharia reversa:

- diagramas de organização de código a partir do código fonte;
- preenche definições de dados a partir do código fonte;
- carrega especificações de módulo a partir do código;
- projeto para reuso;
- recupera código;
- sincronização do projeto.

Suporte de editor de diagrama:

- converte diagramas de projeto para organização de código
- converte organização de código para diagramas de projeto
- associa *link* aos arquivos fontes a partir dos diagramas de organização de código

A.3.7 DOC

O módulo DOC permite gerar documentos.

A ferramenta permite reportar tabelas e matrizes.

- cria hierarquia de descrições de documentos e *template* de páginas;
- gera documento a partir do pacote de ferramentas e arquivos para impressão.

A.3.8 WEBP

O módulo WEBP permite publicar seções de uma base de dados para *hyperlinked website*.

- publica itens para website estático
- útil para
 - Intranets
 - cd, dvd e outras mídias

Podem ser compartilhados:

- dentro do projeto
- com clientes
- com subcontrados

Pode ser uma alternativa para múltiplos documentos como:

- dados navegáveis para leitura de uma forma apropriada

Pode ser publicado qualquer tipo de item em HTML personalizável, e as referências cruzadas na base de dados tornam-se Hiperlinks.

A.3.9 WEBA

O módulo WEBA provê leitura e escrita por acesso web aos dados do projeto.

Esse módulo conecta navegadores aos dados do projeto. Os navegados são:

- Firefox, Internet Explorer, Mozilla, Opera e Safári

Suporta:

- HTTP e HTTPS
- e oferece navegadores / abas / editores

Baseado em *templates* HTML:

- diferentes interfaces de usuários para cada tipo de usuário
- interfaces de usuários personalizados e orientados a tarefas

ANEXO A - CONCURRENT ENGINEERING OF AN ELECTRICAL GROUND SUPPORT EQUIPMENT FOR A SATELLITE ON-BOARD COMPUTER SYSTEM

O trabalho aqui em anexo foi apresentado e publicado nos anais do 17th ISPE *International Conference on Concurrent Engineering* (ISPE, 2010), e recebeu indicação para ser submetido à revista *Advanced Engineering Informatics*, no qual está sob análise.

Concurrent Engineering of an Electrical Ground Support Equipment for a Satellite On-Board Computer System

Jonas Bianchini Fulindi^a, Geilson Loureiro^{b,1}, Alessandro Gerlinger Romero^c, Fabrício de Novaes Kucinskis^d, Carlos Eduardo Andrade Lemonge^e, Renan Fernandes Vazquez^f, Magda Aparecida Silverio Miyashiro^g

^bTechnologist and Professor at the Integration and Testing Laboratory, Brazilian Institute for Space Research, INPE (São José dos Campos), Brazil.

^{a,c,d,e,f,g}Post graduate students at Brazilian Institute for Space Research.

Abstract. This paper is an extended and more detailed version of a paper presented at CE2010, the 17th ISPE (International Society of Productivity Enhancement) International Conference on Concurrent Engineering. It presents a systems concurrent engineering approach for the development of an electrical ground support equipment (EGSE) for a satellite on-board computer. Traditional approaches focuses on the product, development organization and the product concepts of operation (CONOPS). In those approaches the overall view of the inherent complexity in the development of a product, its life cycle processes and their performing organizations are not taken into consideration. The systems concurrent engineering performs stakeholder analysis, requirements analysis, functional analysis and implementation architecture analysis, simultaneously, for the product, its life cycle processes and their performing organization. From the analysis, requirements and attributes are captured for the product and its life cycle processes organizations and the relationship among them are identified. Conclusions are that impact, traceability and hierarchy links promote the anticipation of life cycle process requirements to the early stages of systems architecting. Late changes are avoided, development costs are dramatically reduced while satisfaction of stakeholders over product life cycle is increased.

¹ Technologist and Professor at the Integration and Testing Laboratory, Brazilian Institute for Space Research, Av. dos Astronautas 1758, São José dos Campos, Brasil; 12227-010; Tel: +55 (12) 39456317; Fax: +55 (12) 39411884; Email: geilson@lit.inpe.br

Keywords. Systems concurrent engineering, systems engineering, concurrent engineering, complex product, integrated product development.

1 Introduction

All equipment to be used in the space segment should pass through a series of environmental and functional tests, in different development steps of a satellite. For carrying out functional tests, it is necessary to use an auxiliary equipment called Electrical Ground Support Equipment – EGSE. The EGSE emulates input and output loads to a given equipment under test.

For a satellite On-Board Computer (OBC) testing, all telecommands and telemetries, all signals from and to the OBC must be emulated and monitored by an EGSE. The OBC EGSE emulates of the OBC satellite in orbit operation, during satellite assembly, integration and testing (AIT) activities providing a reference to the expected behaviour of the OBC under test.

This paper aims to present a systems concurrent engineering approach for the development of an OBC EGSE. The approach is different from traditional systems engineering approach because it anticipates to the early stages of system architecting the product life cycle process requirements. It proposes to simultaneously develop, from the outset, the product and its life cycle processes performing organizations. This paper is an extended version of the paper presented at CE2010 in Krakow, Poland [9].

The paper is organized as following: Section 2 presents the traditional systems engineering and concurrent engineering approaches. Section 3 presents the systems concurrent engineering approach framework and method. Section 4 presents the models derived for the OBC EGSE using the approach. Section 5 discusses the advantages and opportunities for improving the proposed approach. Section 6 concludes this paper.

2 Traditional systems engineering and concurrent engineering

Space products are complex [16]. They are multidisciplinary products, they must cope with extreme environmental conditions over their life cycle (vibration, temperature ranging from -196Celsius to 150 Celsius in vacuum), they must undergo very strict assembly, integration and testing (AIT) procedures. AIT organizations are worth the order of hundred million dollars. Two ton satellites may take the order of 18 months just for the AIT process. There are many opportunities to improve productivity over satellite life cycle if a concurrent engineering approach takes place from the beginning of the satellite architecting stage.

Traditional systems engineering approaches do not provide an overall view of the system during its various life cycle processes. They focus on an operational product development starting from product concept of operations. They also focus on the development organization that must be put in place in order to assure that the product meets its operational requirements [2,3,7,15]. A product has life cycle processes other than operations and it must be recognized from the outset in order to promote gains in productivity in the product development organization, by the avoidance of late changes, and in other product life cycle process organizations, as the product will be developed taking into consideration their requirements. Life cycle process organizations themselves can be developed simultaneously to product development, when they are part of the scope of the whole product development effort.

For example the NASA systems engineering handbook [15] states that systems engineering focuses in the development and the realization of a final product. Modern commercial standards, such as EIA 632 [2], state that systems engineering focuses on the operations product and on capturing requirements for the other product life cycle processes. In other words, these requirements are captured not to impact product development. The product will be systems engineered with operations in mind. When its architecture (and maybe detailed design) is defined, then life cycle processes requirements are captured to be implemented in life cycle process performing organizations. This paper proposes a method to take into consideration the impact of these organizations on the product during the product architecting process.

Conceptually, concurrent engineering acknowledges benefits of anticipating life cycle process requirements to the early stages of product development. For space products, these early stages are the system architecting phases. A systems approach requires life cycle process requirements to be balanced in the beginning of the product development process. Concurrent engineering, however, in practice, treats life cycle processes separately and optimizes product design seeking each life cycle process productivity

increase. For example, DFA optimizes for assemblability, QFD, for customer satisfaction, DFI, for inspectability, and so on. Also, concurrent engineering is, in practice, applied to parts design and not to systems composed of many integrated parts [6]. This paper proposes how the concurrent engineering concept can be used for systems engineering.

3 The systems concurrent engineering approach

Hitchins [5] states that complexity can be understood by what he calls complexity factors. They are variety, connectedness and disorder. Variety accounts for the number of different elements you have in a set. Regarding products, variety refers, for example, to the number of different parts a product may have, number of different functions it accomplishes, number of different requirements categories it is supposed to meet, number of different stakeholders it should satisfy. Connectedness refers to the relationships among elements. For example, how parts interact, how functions affect one another, how requirements conflict to each other, how value flow among stakeholders. Disorder refers to the level of tangling of those relationships. For example, is there a structure pattern of deploying stakeholder requirements through functional concept up to implementation architecture?

Figure 1 presents a framework to address complexity in product development – the total view framework evolved from Loureiro [8]. It has three dimensions. Each dimension addresses one of the complexity factors mentioned above. The analysis dimension addresses the variety factor. Along the analysis dimension, it is deployed what must be analysed in order to develop a complex product. A systems engineering process consists of stakeholder analysis, requirements analysis, functional analysis and implementation or physical analysis. The integration dimension addresses the connectedness factor. It defines what must be integrated along an integrated product development process: product elements and organization elements. Organization here refers to the organizations that perform product life cycle processes. Product elements and organization elements are the system elements. The structure dimension addresses the disorder factor. According to Alexander [1] all structures evolve into a hierarchy. System breakdown structures are also represented in hierarchies.

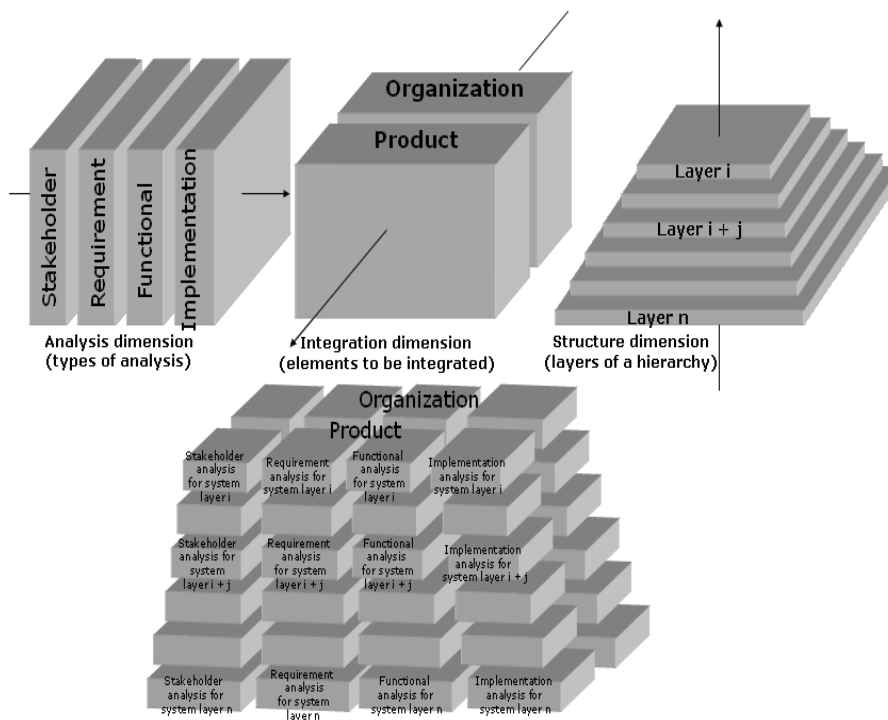


Figure 1. A framework to address complexity in complex product development – the total view framework

Figure 2 provides an overview of a method within the total view framework. The method is called concurrent structured analysis method evolved from Loureiro [8]. Stakeholder analysis, requirements analysis, functional analysis and implementation (or physical) analysis is performed, simultaneously, for the product under development and its life cycle process performing organizations. The analysis processes are performed at each layer of the system breakdown structure. For example, if a car is the product under development, the analysis processes are performed at the car layer, at the powertrain layer, at the engine layer and so on.

The total view framework and the concurrent structure analysis method was applied on around 200 examples, what derives lessons learned [14]. Some examples of the implementation of such lessons are: a turbo-generator, an electric bike, the conception of a hybrid vehicle, the development of a green car and the conception of a microsatellite launch vehicle [10,11,12,13,4].

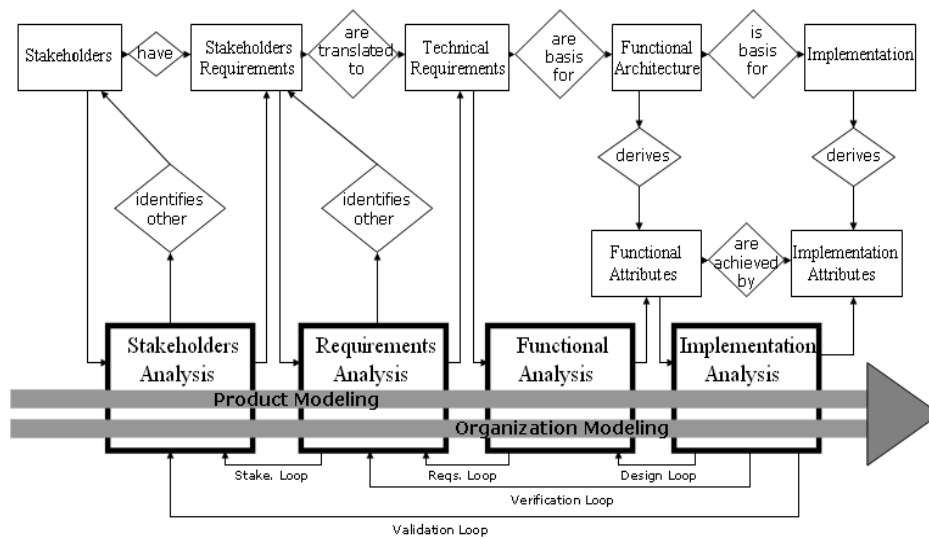


Figure 2. A method within the total view framework – the concurrent structured analysis method

Figure 3 details the concurrent structured analysis method showing how to incorporate the concurrent engineering concept in the systems engineering process [14]:

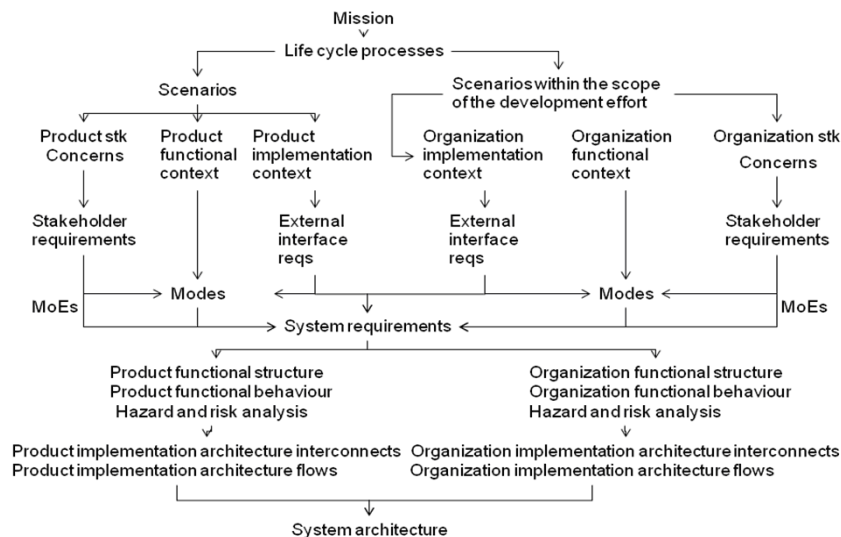


Figure 3. The system concurrent engineering method in detail

Step 1: Identify the product mission, the product life cycle processes and their scenarios and, the scope of the development effort. Product mission refers to the product purpose or reason of being. Life cycle process scenarios are the alternatives in each process (for example, preventive or corrective maintenance) or the decomposition of a process (for example, advanced technology development, process engineering as components of the development process). The scope of the development effort consists of the life cycle processes or their scenarios that the development organization is also responsible for accomplishing. For example, EMBRAER is responsible for developing aircraft but is also responsible for providing maintenance services.

Step 2: Identify product stakeholders and their concerns for each product life cycle process scenario. Product stakeholders are the people who affect or are affected by the product during its life cycle. Product stakeholders are identified per life cycle process scenario. Identify organization stakeholders and their concerns for each process within the scope of the development effort. Organization stakeholders are the people who affect or are affected by the business of the organization in question. Organization stakeholders are identified per life cycle process scenario within the scope of the development effort. From stakeholder concerns, stakeholder requirements are identified and measures of effectiveness (MoEs) are derived. MoEs must measure how the system meets the stakeholder requirements. From stakeholder requirements, functions, performance and conditions are identified. The definition of what functions the system will perform, how well the system is going to perform such functions and under which conditions comprise the requirements analysis process. Requirement analysis transforms stakeholder requirements into system requirements. System requirements will be met not only by product elements but also by organization elements.

Step 3: Identify functional context for product at each life cycle process scenario and for organization at each life cycle process scenario within the scope of the development effort. Functional context defines the function performed by the system element and identifies the elements in the environment of the system. The environment of the system contains the elements outside the system function scope and that exchanges material, information and energy flows with the system. Those flows define logical interface requirements. Environment elements may have different relevant states. Sets of environment element states are called circumstances. The system must have different modes depending on the circumstances. Behaviour modelling is required to show under which conditions system mode and system state transition occurs. Functions are identified per mode. Functions are identified from outside in by identifying which responses the system is supposed to give to deal with each stimulus provided by the environment elements. For each function, performance requirements are identified. Circumstances, flows between the system and the environment and function failures are sources of hazards. Risk analysis is performed on each identified potential hazard and exception handling functions are also identified at this stage.

Step 4: Identify implementation architecture context for product at each life cycle process scenario and for organization at each life cycle process scenario within the scope of the development effort. Physical connections between the system and the environment elements define the physical external interface requirements. Physical parts are identified. Physical internal interfaces are defined by architecture connections and architecture flows among those parts. Allocation matrix relates physical parts and physical interfaces to the functions and functional flows.

4 The OBC EGSE system concurrent engineering

This section illustrates the steps listed in Section 3 highlighting where the proposed approach is different from traditional approaches. The proposed approach is stakeholder driven whereas traditional approaches are customer or user driven. In the various steps listed in Section 3, analysis are performed for each life cycle process scenario, simultaneously, for product and organization. Traditional approaches focus on product operation and development organization.

Figure 4 presents the life cycle processes and scenarios of a satellite OBC EGSE. Development process is decomposed in scenarios presented in boxes 15, 16 and 17 in Figure 4. The processes highlighted in grey in Figure 4 are the ones for which the stakeholder analysis, requirements analysis, functional analysis and implementation architecture analysis will be exemplified. In practice steps 1 to 4

in Section 3 must be run for all life cycle process scenarios. Figures 6 to 30 just exemplify the steps for some selected processes.

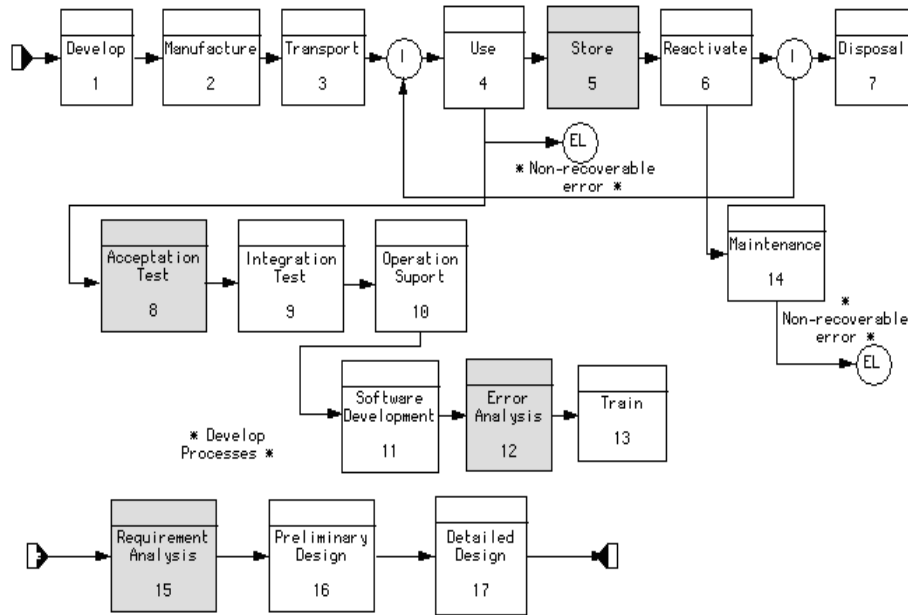


Figure 4. Life cycle processes and scenarios

Figure 5 presents the life cycle processes that are within the scope of development effort. The organization that develops the OBC EGSE is also responsible for the use, storage, reactivation and disposal of the OBC EGSE.

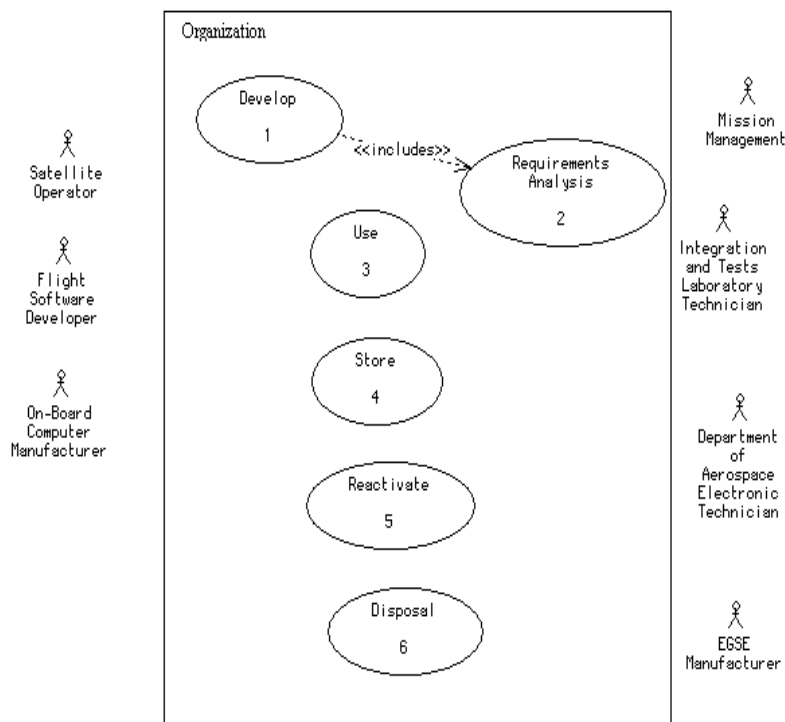


Figure 5. Scope of the development effort and organization stakeholder identification

Figures 6 and 7 exemplify the identification of organization stakeholders for two life cycle process scenarios: 'requirements analysis' and 'store'. The 'requirements analysis' scenario belongs to the development life cycle process and the 'store' process is not a product development process but is within the scope of the development effort. This is to show that it is necessary and possible to develop from the outset all processes within the scope of development effort, even though not all of them are product development processes. This innovates the traditional focus on systems engineering the product. This approach recognizes that the system solution is not only made of product elements but also of organization elements. Figures 6 and 7 also capture the stakeholder concerns represented by the connections between the stakeholders and the central bubble (containing life cycle process or scenario).

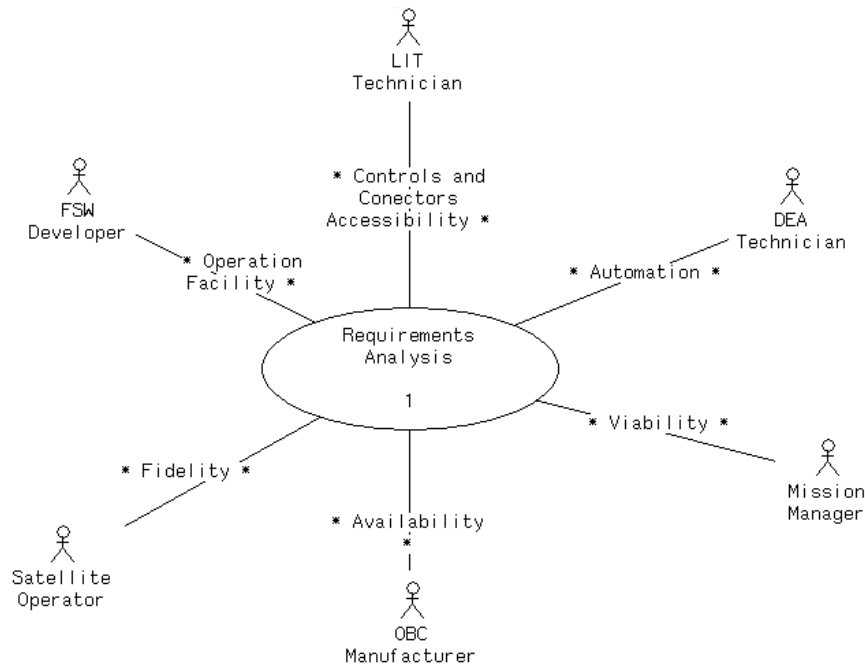


Figure 6. Organization stakeholders and their *concerns* for the requirements analysis scenario

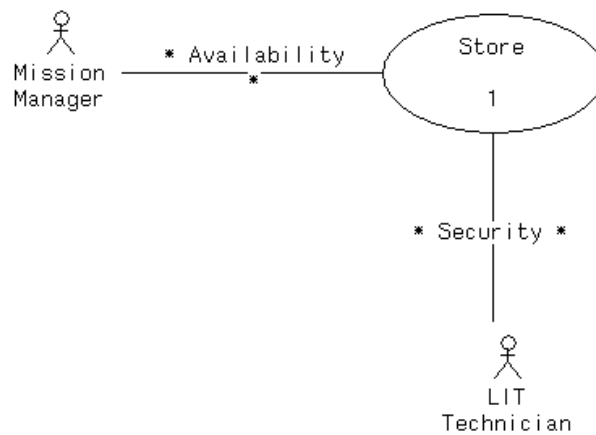


Figure 7. Organization stakeholders and their concerns for the 'store' life cycle process

Figures 8 and 9 presents the product stakeholders identified and their concerns for two other life cycle process scenarios (OBC acceptance test and error analysis). OBC acceptance tests are one of the use scenarios of the EGSE. Error analysis is one of the scenarios of operations support.

From stakeholder concerns, stakeholder requirements are identified and measures of effectiveness (MoEs) are derived. From stakeholder requirements, functions, performance and conditions are identified. Requirement analysis transforms stakeholder requirements into system requirements. System requirements will be met not only by product elements but also by organization elements.

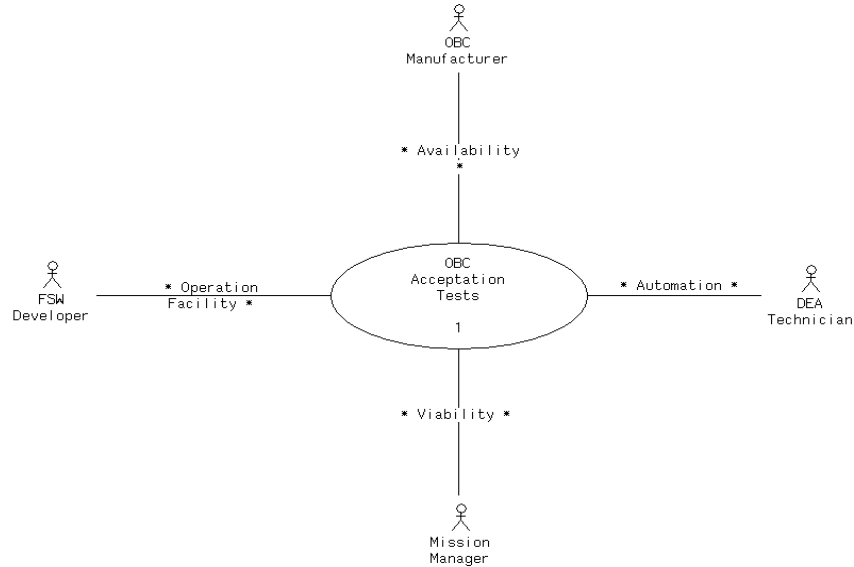


Figure 8. Product stakeholders and their concerns for the ‘OBC acceptance test’ scenario

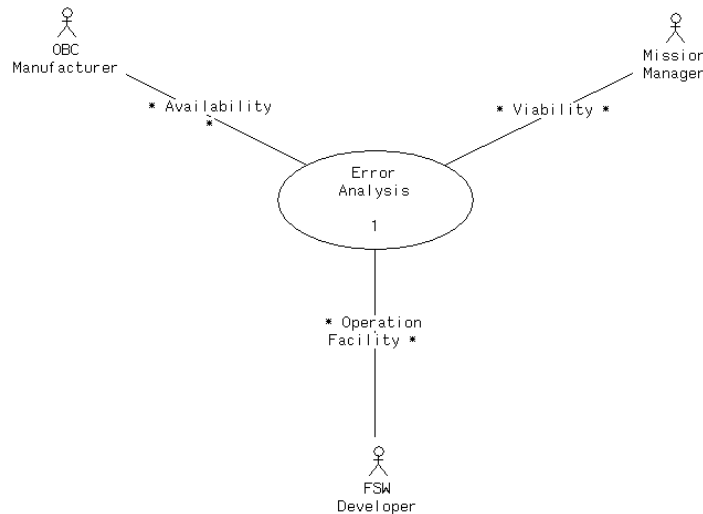


Figure 9. Product stakeholders and their concerns for the ‘error analysis’ process scenario

From stakeholder concerns, stakeholder requirements are identified and measures of effectiveness (MoEs) are derived.

ID	Stakeholder	Stakeholder requirements
EGSE-STK-001	Satellite operator	"I need that all TCs possible combination be transmitted and its results analyzed."
EGSE-STK-003	FSW developer	"I want operating the OBC through this equipment to test the FSW."
EGSE-STK-006	OBC developer	"I need that EGSE to be ready before the OBC."

Figure 10. Stakeholder requirements

The MoEs measure how the system meets the stakeholder requirements.

Stakeholder	Concern	Metrics	Measure
Satellite operator	Fidelity	Manual of operation	Similarity with the satellite operation system
FSW developers	Ease of operation	Effort to the operation	Time for the composition and TCs transmission and access to TM
OBC developers	Availability	Completed product	EGSE delivery date
EGSE developers	Stability of the requirements	Requirements history	Modification number in the requirements
DEA technicians	Automation	Testing procedures	Operation number by test item
LIT technicians	Accessibility of the connectors and controls	Effort to the connection and command	Time to make the connection and the control
Mission manager	Feasibility	Cost and time	Budget and schedule

Figure 11. Measures of effectiveness

Requirement analysis transforms stakeholder requirements into system requirements.

ID	Requirement description	Interest	Type	PO	Verif
EGSE-SIS-005	The organization shall deliver the equipment on 15/10/2009 according to the agreement contract.	Availability	C	Org.	I

Type: Functional, Performance, Constraint

PO: Product, Organization

Verif: Test, Inspection, Demonstration

Figure 12. System requirement

Figures 13 and 14 depict the organization functional context for two life cycle process scenarios: ‘requirements analysis’ (a development scenario) and ‘store’ (a non-development scenario). In Figures 13 and 14, the links between the ‘central bubble’ and the elements in the organization environment at that scenario are identified. These links show the flows of information (in this case) between the environment and the system.

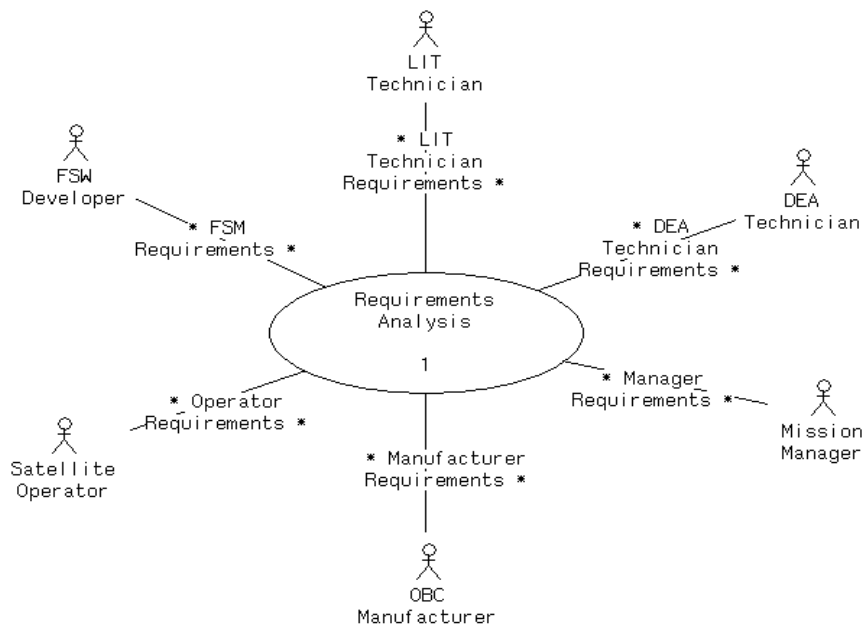


Figure 13. Organization functional context for the 'requirements analysis' process scenario

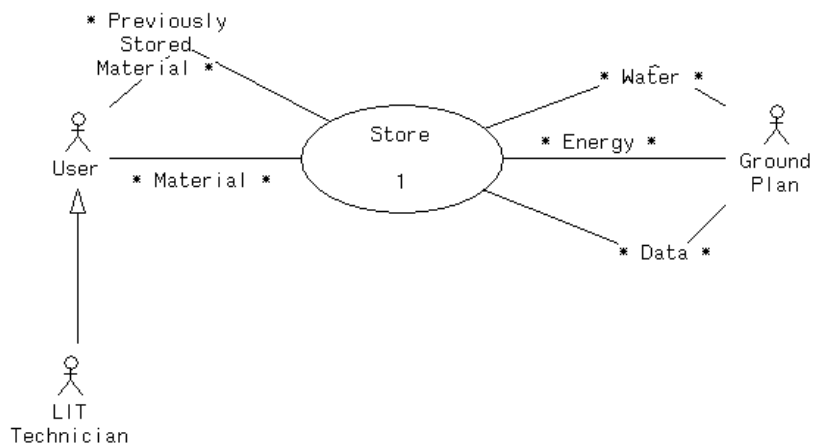


Figure 14. Organization functional context for the 'store' process scenario

Figures 15 and 16 depict the product during 'OBC acceptance testing' and 'EGSE operation' in the central bubble and the elements in the environment during those processes. Links between product and environment are energy, material and information flows.

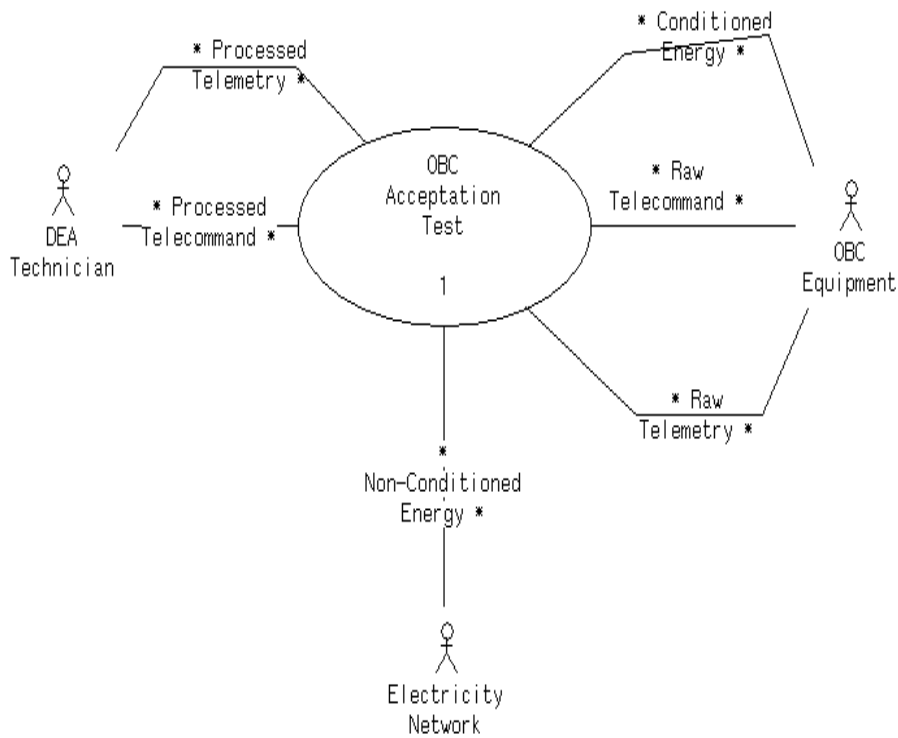


Figure 15. Product functional context for the ‘OBC acceptance test’ process scenario

Figure 16 also presents, besides each element in the environment, some of their potential states. For example, for the ‘electricity network’ potential states are ‘nominal’, ‘peak power’ or ‘lack of energy’. The composition with states of other elements in the environment results in the potential circumstances a system must cope with.

The system must have different modes depending on the circumstances. Behaviour modelling is required to show under which conditions system mode and system state transition occurs. Functions are identified per mode. Functions are identified from outside in by identifying which responses the system is supposed to give to deal with each stimulus provided by the environment elements. For each function, performance requirements are identified. Circumstances, flows between the system and the environment and function failures are sources of hazards. Risk analysis is performed on each identified potential hazard and exception handling functions are also identified at this stage.

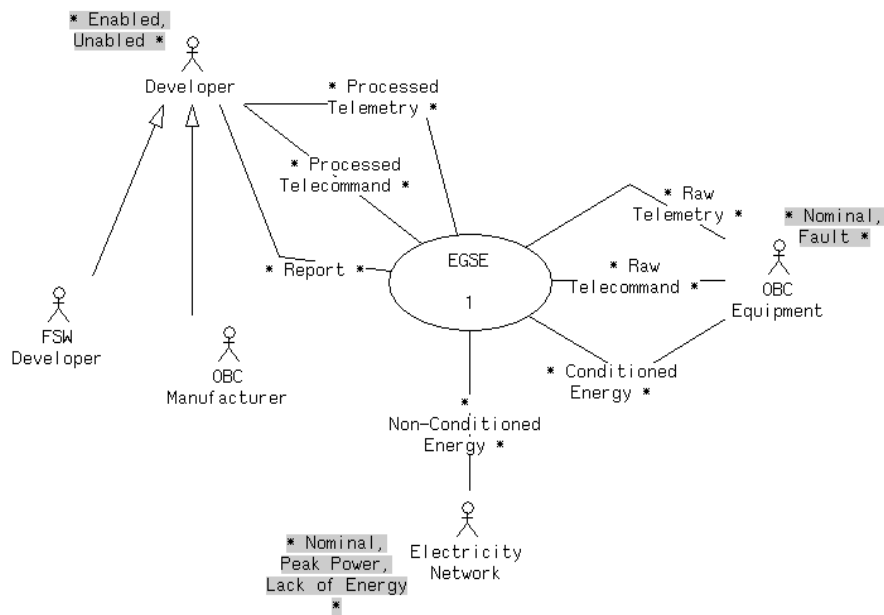


Figure 16. Product functional context for the ‘operations’ process and circumstances identified

The figures below represent the stimulus provided by the environment elements and the responses that the system is supposed to give. Figure 17 shows the stimulus and responses captured from the organization functional context model for the ‘store’ process scenario, described in Figure 14.

Stimulus	Responses
LIT technician delivers EGSE	System receives EGSE
Warehouse responsible requires identification of the EGSE	System generates identification to the EGSE

Figure 17. Stimulus and responses for ‘store’

Figure 18 shows the stimulus and responses captured from the product functional context model for the ‘error analysis’ process scenario during operation of the EGSE, described in Figure 16.

Stimulus	Responses
Developer transmits telecommand	System receives telecommand
Developer receives telemetry	System transmits telemetry
OBC receives telecommand	System transmits telecommand
OBC transmits telemetry	System receives telemetry

Figure 18. Stimulus and responses for ‘error analysis’

Functional structure and functional behaviour of product and organization are depicted in the Figures 19 to 22. It identifies the functions of the system under conditions of the system mode and system state transitions.

Figures 19 to 22 depict the product functional structure and product functional behaviour for ‘error analysis’ and organization functional structure and organization functional behaviour for ‘store’.

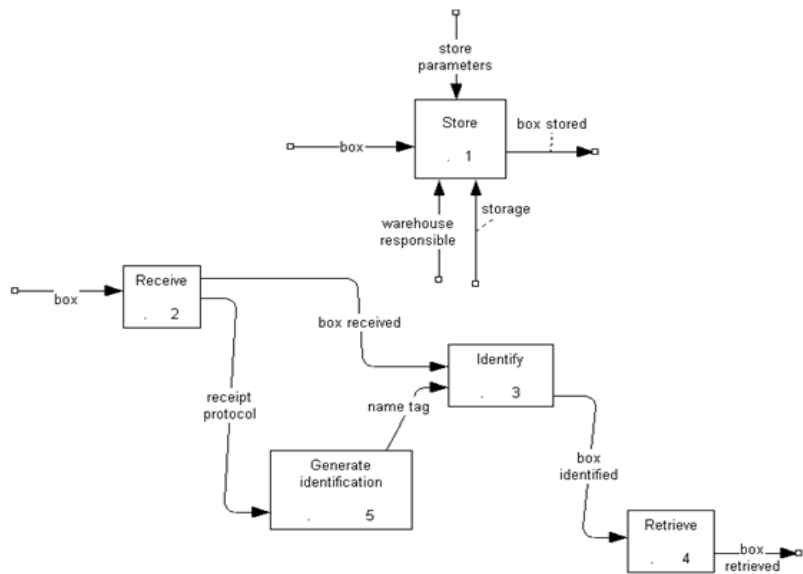


Figure 21. Organization functional structure for 'store'

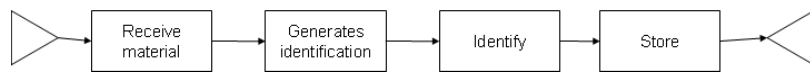


Figure 22. Organization functional behaviour for 'store'

Circumstances, flows between the system and the environment and function failures are sources of hazards. Passport diagram helps to identify function failures in the flows between system and environment. Figures 23 and 24 depict the passport diagram for 'error analysis' and 'store'.

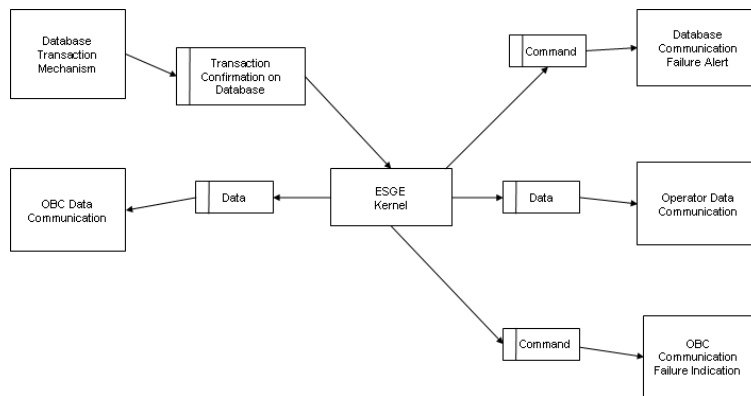


Figure 23. Passport diagram for 'error analysis'

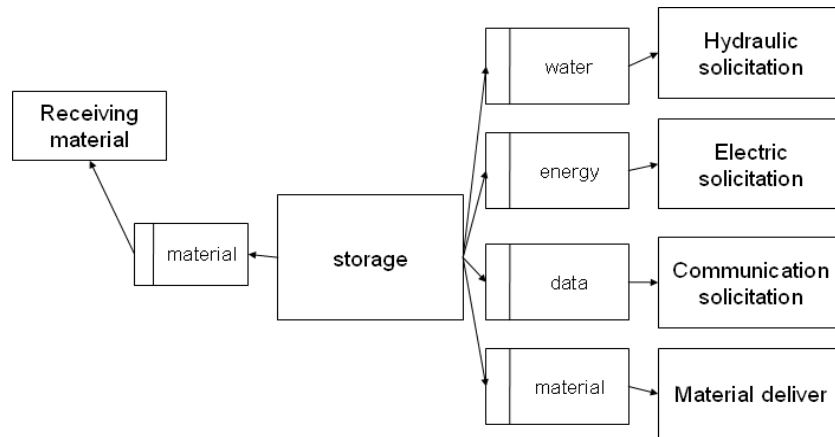


Figure 24. Passport diagram for 'store'

The Figure 25 depicts the risk analysis for potential hazards. The source of potential hazards comes from the circumstances, passport and non-function depicted in Figures 16 and 23.

Circumstances	Type	Failure	Hazard	Consequence
Electrical network peak	Circumstance	Equipment overvoltage	Burning OBC	Loss the OBC
OBC data communication	Passport	Lack of communication between OBC and EGSE	Loss the test	Test inconsistency record
Telecom record	Non-function	No full registration	Lack of information	Test inconsistency record

Severity	Cause	Detection	Probability	Risk	Functions	Verification
5	Peak	4	3	15	Stabilization/ prevention	Inspection
1	Rupture of cables, no connection	5	1	1	Connection of flags (leds)	Demonstration
3	Database unavailability	3	2	6	Transaction mechanisms/ protection	Test

Figure 25. Risk analysis

Figure 26 presents the external physical connections between the 'store' organization and the elements in its environment.

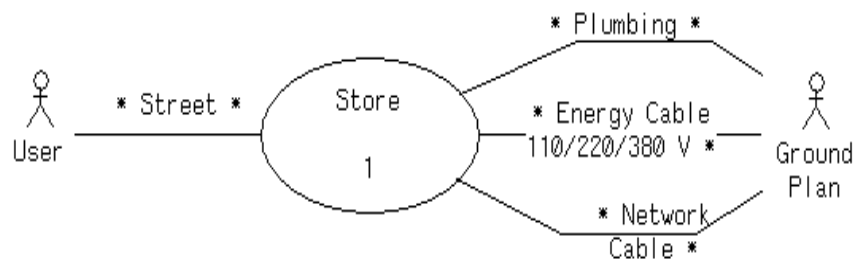


Figure 26. Organization implementation architecture context during the 'store' process and external physical interfaces identified

Figure 27 presents a decomposition of the ‘store’ organization into its internal elements and also shows the internal physical interfaces among internal elements.

Figure 28 shows the product EGSE and its external physical interfaces with the elements in the environment, during EGSE use process. Figure 29 decomposes the product EGSE into its constituent parts. EGSE internal physical interfaces among its parts are also depicted in Figure 29.

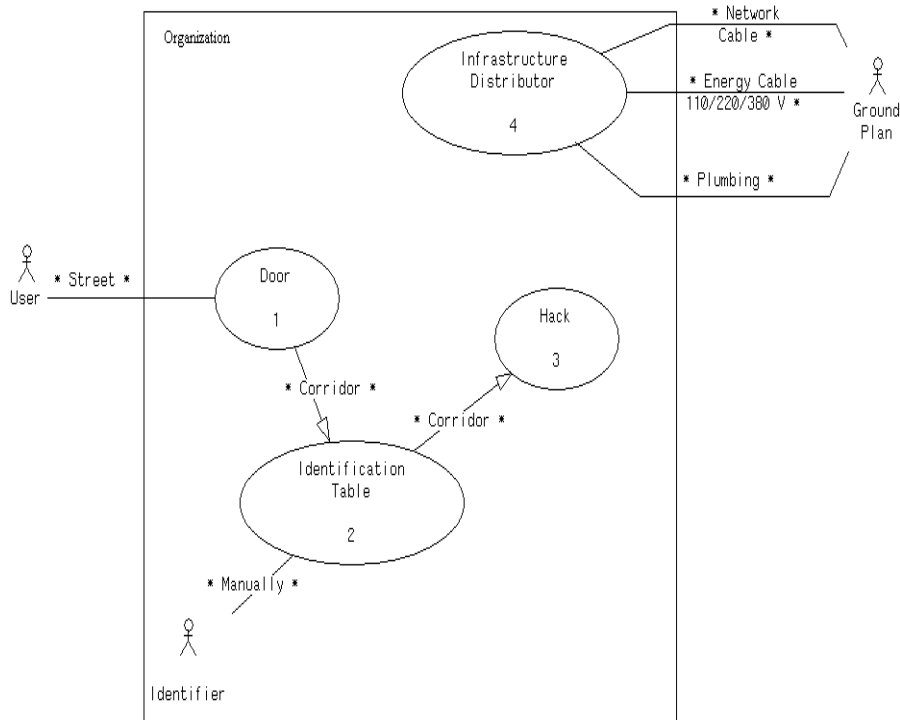


Figure 27. Organization implementation architecture with internal elements and internal physical interfaces for the ‘store’ process

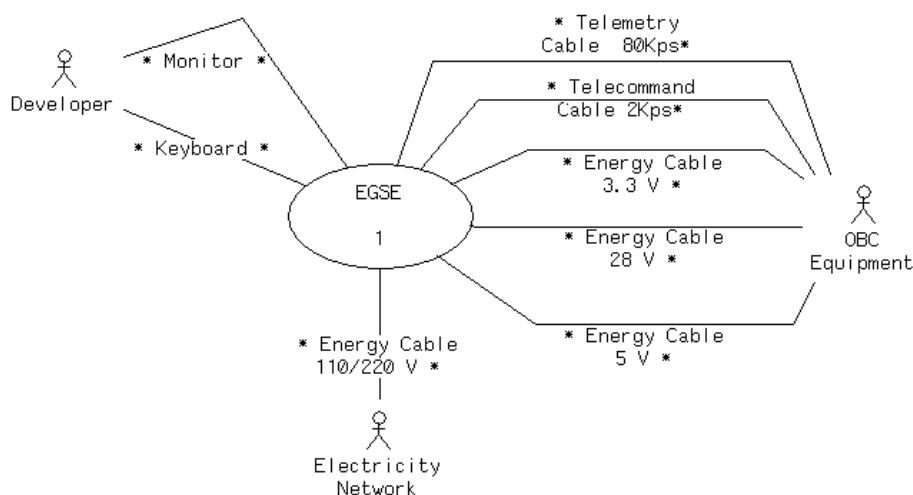


Figure 28. Product implementation architecture context during the ‘operation’ process and external physical interfaces identified

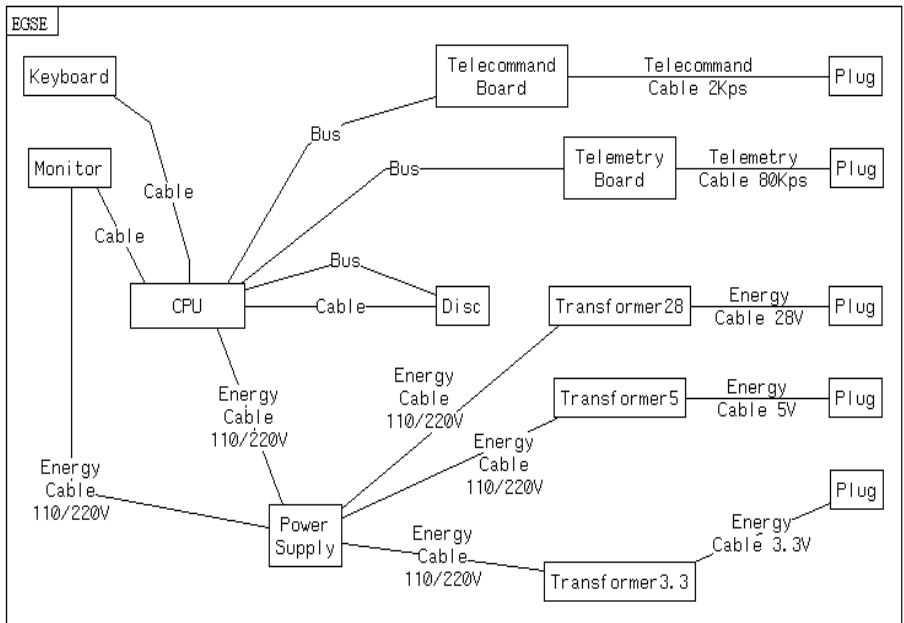


Figure 29. Product implementation architecture decomposition with internal physical interfaces

An allocation matrix relates physical parts and physical interfaces to the functions and functional flows. For example, from the functional decomposition of the context diagram in Figure 16, functions can be allocated to the parts identified in Figure 29. This allocation is depicted on an, so called, allocation matrix, depicted in Figure 30.

		Subsystems										
		monitor	keyboard	disc	CPU	telemetry board	telecommand board	transformer 28 V	transformer 5 V	transformer 3.3 V		
Functions	Provide telecommand receiving											
	Record telecommand											
	Provide telecommand sending											
	Provide telemetry sending											
	Record telemetry											
	Provide telemetry receiving											
	Provide report											
	Receive energy											
	Provide conditioned power											
	Distribute conditioned power											

Figure 30. Allocation matrix

5 Discussion

This sections highlights the differences between traditional and proposed approaches.

Complex products such as space products or space product support equipment such as the OBC EGSE analyzed in this paper have many stakeholders. It is not possible to consider only customer or user as stakeholders of interests, like in the traditional approaches. Stakeholders related to all product life cycle process must be taken into consideration from the outset of the system architecting process. The proposed approach accomplishes it. (see Steps 1 and 2 in Section 3)

Traditional systems engineering approaches perform functional context analysis only during product operations (the so called CONOPS or concept of operations) and for product development organization processes. However, a system solution is comprised of product and organization elements and many enabling elements must be also developed for mission success. These elements are only identified if context for each life cycle process scenario is performed. Therefore, the proposed approach covers the overall product life cycle, not only operations and development. (see Step 3 in Section 3).

By considering product life cycle processes from the beginning of the system architecting process and from the top level context diagrams to be decomposed in lower level functions and lower level physical architectures, the concurrent engineering concept is implemented within the systems engineering process. This fulfills the framework proposed in Figure 1.

The proposed approach allows requirements from the whole product life cycle to be anticipated to the early stages of a system architecting process. Stakeholder requirements are captured for the whole product life cycle process. Functions, performance, conditions, circumstances, modes and exception functions are captured for the whole product life cycle process. External physical and logical interfaces and internal physical and logical interfaces are identified for the whole product life cycle process. The system solution here is composed of product and organization elements. The product interaction with other system elements is identified in the beginning of the system architecting process. This promotes dramatic gains in productivity during product development and during product life cycle. System quality increases. Product changes are avoided. Changes cost and time are eliminated.

6 Conclusion

This paper presented a system concurrent engineering approach for a satellite OBC EGSE. The proposed approach addressed the deficiencies of traditional methods, such as, product focus, operation and development focus, and part focus. The paper described the approach as a way to perform stakeholder analysis, requirements analysis, functional analysis and implementation architecture, simultaneously, for the product and organization elements of a system at every layer of the system breakdown structure. This is necessary to address all complexity factors that are inherent to complex product development. Conclusions are that impact, traceability and hierarchy links promote the anticipation of life cycle process requirements to the early stages of systems architecting. Late changes are avoided, development costs are dramatically reduced while satisfaction of stakeholders over product life cycle is increased.

7 Acknowledgement

The authors would like to thank CAPES (Coordination for Supporting and Development Superior Education Personnel, www.capes.br) for Jonas Bianchini Fulindi scholarship.

The authors would like to thank INPE (the Brazilian Institute for Space Research, www.inpe.br) for the post graduate course and the case study opportunity. The authors would like to thank IAE (Aeronautics and Space Institute, www.iae.cta.br) and 3SL (www.threesl.co.uk) for providing Cradle, the systems engineering environment software used for the development of this work.

8 References

- [1] Alexander C. Notes on the synthesis of the form. Harvard university press. Cambridge, USA, 1964.
- [2] Electronic industries alliance. EIA-632: process for engineering a system. Arlington, 1997.
- [3] European cooperation for space standardization. ECSS-E-ST-10C: space engineering: system engineering general requirements. Noordwijk: ESA, 2009.

- [4] Fulindi, J. B., Loureiro, G., Loures da Costa, L.E.V. Concurrent systems engineering of a microsatellite launch vehicle. In: proceedings of the 61st International Astronautical Congress. IAC-10-D1.6.9. Prague: IAF, 2010.
- [5] Hitchins D. K. Getting to grips with complexity. In: proceedings of the 2nd annual conference of the incose – UK chapter, 1996.
- [6] Huang G. Q. Design for x: concurrent engineering imperatives. London: Chapman Hall, 1996.
- [7] Institute of electrical and electronics engineers. Systems engineering: application and management of the systems engineering process. New York: IEEE, 2005.
- [8] Loureiro G., A systems engineering and concurrent engineering framework for the integrated development of complex products. Ph.D. thesis, Loughborough University, 1999.
- [9] Loureiro, G., Fulindi, J. B., Romero, A. G., Kucinskis, F. N., Lemonge, C. E. A., Vazquez, R. F., Miyashiro, M. A. S. Systems Concurrent Engineering of an Electrical Ground Support Equipment for an On-Board Computer. In: Proceedings of the 17th ISPE International Conference on Concurrent Engineering. Krakow: Springer-Verlag, 2010.
- [10] Loureiro, G., Fulindi, J. B., Arandiga, D., Miura, A. E. M. M, Arandiga, F. Systems Concurrent Engineering of a Turbogenerator. In: Proceedings of the 17th ISPE International Conference on Concurrent Engineering. Krakow: Springer-Verlag, 2010.
- [11] Loureiro, G., Fulindi, J. B., Ferreira, E. Z., Silvério, E., Leão, M. S. Systems concurrent engineering of an electric bike. In: Proceedings of the 17th ISPE International Conference on Concurrent Engineering. Krakow: Springer-Verlag, 2010.
- [12] Loureiro, G., Fulindi, J. B., Fidelis, L. A. O., Fernandes, D., Semabukuro, R., Lino, C. O. Systems Concurrent Engineering for the Conception of a Hybrid Vehicle. In: Proceedings of the 17th ISPE International Conference on Concurrent Engineering. Krakow: Springer-Verlag, 2010.
- [13] Loureiro, G., Fulindi, J. B., Gonzales, J., Trivelato, L., Eller, M., Montandon, V. S. Systems Concurrent Engineering to Develop a Green Car. In: Proceedings of the 17th ISPE International Conference on Concurrent Engineering. Krakow: Springer-Verlag, 2010.
- [14] Loureiro, G. Lessons learned in 12 years of space systems concurrent engineering. In: Proceedings of the 61st International Astronautical Congress. IAC-10-D1.5.2. Prague: IAF, 2010.
- [15] National aeronautics and space administration. Nasa systems engineering handbook. Washington, 2007.
- [16] Stevens R, et al. Systems engineering: coping with complexity. London: Prentice Hall Europe, 1998.

ANEXO B - CONCURRENT SYSTEMS ENGINEERING OF A MICROSATELLITE LAUNCH VEHICLE

Apresentado no International Astronautical Congress - IAC 2010.

IAC-10-D1.6.9

J. B. Fulindi

Brazilian Institute for Space Research, Brazil, jonas.bianchini@yahoo.com.br

G. Loureiro*, L. E. L. da Costa[†]

This paper aims to propose a concurrent systems engineering method demonstrated by the use of a commercial systems engineering environment software for the concurrent engineering of a microsatellite launch vehicle (VLM) system. The product (VLM), its life cycle processes and their performing organizations are concurrently modeled throughout the processes of stakeholder analysis, requirements analysis, functional analysis, implementation analysis at each layer of the system breakdown structure. Requirements and attributes captured from those processes are then related to each other obtaining traceability, impact and hierarchical relationships. Concurrent engineering guidelines are then fed to the multidisciplinary integrated product development teams. The paper presents the method, its demonstration on the VLM example and draws some perspectives on the method potential. Conclusions are that the anticipation of product-processes-organization interactions to the early stages of the aerospace product development produces a four-fold gain in productivity and in development time. This in a time of shortening funding for space development is good news.

I. INTRODUCTION

The VLM is an acronym for Micro-satellite Launch Vehicle, in Portuguese – Veículo Lançador de Microsatélites – it is to carry small payloads. The project is under development at Aeronautics and Space Institute - IAE in Brazil.

The development of the VLM project was stated as a cost-driven process where it involves every aspect of the stakeholders, organization, manufacture, new buildings to store and to handling propellers, elements to support the operation of the product and new ground support equipments with traceability during all life cycle processes.

This process should be traceable throughout of the life cycle processes in order to keep the integrity and to gain with the analysis from the outset of the development, saving money and time. Traceability enables the communication between engineers, developers, suppliers and stakeholders.

To accomplish this, an integrated development approach that addresses properly each element should be applied accurately. The method that adopts this approach is an evolution of the Loureiro [1, 2], the Total View Framework and the Concurrent Structured Analysis Method. The method is described in section III. This method was applied in many study cases [3-7].

To address the need for an integrated and traceable development process in the VLM project, it was decided to use a commercial systems engineering software.

This software called Cradle [14] also assists in communication and in clarification of the method applied to VLM development phases. Section IV explains about this software.

This paper aims to:

- 1) Present the concurrent systems engineering approach;
- 2) Demonstrate the use of the approach using Cradle on the VLM example;
- 3) Discuss the potential of the application of the approach and tool.

This paper is organized as following: Section II presents the traditional systems engineering and concurrent engineering approaches. Section III presents the concurrent systems engineering approach used in the VLM project. Section IV presents the systems engineering environment software. Section V presents the concurrent systems engineering approach applied to the VLM case. Section VI discusses the advantages and opportunities for improvement. Section VII concludes this work. Section VIII acknowledges the support received from other parties. Section IX presents the list of references.

II. TRADITIONAL SYSTEMS ENGINEERING AND CONCURRENT ENGINEERING

The systems engineering traditional approaches do not provide an overall view of the system during its various life cycle processes. They focus on an operational product development starting from product concept of operations. They also focus on the development organization that must be put in place in order to assure that the product meets its operational requirements [8-11]. A product has life cycle processes other than operations and it must be acknowledged from

*Brazilian Institute for Space Research, Brazil, geilson@lit.inpe.br

[†]Aeronautics and Space Institute, Brazil, loures@iae.cta.br

the outset in order to promote gains in productivity in the product development organization, by the avoidance of late changes, and in other product life cycle process organizations, as the product will be developed taking into consideration their requirements. Life cycle process organizations themselves can be developed simultaneously to product development, when they are part of the scope of the whole product development effort.

For example, the NASA systems engineering handbook [11] states that systems engineering focuses in the development and the realization of a final product. Modern commercial standards, such as EIA 632 [8], state that systems engineering focuses on the operations product and on capturing requirements for the other product life cycle processes. In other words, these requirements are captured not to impact product development. The product will be systems engineered with operations in mind. When its architecture (and maybe detailed design) is defined, then life cycle processes requirements are captured to be implemented in life cycle process performing organizations. The method proposes to take into consideration the impact of these organizations on the product from the outset of the product architecting process.

Conceptually, concurrent engineering acknowledges benefits of anticipating life cycle process requirements to the early stages of product development. For space products, these early stages are the system architecting phases. A systems approach requires life cycle process requirements to be balanced in the beginning of the product development process. Concurrent engineering, however, in practice, treats life cycle processes separately and optimizes product design seeking each life cycle process productivity increase. For example, DFA (design for assemble) optimizes for assemblability, QFD (quality function deployment), for customer satisfaction, DFI (design for inspectability), for inspectability, and so on. Also, concurrent engineering is, in practice, applied to parts design and not to systems composed of many integrated parts [12]. The concurrent systems engineering proposes how the concurrent engineering concept can be used for systems engineering.

III. THE CONCURRENT SYSTEMS ENGINEERING APPROACH

Fig. 1 presents a framework to address complexity in product development – the total view framework evolved from Loureiro [1]. It has three dimensions. Each dimension addresses one of the complexity factors mentioned above. The analysis dimension addresses the variety factor. Along the analysis dimension, it is deployed what must be analysed in order to develop a complex product. A systems engineering process consists of stakeholder analysis, requirements analysis,

functional analysis and implementation or physical analysis. The integration dimension addresses the connectedness factor. It defines what must be integrated along an integrated product development process: product elements and organization elements. Organization here refers to the organizations that perform product life cycle processes. Product elements and organization elements are the system elements. The structure dimension addresses the disorder factor. According to Alexander [13] all structures evolve into a hierarchy. System breakdown structures are also represented in hierarchies.

Fig. 2 provides an overview of a method within the total view framework. The method is called concurrent structured analysis method evolved from Loureiro [1]. Stakeholder analysis, requirements analysis, functional analysis and implementation (or physical) analysis is performed, simultaneously, for the product under development and its life cycle process performing organizations. The analysis processes are performed at each layer of the system breakdown structure.

Fig. 3 details the concurrent structured analysis method showing how to incorporate the concurrent engineering concept in the systems engineering process:

Step 1: Identify the product mission, the product life cycle processes and their scenarios and, the scope of the development effort. Product mission refers to the product purpose or reason of being. Life cycle process scenarios are the alternatives in each process (for example, preventive or corrective maintenance) or the decomposition of a process (for example, advanced technology development, process engineering as components of the development process). The scope of the development effort consists of the life cycle processes or their scenarios that the development organization is also responsible for accomplishing.

Step 2: Identify product stakeholders and their concerns for each product life cycle process scenario. Product stakeholders are the people who affect or are affected by the product during its life cycle. Product stakeholders are identified per life cycle process scenario. Identify organization stakeholders and their concerns for each process within the scope of the development effort. Organization stakeholders are the people who affect or are affected by the business of the organization in question. Organization stakeholders are identified per life cycle process scenario within the scope of the development effort. From stakeholder concerns, stakeholder requirements are identified and measures of effectiveness (MoEs) are derived. MoEs must measure how the system meets the stakeholder requirements. From stakeholder requirements, functions, performance and conditions are identified. The definition of what functions the system will perform, how well the system is going to perform such functions and under which conditions comprise the requirements analysis process

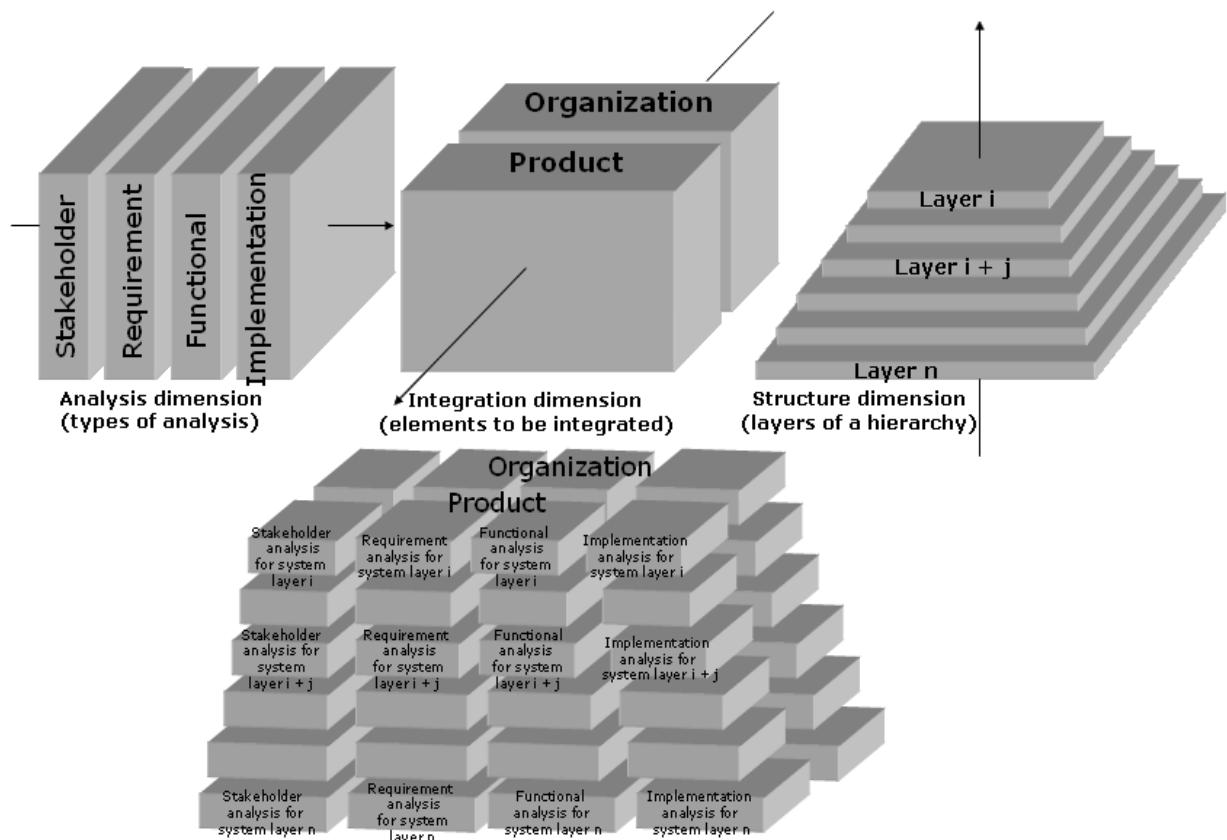


Fig. 1: A framework to address complexity in complex product development – the total view framework.

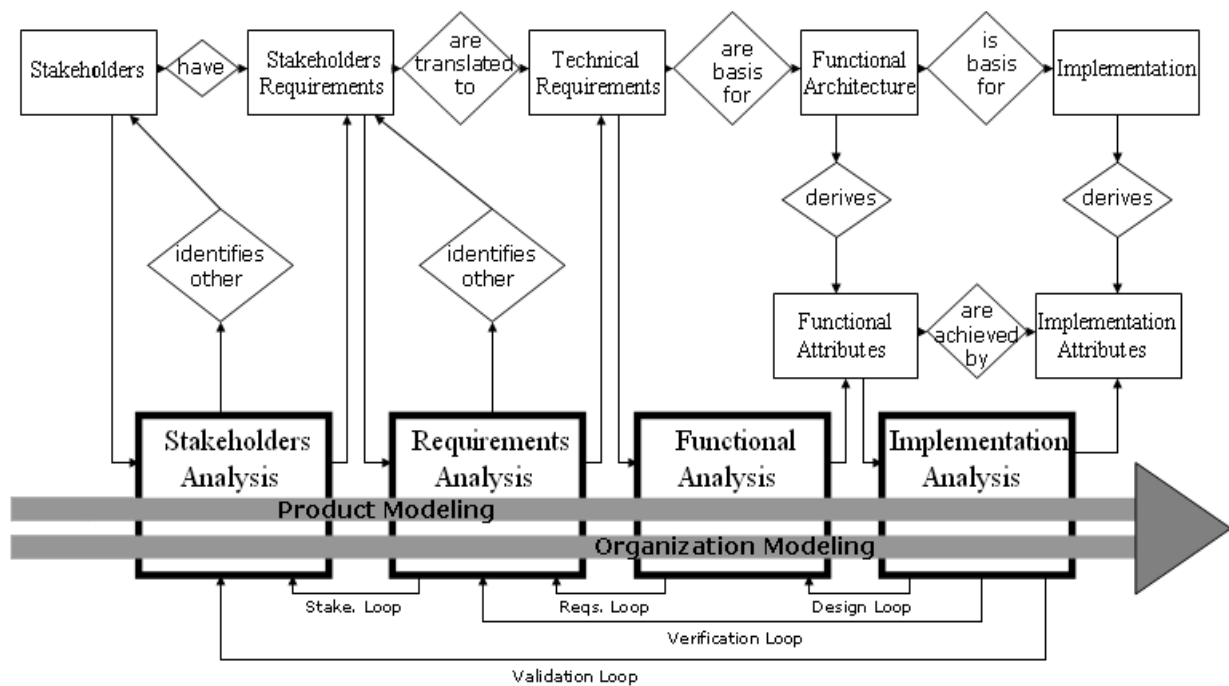


Fig. 2: A method within the total view framework – the concurrent structured analysis method.

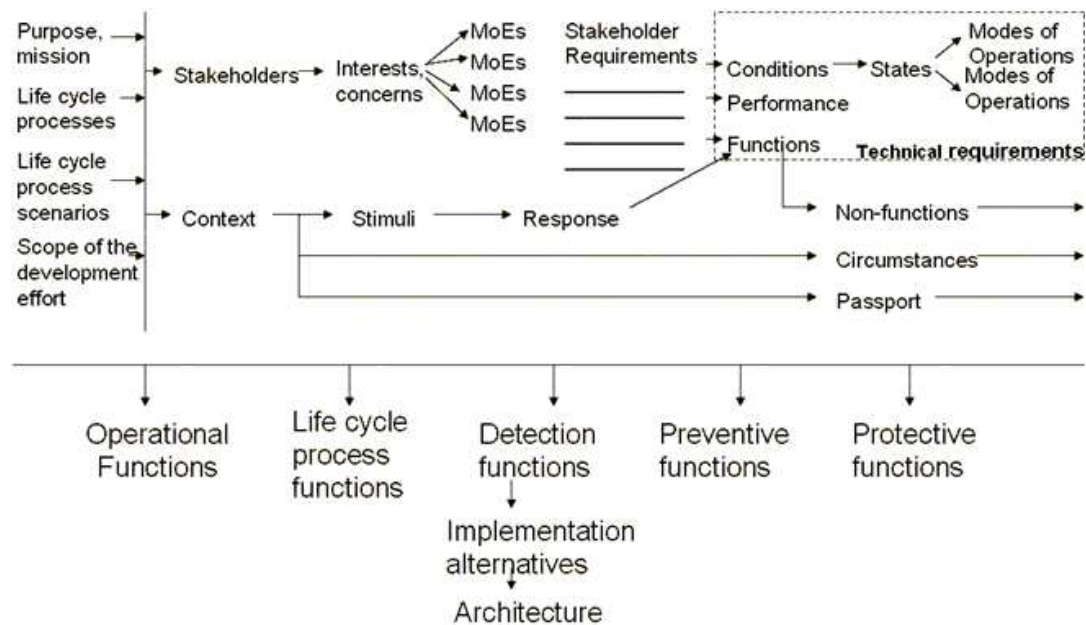


Fig. 3: The system concurrent engineering method in detail. Source: [16].

Requirement analysis transforms stakeholder requirements into system requirements. System requirements will be met not only by product elements but also by organization elements.

Step 3: Identify functional context for product at each life cycle process scenario and for organization at each life cycle process scenario within the scope of the development effort. Functional context defines the function performed by the system element and identifies the elements in the environment of the system. The environment of the system contains the elements outside the system function scope and that exchanges material, information and energy flows with the system. Those flows define logical interface requirements. Environment elements may have different relevant states. Sets of environment element states are called circumstances. The system must have different modes depending on the circumstances. Behaviour modelling is required to show under which conditions system mode and system state transition occurs. Functions are identified per mode. Functions are identified from outside in by identifying which responses the system is supposed to give to deal with each stimulus provided by the environment elements. For each function, performance requirements are identified. Circumstances, flows between the system and the environment and function failures are sources of hazards. Risk analysis is performed on each identified potential hazard and exception handling functions are also identified at this stage.

Step 4: Identify implementation architecture context for product at each life cycle process scenario and for

organization at each life cycle process scenario within the scope of the development effort. Physical connections between the system and the environment elements define the physical external interface requirements. Physical parts are identified. Physical internal interfaces are defined by architecture connections and architecture flows among those parts. Allocation matrix relates physical parts and physical interfaces to the functions and functional flows.

IV. SYSTEMS ENGINEERING ENVIRONMENT SOFTWARE

Cradle is a requirements management and systems engineering tool developed by 3SL. Cradle is available in 6.3 version [14]. A systems engineering software needs to provide basic capabilities to perform requirements analysis and management, functional analysis, implementation analysis and configuration management. Cradle is composed of the following modules to support these capabilities: Cradle Project Data Module, Requirements Management, Metrics, Document Generator, System Modelling, Software Engineering, Web Access, Web Publisher, Performance Modelling, and Requirements Traceability. All of these functionalities are integrated to cover the whole life cycle development process.

Cradle has a suite of graphical modelling tools to build business process, analysis, architecture and design models. One modelling notation by itself cannot express what is necessary along the systems engineering process, at all levels of system abstraction. Models with one or a combination of modelling notations for

functional or implementation analysis, structural and behavioural models are necessary. Structural models are provided by notations such as IDEF0 – integrated definition for functional modelling and DFD – data flow diagram, AID - architecture interconnect diagram, PAD – physical architecture diagram. Behavioural models are provided by notations such as STD – state transition diagram, behaviour diagram, eFFBD – enhanced functional flow block diagram. Cradle provides also all UML 2.0 notations such as UCD – use case diagram, SQD – sequence diagram, COD – collaboration diagram, CD – class diagram, SCD – statecharts, ACD – activity diagram, CPD – component diagram, DPD – deployment diagram. UML notations also provide for structural and behavioural modelling of functional and implementation models of product and of organizations [15].

Cradle can manage design work and engineering information flow. It plays a major role in communication and to solve technical problems. The engineer can visualize what must be covered by the analysis throughout different scenarios. For example, a cross-reference links the stakeholder statement to the requirement. The requirement is linked to its system requirement. The system requirement will perform a function and this function will be implemented by a physical element. All cross-references are manageable in Cradle. It permits a concise, traceable and integrated design with the relationship of the elements of the information.

The application of the method utilising Cradle is presented in Section V. The analysis is illustrated by the diagrams and their application, simultaneously, to product and organization, in the various scenarios of the product life cycle processes.

V. THE VLM CONCURRENT SYSTEMS ENGINEERING

This section illustrates an approach for the automation of the concurrent systems engineering method described in Section III using Cradle. The analysis is performed for product and organization simultaneously for each life cycle process scenario. The attributes are derived from the functional and physical, structural and behavioural models. The models aid to derive attributes depending on the level of abstraction and, as it is systematically performed, reduces the risk of forgetting relevant information for product and organization integrated development.

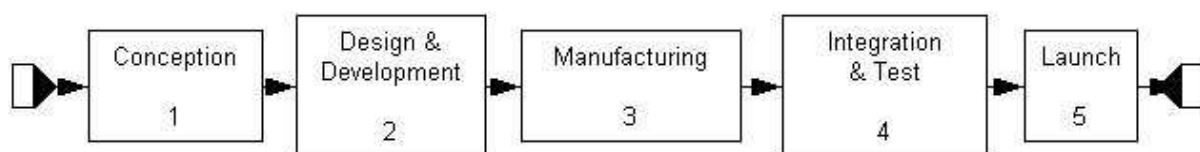


Fig. 5: Product life cycle processes.

All of these elements are captured using Cradle. Having the attributes, the identification of the interactions among product and organization elements attributes will show restrictions imposed by product on organization and vice-versa. Fig. 4 illustrates the scope of the development effort.

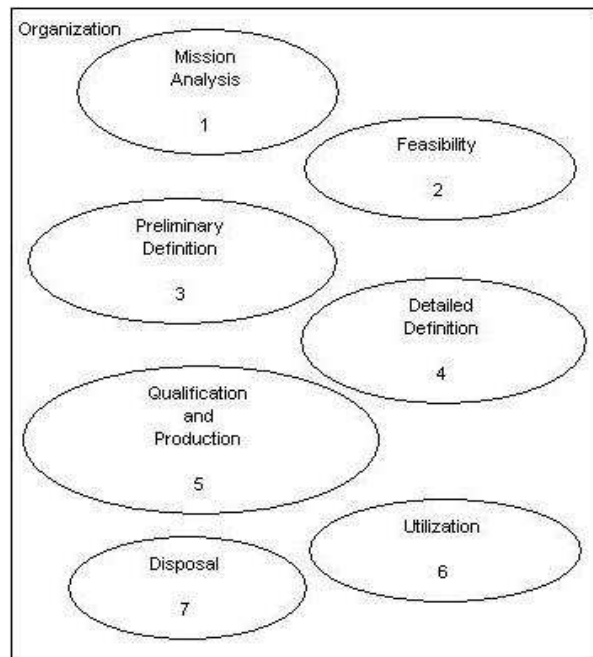


Fig. 4: Scope of the development effort.

Fig. 5 illustrates the product life cycle processes. It depicts what the product must comply from the conception to the mission realization.

After identifying the scope of the development effort and the product life cycle processes the stakeholder analysis for the product and the organization can be developed.

The concerns of the stakeholders are captured from the models. It will be translated later in stakeholder requirements.

Fig. 6 illustrates the organization stakeholders. The stakeholders express their concerns. In practice, Step 2 in Section III the organization stakeholder analysis must be performed for all life cycle processes within the scope of the development effort.

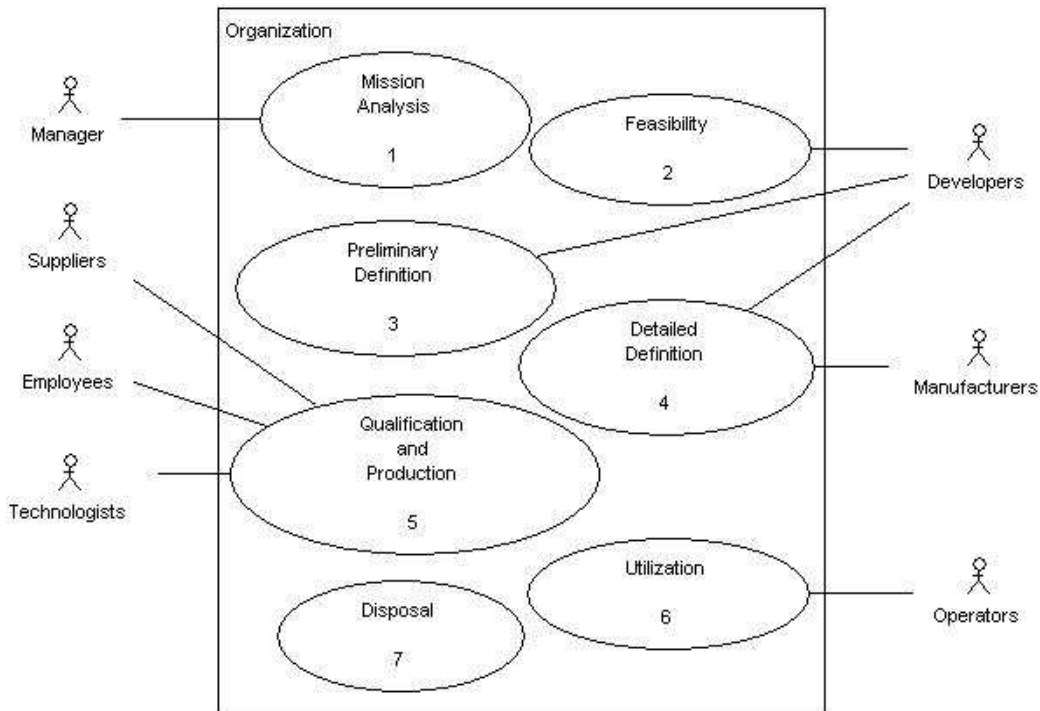


Fig. 6: Organization stakeholders.

Information items are created in Cradle to represent any element in the models. These information items are cross-referenced and the traceability can be viewed by an hierarchy diagram. Hierarchy diagram is a function in Cradle to list the cross-referenced elements and to check, follow and manage the information in the links.

Fig. 7 illustrates the organization stakeholders and their concerns. Stakeholder is an information item in Cradle and can be linked with other information items. An item associated with stakeholder is the item 'concern'. The cross-reference created to link them can receive a name. For example, the link name between stakeholder and concern may be 'declared by'. It means that a stakeholder is linked by a declaration of its concern. Or yet the concern is declared by a stakeholder. The system is validated by the stakeholder

satisfaction through the traceability from their concerns to the implementation. An example is showed in Fig. 21.

Fig. 8 illustrates the stakeholders and their concerns for the VLM product. The stakeholders are identified for each product life cycle process scenario. For each scenario, the stakeholders and their concerns are identified for the VLM. In this example the concerns are listed in the arrows. These elements listed in the arrows can be created as an information item to be cross-referenced with other information items to represent concise linked elements to describe the product.

Fig. 9 illustrates the items of information related to stakeholders and their concerns in the hierarchy diagram, based on Fig. 8.

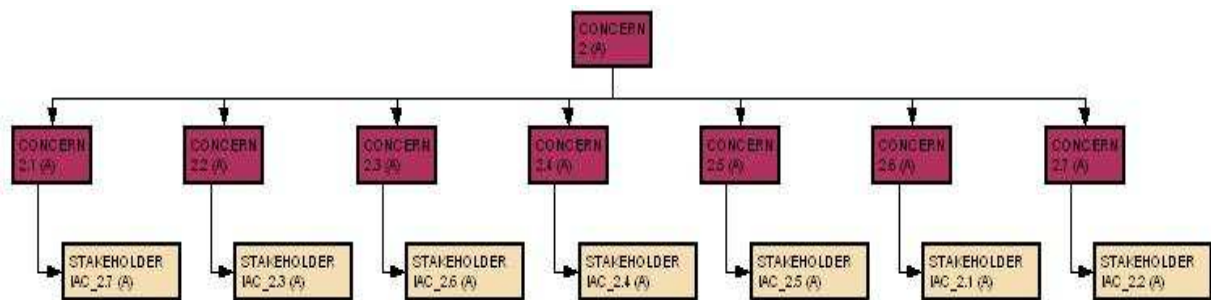


Fig. 7: Organization stakeholders and their concerns viewed by the hierarchy diagram in Cradle.

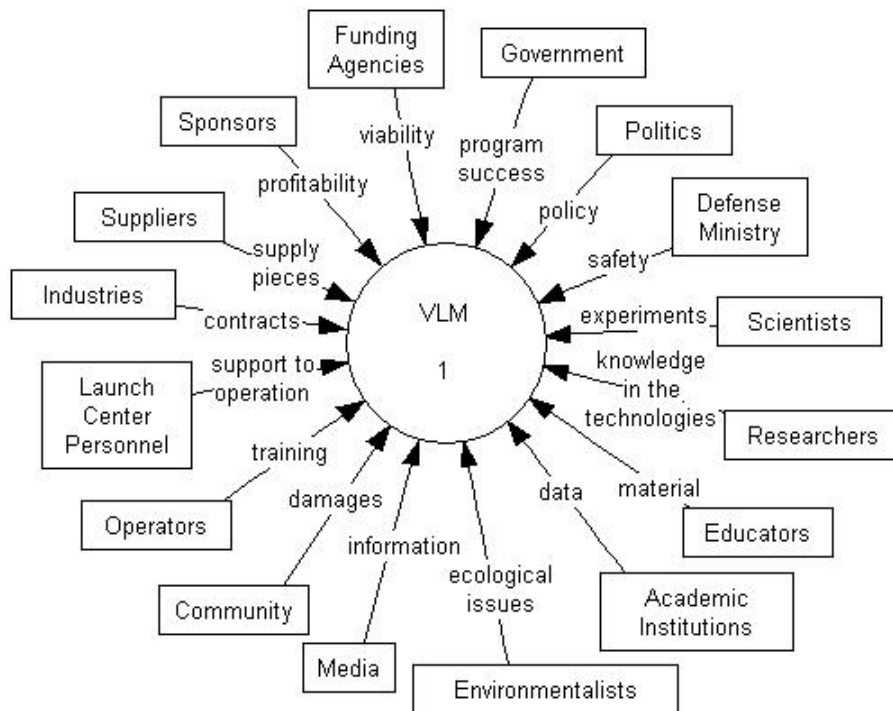


Fig. 8: VLM Product stakeholders and their concerns.

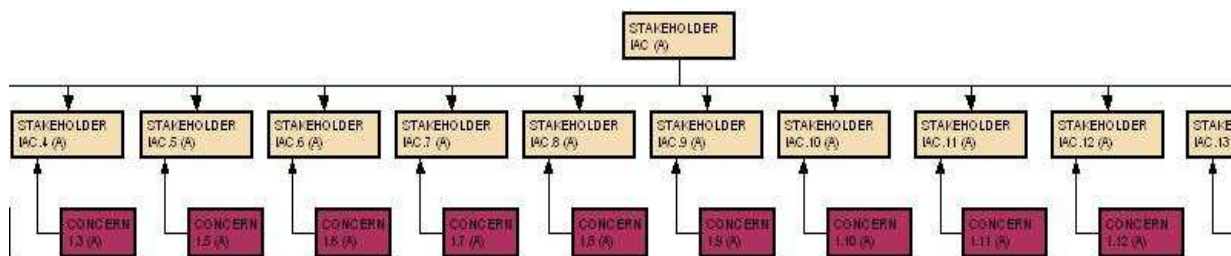


Fig. 9: Product stakeholders and their concerns viewed by the hierarchy diagram.

From the models above the stakeholder requirements can be translated. The stakeholder requirements should be linked with the stakeholder and their concerns to keep the traceability and to cover their concerns. The covering of the concerns means that requirements translate the concerns of the stakeholders.

Fig. 10 illustrates in Cradle how the stakeholder, concerns and requirements are related. This relationship is created cross-referencing the information items based on the models as depicted in Fig 7 and Fig.9. Fig. 10 illustrates a simple relationship of the items of the information representing stakeholders, concern and requirement. Giving a double click in the element, text can be added (See Fig. 11).

Fig. 11 illustrates the windows captured from Cradle. Each element can receive texts in the form in order to provide the description, the details, acceptance criteria and other information.

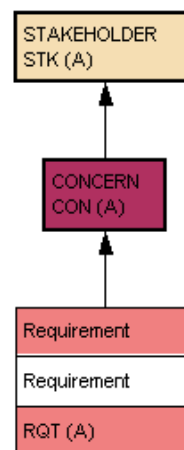


Fig. 10: The relationship of stakeholder to concern item to the requirement.

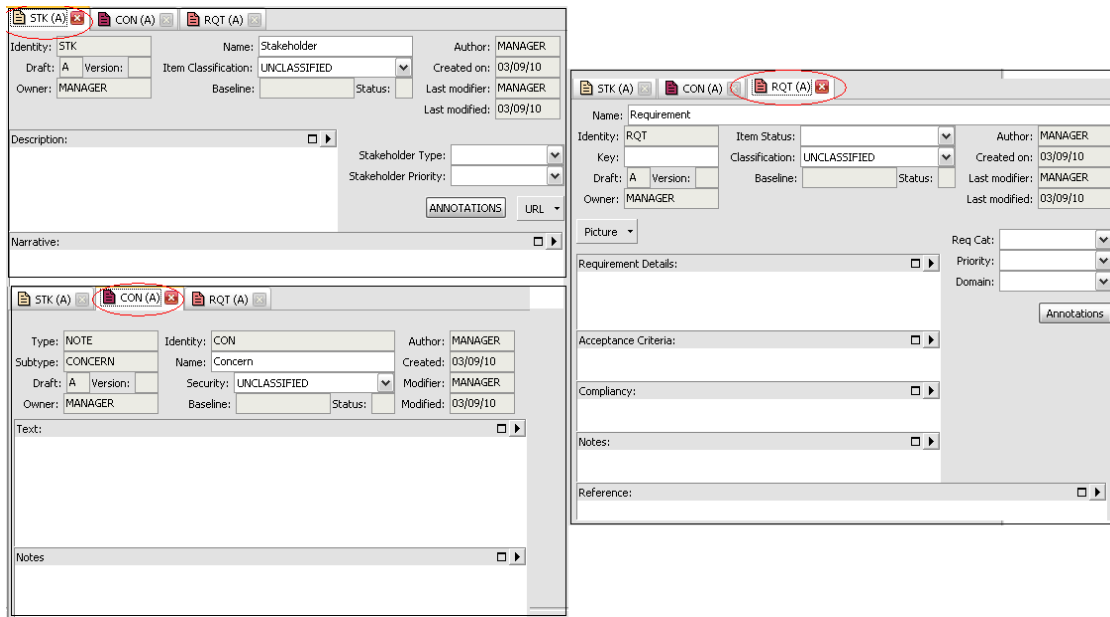


Fig. 11: The expanded elements to be written texts.

Cradle allows to describe all elements and to control access to this information. Access control impacts the project. For example the elements defined by the stakeholder can not be altered by other people in the Cradle project if they do not have a given privilege.

From the stakeholder analysis and the requirements analysis, the functional analysis is developed. Functional context modelling seeks to identify what exchanges material, information and energy with the system.

Fig. 12 illustrates the flow as material (funding) and information (requirements) from the requirements analysis.

Fig. 13 illustrates the functional context of the VLM product with the payload. There is all kind of interactions, for example, electric, electromagnetic, mechanical, environmental interactions as atmosphere, rain and wind. Based on this a contingency plan can be prepared preventing if something going wrong.

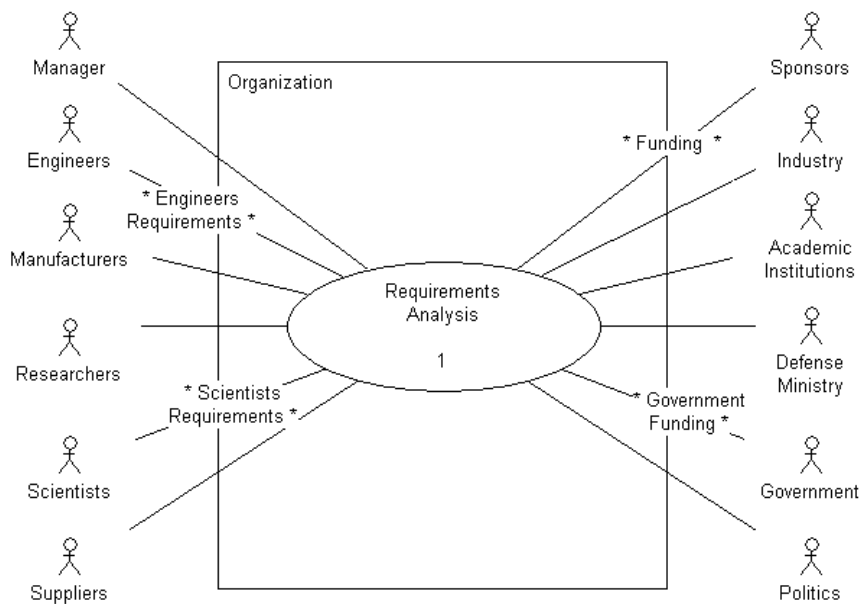


Fig. 12: Requirements analysis organization – functional context.

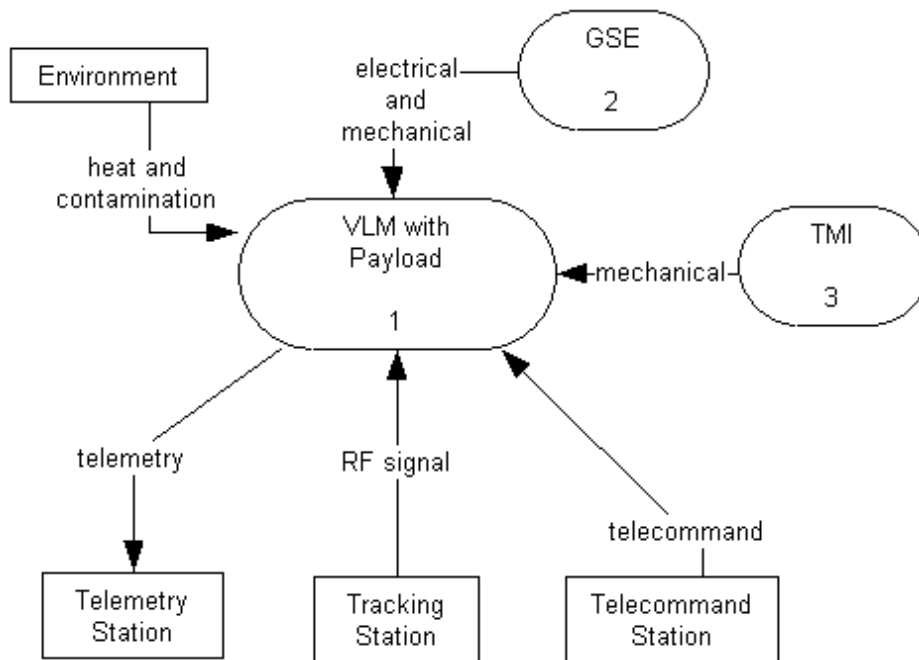


Fig. 13: Product functional description. Interaction with many elements during integration of the payload.

Figures 14 and 15 refer to organization functional decomposition. Functional decomposition, using even the same notation, is also performed for the product but is not presented in this paper. The diagrams describe the sequence of activities from the material transportation from the laboratories up to the launch pad and the realization of the mission. Each block illustrated in the diagram is a function to be performed considering many elements and features of the product and the organization.

This example was chosen because, in order to perform those functions, product and organization attributes must be considered in conjunction, from the outset. The strong relationships among these attributes show that for those functions, product and organization elements are highly coupled. Usually these activities are critical in terms of cost, rework, and delay in the schedule. The models serve as a means to communicate with the various product and organization experts involved, so that everyone share the same information from the model. The diagrams are set based on information captured from the experts. Once it is captured it can be stored in Cradle. For a new future development project it can be reused. The models and information items created in Cradle can be re-visited for improvement.

Fig. 14 illustrates the main functions to be performed. Fig. 15 expands the 'transport' function in more details. The product is affected in each function in the 'transport' process functional decomposition. For each function, it is necessary to take into consideration

the interactions between product and organization elements performing the 'transport' process. In this level of abstraction, it is already possible to identify the need to improve the transportation facilities and the product to comply with it. Thinking in the 'transport' process or function, one key element identified was a new MGSE - mechanical ground support equipment - to carry tanks filled with fuel from the airport to the integration building.

For example, the experts identified, after a brainstorming, how the tanks could be transported. Anticipation of MGSE attributes in order to support transportation takes place in a moment when the physical product, itself, does not exist yet. For example, there are differences between carrying tanks made by iron and tanks made by composites. In this case there are impacts that affect the product and the organization architectural elements.

The concurrent systems engineering allows to identify, from the outset of the development, the requirements and deriving attributes that describe the product and the organization.

Fig.14, describes major functions to be implemented. It illustrates the behaviour of the systems in the "prepare to launch" scenario. It starts when the VLM subsystems to be integrated in the launch centre arrives in the launching campus. In this campus there are many facilities. Each facility performs launching preparation activities, functional tests and integration for equipments, subsystems and system.

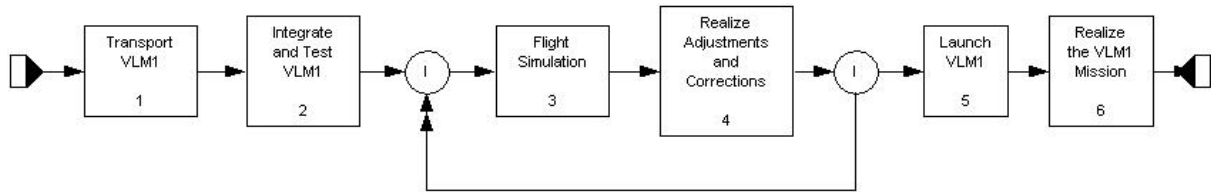


Fig. 14: Functional flow describing the mission.

Fig. 16 illustrates the physical organization elements where these activities are performed. For the ‘transport’ function the experts identified that it could not be possible to transport the filled tanks with the available aircraft, because the MGSE to accommodate the tanks in the aircraft have different attributes from the previous MGSE’s.

Transportation is a stakeholder concern. Using Cradle allows to link such concern to the function and the function to the actual organization elements that will

implement it. The links between stakeholder concerns and functions and between functions and organization elements are called satisfaction links. They mean: a function ‘satisfies’ a concern or an organization element ‘satisfies’ a function.

As already mentioned, Fig. 15 details the ‘transport VLM’ into lower level functions. For each lower level function, functional attributes, such as ‘time’, and physical attributes, such as ‘dimension’, are identified as relevant characteristics that impact the activities.

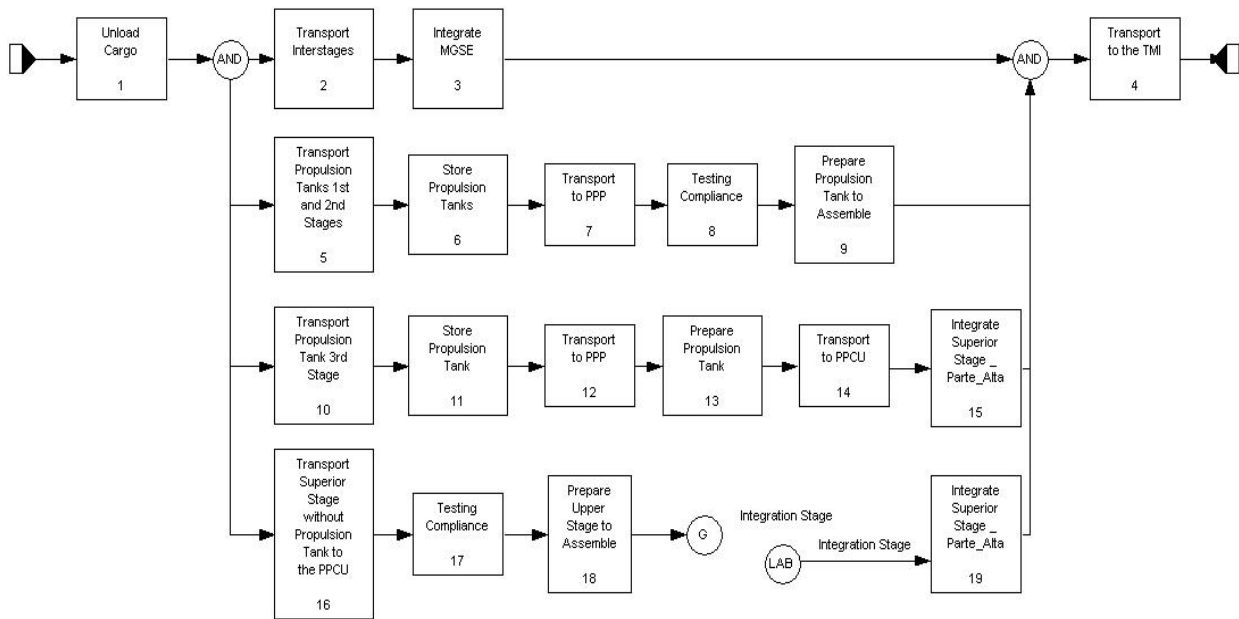


Fig. 15: Higher level of detail (lower level of abstraction) of the ‘transport VLM’.

Fig. 16 illustrates how the ‘prepare to launch’ organization physical elements connect to one another. These elements and the interfaces interact with the product during launch preparation. For example, communication between hardware elements, functional test of the telemetry. From Fig. 16, it can be identified more relevant elements to be implemented in the organization. Attributes derived from this model describes the resources of the physical organization. Having product and organization attributes allows to identify the interaction among them. For example, the

physical organization has facilities where functional tests in the electrical subsystem are performed. The organization performing the test must comply with electrical requirements considering the subsystem under test. Then the interaction between attributes of the test resources in the testing organization and the attributes of the subsystem under test can be identified from the outset of the VLM system development process, anticipating solutions to the early stages of development, in order to integrate product and organization elements.

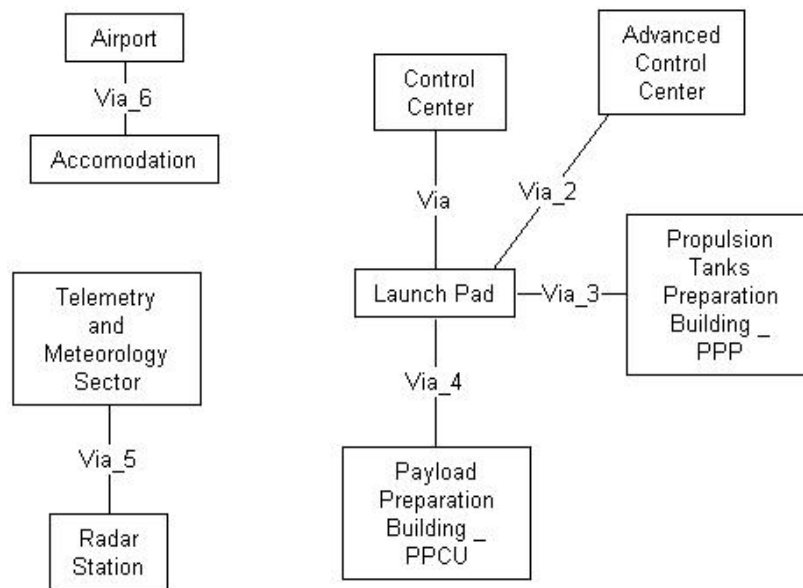


Fig. 16: Physical analysis of 'preparation for launch' organization.

Fig. 17 illustrates the physical elements of the product in the assembly process. The product attributes such as the geometry of the elements to be welded, the stiffness and the length of the tank, at this level of abstraction, will impact the assembly process and

organization. Knowing the value those attributes may assume provides inputs for a better planning of the assembly process and the anticipation of the assembly organization resources needed.

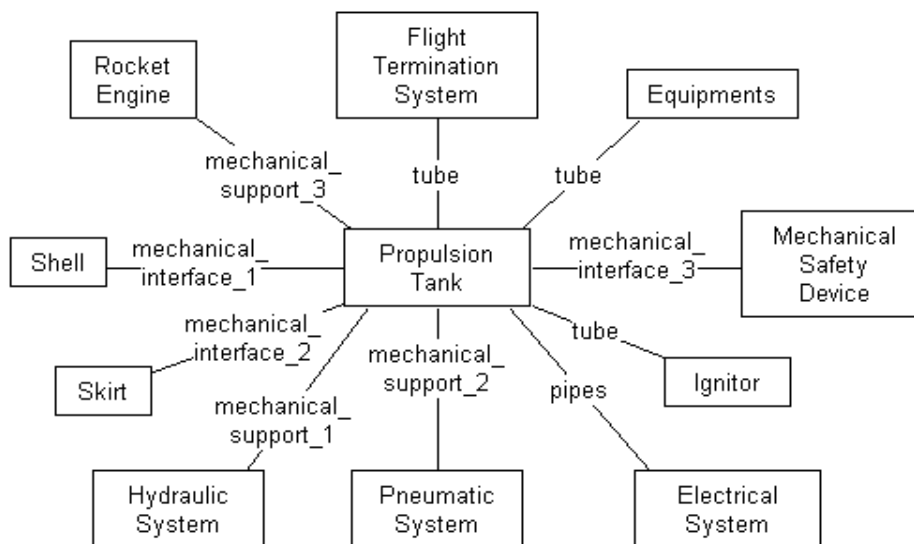


Fig. 17: Propulsion system architecture interconnect diagram – part of implementation analysis.

Fig. 18 illustrates the architecture flow diagram as part of the implementation analysis.. The architecture diagrams show the actual implementation elements of the system and the logical (flows, as in Fig. 18) and physical (interconnections, as in Fig. 17) interfaces among those implementation elements. The

implementation elements in the architecture diagrams implement the functions resulting from the functional analysis. It is commonly said then, that the functions are allocated to these implementation elements. Cradle also allows allocation links to be set up.

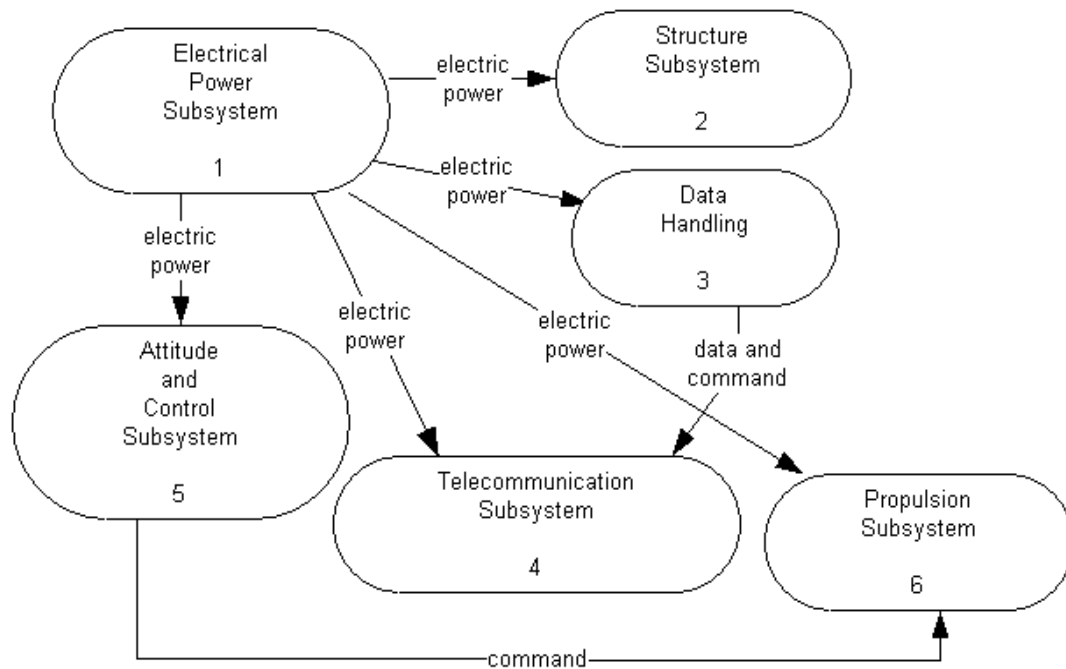


Fig. 18: VLM product implementation analysis – architecture flow diagram.

An allocation hierarchy automatically provided by Cradle allows to identify, for example, whether there is a function not met by any implementation element or whether there is an implementation element for which no function was anticipated during the functional analysis. Traceability links from stakeholders, to concerns, to measures of effectiveness, to requirements, to functions, to functional attributes, to implementation elements and, then to implementation attributes, can all be set up by Cradle.

Tab. 1 exemplify the derivation of functional and implementation attributes for product and organization, respectively. Those attributes provide additional description of the elements in the models depicted in Figures 12, 13, 16, 17 and 18. The relationships between product attributes and organization attributes are identified. The knowledge about those relationships contributes for avoiding late changes in the product due to resources limitations. It also contributes for a better planning of the necessary organization resources for performing a given life cycle process. For example, the attribute ‘amplitude’ from the product functional analysis informs that the organization must have the appropriate resources to meet the requirements of measuring product signals with that ‘amplitude’.

The models allow for functional and implementation elements identification, for product and for organization. From the models, requirements and attributes are captured. Once they are identified, the relationship among them can also be identified.

Integration among product and organization elements is achieved by understanding how these elements impact each other.

	Product	Organization
Functional	-amplitude -frequency -operating bandwidth	-simplicity -clarity -reliability -integrity -characteristics of receiving amplitude signals
Implementation	-geometry -stiffness -length -size -shape	-propulsion storage building size -airport runway capacity to landing and takeoff

Table 1: Attributes from the product and organization models.

Fig. 19 illustrates the relationship of the attributes of product and organization in a matrix created in Cradle. It is a view of the linked attributes. This view can be sorted by type of link. An ‘impact’ link shows which product attributes impact in the organization attributes, and vice-versa.

	ATT::A	ATT.1::	ATT.2::	ATT.3::	ATT.4::	ATT.5::	ATT.6::
Amplitude::A:							
Contaminat::A:				↔			
EMI/EMC::A:					↔		
Geometry::A:		↔					↔
Size::A:		↔	↔				↔
Weight::A:		↔	↔				

Fig. 19: Relationship of product and organization attributes.

Fig. 20: illustrates a form to describe the relationship of attributes. Selecting the link between the information items a description of the relationship can be created in Cradle.

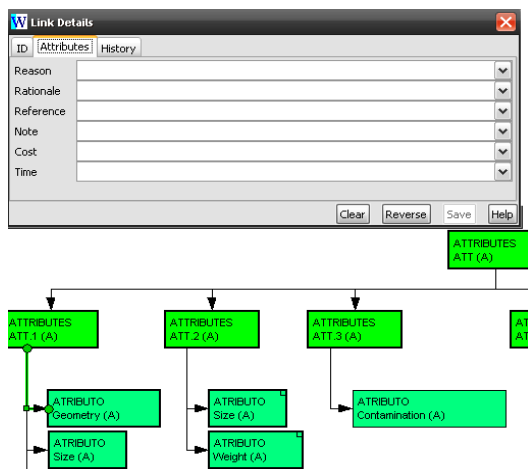


Fig. 20: Form to describe attribute relationship.

Attributes are captured in Cradle as information items. Information items, in Cradle, can be cross-referenced to other information items. This is the way by which product and organization modelling integration is implemented.

Fig. 21 illustrates cross-referenced items captured in the analysis, from stakeholder to implementation

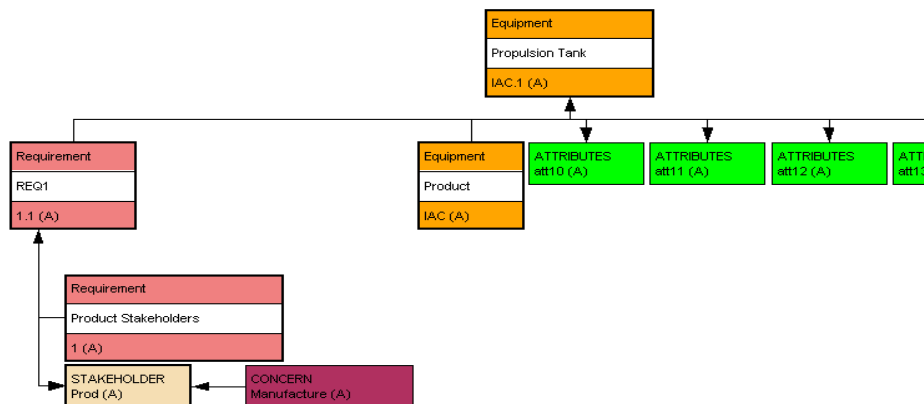


Fig. 21: Traceability among information items in Cradle.

elements. A requirement is related to an attribute, the attribute is linked to the element of the implementation model from Fig. 17.

VI. DISCUSSION

The proposed approach acknowledges that a system is composed by product and organization elements that interact along the product life cycle processes. This improves the traditional systems engineering approach. The proposed approach includes product and organization elements in the system solution whereas the traditional approach is product focused. The proposed approach covers the whole life cycle whereas the traditional approach focuses on development and operations processes.

Cradle allows stakeholder analysis, requirements analysis, functional analysis and implementation analysis to be done, via modelling, simultaneously for the product and for the organization. Cradle allows the creation of additional attributes that further describes the elements in those models. Those attributes are cross referenced by Cradle to the elements they describe. Furthermore, Cradle allows the identification and description of the impact links between those attributes. The relationships between product attributes and organization attributes are then identified and described.

For the VLM project, Cradle allowed to anticipate to the early stages of product development constraints imposed by organization attributes. For example, the dimensions of the facilities available for testing may impose the need for increasing the number of parts in the VLM. So one can adequate the product to test from the outset, not having to wait and avoiding late changes in the product. This reduces development time, reduces cost and improves product quality. On the other hand, giving the VLM product attributes the testing organization can be better planned in order to provide the measurement facilities for those attributes.

VII. CONCLUSION

The concurrent systems engineering approach was presented in a way to perform the stakeholder analysis, requirements analysis, functional analysis and implementation analysis simultaneously for product and organization elements.

The approach was performed by the use of the systems engineering software, called Cradle.

The application of the concurrent systems engineering approach using Cradle was demonstrated using the VLM example. Product and organization attributes were derived from the models and integration was obtained by capturing the relationships between those attributes.

The discussion of the approach showed the improvement on the traditional systems engineering and the contribution of automating the concurrent engineering approach.

VIII. ACKNOWLEDGEMENT

The authors would like to thank CAPES (Coordination for Supporting and Development Superior Education Personnel, www.capes.gov.br) for Jonas Bianchini Fulindi scholarship. The authors would like to thank INPE (Brazilian Institute for Space Research, www.inpe.br) for the post graduate course. The authors would like to thank IAE (Aeronautics and Space Institute, www.iae.cta.br) for the case study opportunity and 3SL (www.threesl.com) for providing Cradle, the systems engineering software used for the development of this work.

IX. REFERENCES

- [1] Loureiro, G. A systems engineering and concurrent engineering framework for the integrated development of complex products. Ph.D. thesis, Loughborough University, 1999.
- [2] Loureiro, G. A systems and concurrent engineering framework for the integrated development of space products. *Acta Astronautica*. Vol. 53, Issue 12, p. 945-961. 2003.
- [3] Loureiro, G. et al. Systems concurrent engineering of a turbo-generator. In: proceedings of the 17th ISPE International Conference on Concurrent Engineering. Cracow, Poland, 2010.
- [4] Loureiro, G. et al. Systems concurrent engineering of an electric bike. In: proceedings of the 17th ISPE International Conference on Concurrent Engineering. Cracow, Poland, 2010.
- [5] Loureiro, G. et al. Systems concurrent engineering to develop a green car. In: proceedings of the 17th ISPE International Conference on Concurrent Engineering. Cracow, Poland, 2010.
- [6] Loureiro, G. et al. Systems concurrent engineering for the conception of a hybrid vehicle. In: proceedings of the 17th ISPE International Conference on Concurrent Engineering. Cracow, Poland, 2010.
- [7] Loureiro, G. et al. Systems concurrent engineering of an electrical ground support equipment for an on-board computer. In: proceedings of the 17th ISPE International Conference on Concurrent Engineering, Cracow, Poland, 2010.
- [8] Electronic industries alliance. EIA-632: process for engineering a system. Arlinton, 1997.
- [9] European cooperation for space standardization. ECSS-E-ST-10C: space engineering: system engineering general requirements. Noordwijk: ESA, 2009.
- [10] Institute of electrical and electronics engineers. Systems engineering: application and management of the systems engineering process. New York: IEEE, 2005.
- [11] National aeronautics and space administration. Nasa systems engineering handbook. Washington, 2007.
- [12] Huang, G. Q. Design for x: concurrent engineering imperatives. Chapman Hall. London, 1996.
- [13] Alexander, C. Notes on the synthesis of the form. Harvard university press. Cambridge, USA, 1964.
- [14] Cradle. 3SL. Structure software systems Ltd. Available from: www.threesl.com. Accessed on: August 16th, 2010.
- [15] Cradle overview. Available from: <http://www.threesl.com/pages/Evaluators/startup/overview.pdf>. Accessed on: August 16th, 2010
- [16] Loureiro, G. CSE-201-4. Introduction to Systems Engineering, Class Notes. INPE. São José dos Campos – SP. 2009.

ANEXO C - DESENVOLVIMENTO INTEGRADO DE SISTEMAS - MICROONDAS

O trabalho aqui em anexo foi apresentado no curso de engenharia de sistemas, para o Programa de Especialização em Engenharia da EMBRAER.




Desenvolvimento Integrado de Sistemas

Forno de Microondas




 Edson Fonseca Eng. Eletricista	 Heron Medeiros Eng. Materials
 Ernesto Salles Eng. Controle e Automação	 Rafael Kido Eng. Eletricista
 Filipe Gorla Eng. Mecânico	 Ricardo Mimura Eng. Eletricista
	 Ricardo Motoyama Eng. Mecânico

PEE XVI – Outubro de 2010 ¹



Objetivo e Estrutura da Apresentação



Objetivo:
Apresentar o desenvolvimento de um produto utilizando conceitos modernos de Engenharia de Sistemas

Conteúdo:

- Missão
- Processos ciclo de vida
- Cenários
- Análise dos stakeholders
- Medidas de efetividade
- Requisitos
- Contexto funcional
- Contexto arquitetura
- Circunstâncias
- Modos
- Lista de eventos
- Estrutura funcional
- Modelo de comportamento
- Diagrama de atividades
- Perigos (FMECA)
- Estrutura funcional
- Estrutura física
- Matriz de alocação
- Análise Crítica
- Conclusões

2



Missão

- Diagrama de atividades
- Perigos (FMECA)
- Estrutura funcional
- Estrutura física
- Matriz de alocação
- Análise Crítica
- Conclusões

- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Análise dos stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura funcional
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões

“O sistema deve aquecer alimentos através da freqüência de ressonância das moléculas de água.”

3



Processos do ciclo de vida



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura funcional
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões



4



Cenários



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura funcional
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões

Cenários do Produto:

Uso e Operação:



Uso e operação

- Stand-by
- Introdução do alimento
- Programação
- Alimento em preparo
- Retirada do alimento
- Limpeza do forno

Descarte:



Descarte

- Avaliação de sistema e subsistemas
- Desmontagem
- Triagem material
- Tratamento de material
- Descarte ou reuso



5



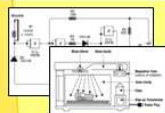
Cenários



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura funcional
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões

Cenários da Organização

Desenvolvimento



Desenvolvimento

- Concepção
- Especificação de Material e Matéria Prima
- Design Preliminar
- Simulação Computacional
- Protótipo
- Ensaio
- Design Final
- Especificação dos Processos de Manufatura
- Cálculo de Custo

Manufatura e Montagem



Manufatura

- Recebimento de matéria-prima
- Recebimento de componentes
- Recebimento sub-sistemas
- Montagem sub-sistemas
- Integração sistema
- Inspeção e testes

Testes e calibração



Teste e calibração

- Teste funcional
- Teste de EMC/EMI
- Teste de vazamento de RF
- Teste de resistência mecânica

Distribuição



Distribuição

- Estoque
- Transporte
- Entrega
- Relacionamento com o Cliente

Vendas



Vendas

- Negociação
- Transporte
- Entrega
- Relacionamento com o cliente

Descarte



Descarte

- Recebimento de material
- Estoque
- Processo de reciclagem
- Reintegração na manufatura



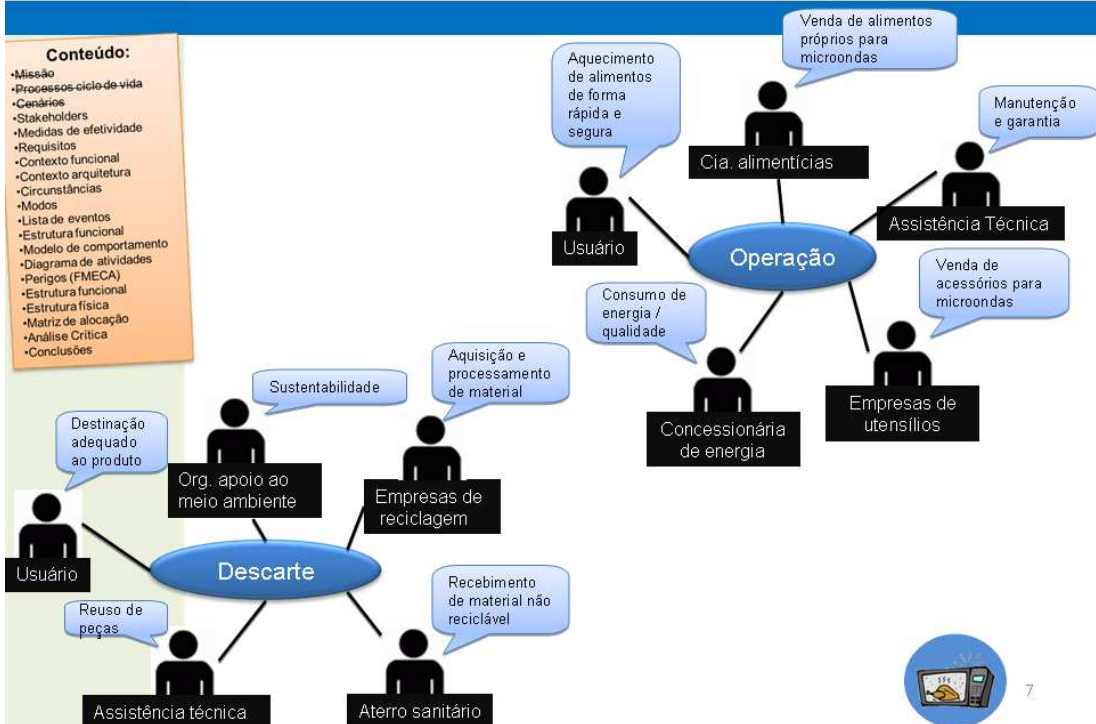
6



Análise dos stakeholders



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura funcional
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões



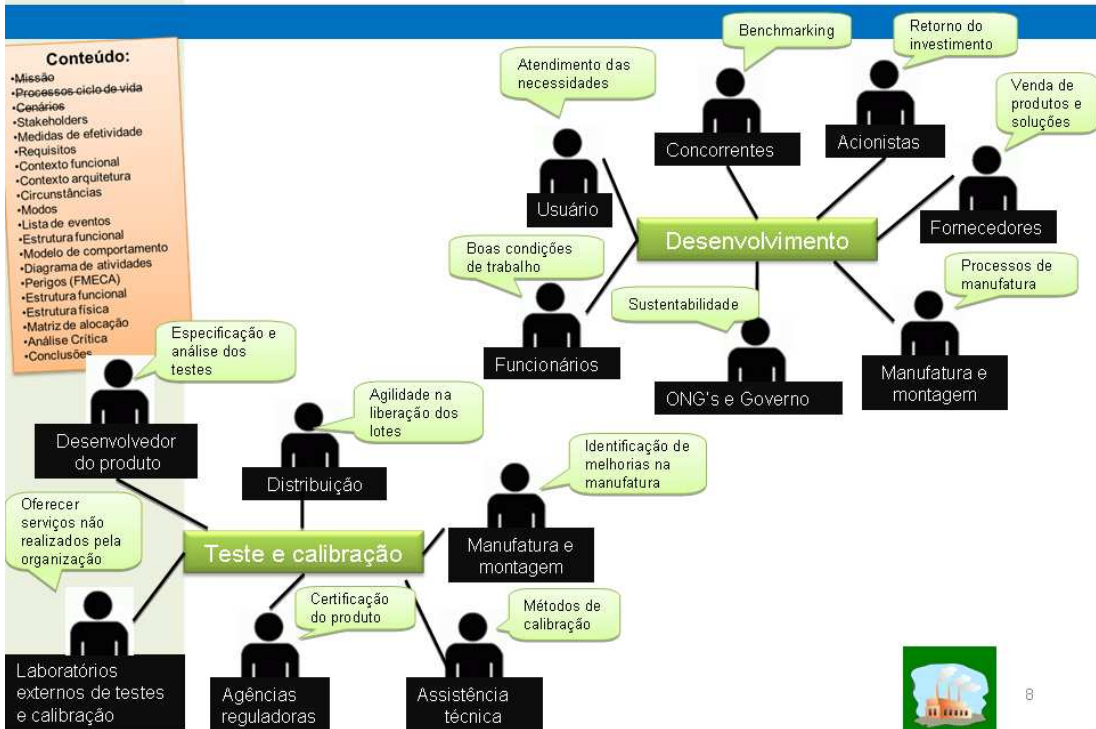
7



Análise dos stakeholders



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura funcional
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões



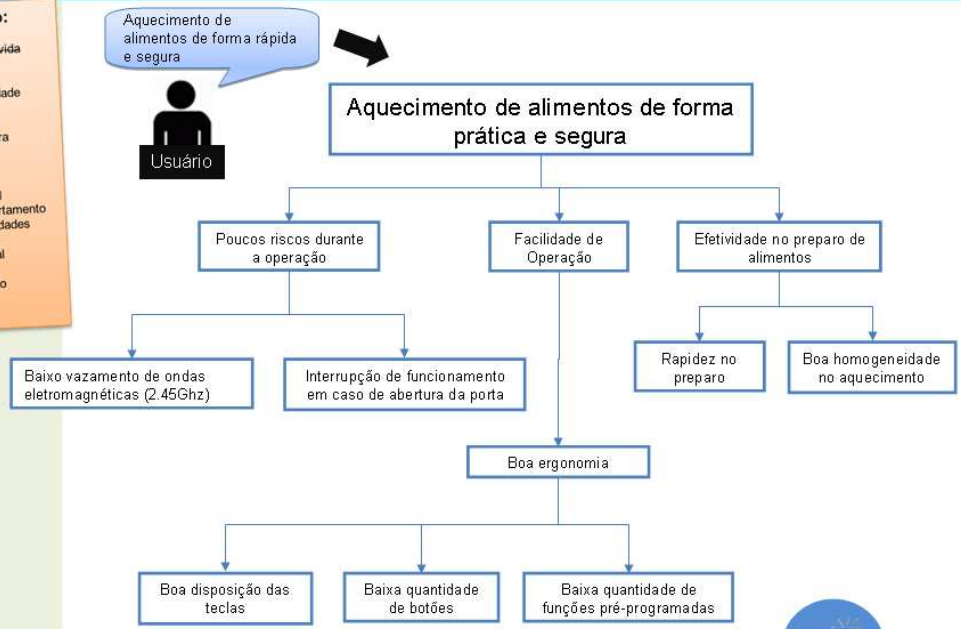
8



Medidas de efetividade



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura funcional
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões



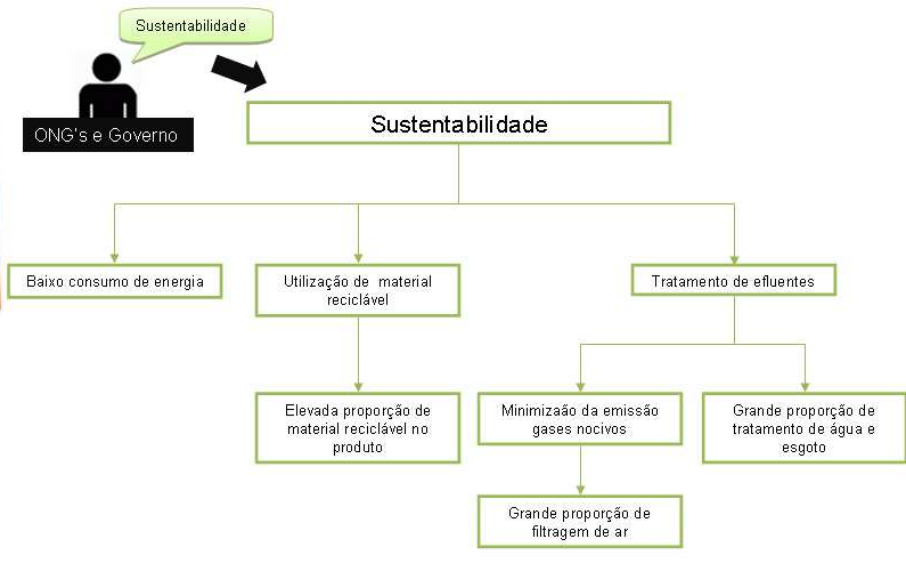
9



Medidas de efetividade



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura funcional
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões



10



Requisitos



Conteúdo:

- Missão
- Processos-ciclo-de-vida
- Cenários
- Stakeholders
- Medidas-de-efetividade
- Requisitos
- Contexto funcional
- Contexto arquitetura
- Circunstâncias
- Modos
- Lista de eventos
- Estrutura funcional
- Modelo de comportamento
- Diagrama de atividades
- Perigos (FMECA)
- Estrutura funcional
- Estrutura física
- Matriz de alocação
- Análise Crítica
- Conclusões

Interesse: Aquecimento do alimento de modo seguro.

Aquecimento de alimentos de forma rápida e segura



Usuário

Estímulo: O usuário deve ser capaz de interromper a operação em qualquer momento

Resposta: O sistema deve interromper o funcionamento em um tempo não maior que 0.5 segundos, sempre que solicitado pelo usuário

Interesse: Aquecimento do alimento de modo rápido.

Estímulo: O usuário deve ser capaz de aquecer o alimento rapidamente

Resposta: O sistema deve aquecer alimentos descongelados até 50°C (+/- 2°C) em não mais que 5 minutos na potência máxima e condições normais de temperatura (9-40°C) e umidade (10-90%)



11



Requisitos



Conteúdo:

- Missão
- Processos-ciclo-de-vida
- Cenários
- Stakeholders
- Medidas-de-efetividade
- Requisitos
- Contexto funcional
- Contexto arquitetura
- Circunstâncias
- Modos
- Lista de eventos
- Estrutura funcional
- Modelo de comportamento
- Diagrama de atividades
- Perigos (FMECA)
- Estrutura funcional
- Estrutura física
- Matriz de alocação
- Análise Crítica
- Conclusões

Sustentabilidade



Interesse: Sustentabilidade.

Estímulo: A agência ambiental deve ser capaz de fiscalizar o tratamento de gases nocivos

Resposta: O organização de desenvolvimento deve garantir que a saída de ar retenha não menos que 97% das partículas tóxicas em condições normais de produção



12



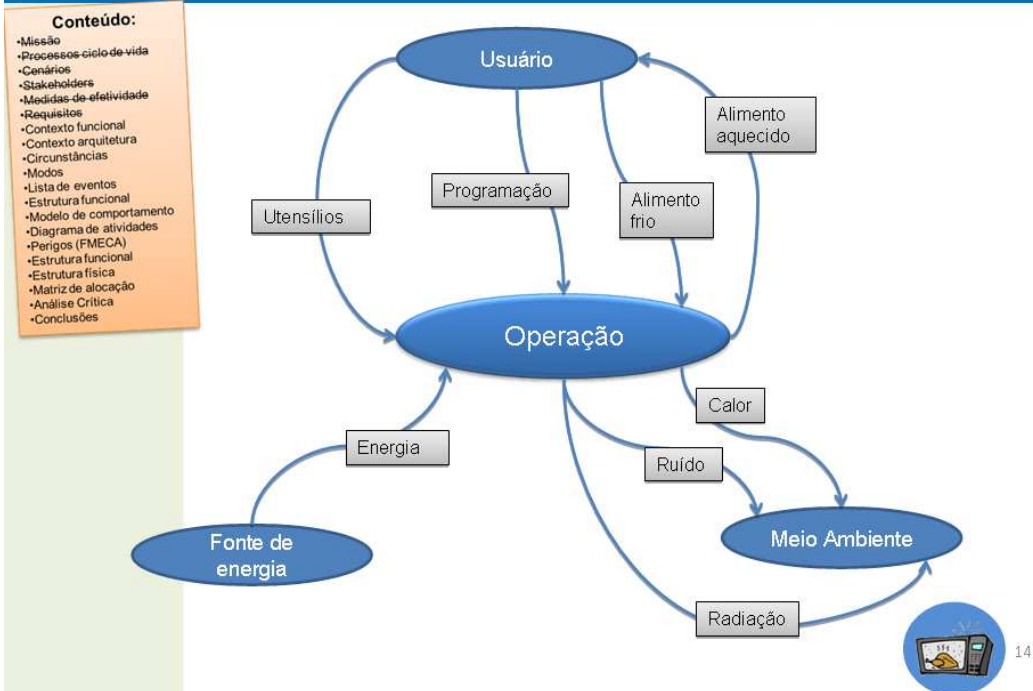
Requisitos



Conteúdo:	Requisitos do sistema	Função/ Desempenho/ Condição	Produto/ Organização	Mandatário/ Desejável	Status	Verificação	
						Método	Procedimento
<ul style="list-style-type: none"> •Missão •Processos-ciclo-de-vida •Cenários •Stakeholders •Medidas-de-efetividade •Requisitos •Contexto funcional •Contexto arquitetura •Circunstâncias •Modos •Lista de eventos •Estrutura funcional •Modelo de comportamento •Diagrama de atividades •Perigos (FMECA) •Estrutura física •Matriz de alocação •Análise Crítica •Conclusões 	1. Aquecimento de alimentos de forma rápida e segura						
	1.1. Sistema de deve interromper o processo...	Função	Produto	Mandatário	TBD	demonstração	TBD
	1.1.1 ...em um tempo não maior que 0.5 segundos...	Desempenho					
	1.1.2 ...sempre que solicitado pelo usuário.	Condição					
	1.2. Sistema deve aquecer o alimento descongelado...	Função	Produto	Mandatário	TBD	teste	TBD
	1.2.1 ...até 50°C (+/- 2°C) em não mais que 5 minutos...	Desempenho					
	1.2.2 ...na potência máxima e condições normais de temperatura (9-40°C) e umidade (10-90%).	Condição					
	2. Sustentabilidade						
	2.1. A Organização deve tratar efluentes gasosos...	Função	Organização	Desejável	TBD	teste	TBD
	2.2.1 ...garantindo que a saída de ar retenha não menos que 97% das partículas tóxicas...	Desempenho					
2.2.2 ...em condições normais de produção.	Condição						



Contexto funcional

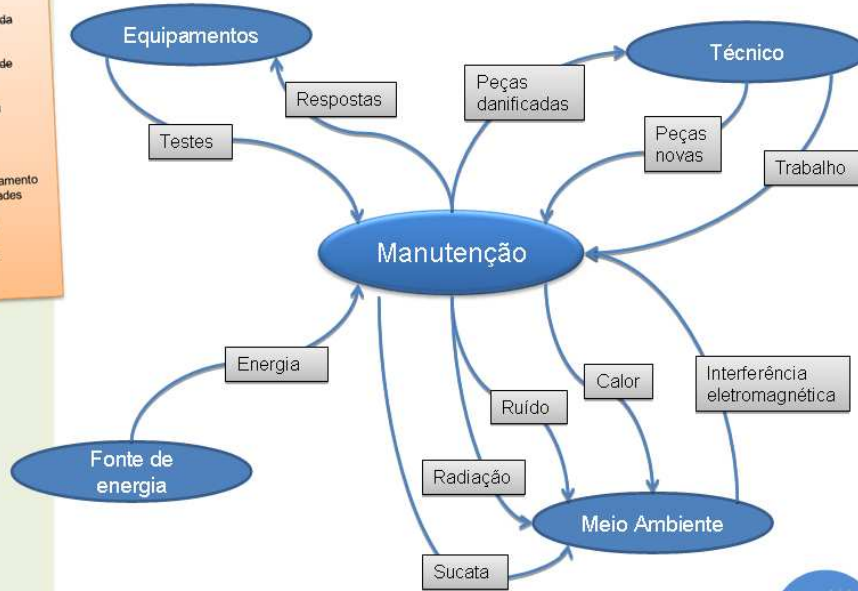




Contexto funcional



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura funcional
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões



15



Contexto funcional



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura funcional
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões



16



Contexto funcional



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura funcional
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões



17



Contexto de arquitetura



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura funcional
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões



18



Contexto de arquitetura



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura funcional
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões



19



Contexto de arquitetura



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura funcional
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões



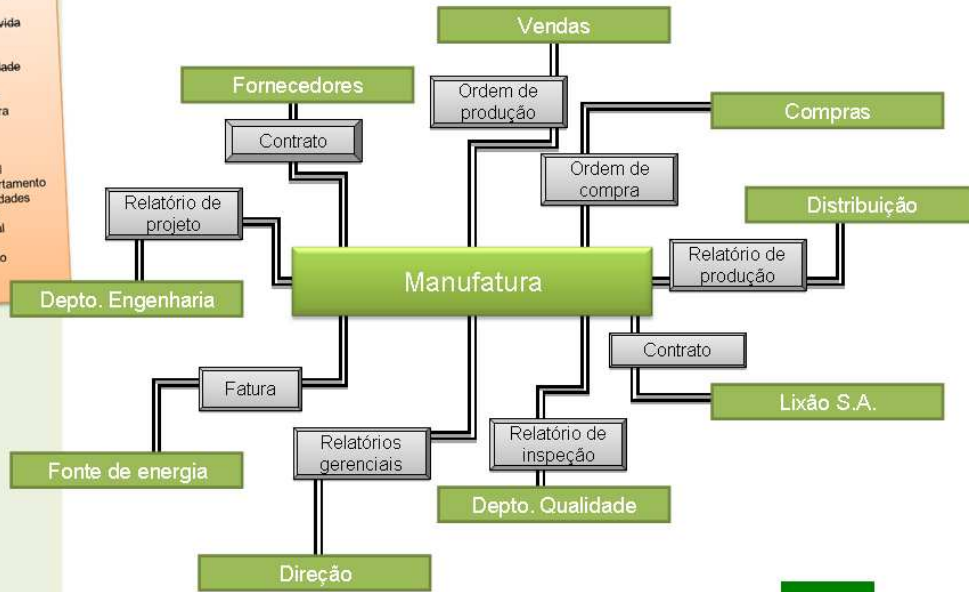
20



Contexto de arquitetura



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura funcional
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões



21



Circunstâncias



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura funcional
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões

- Normal
- Sobretensão
- Subtensão
- Frequência errada
- Black-out



22



Circunstâncias



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões



Circunstâncias

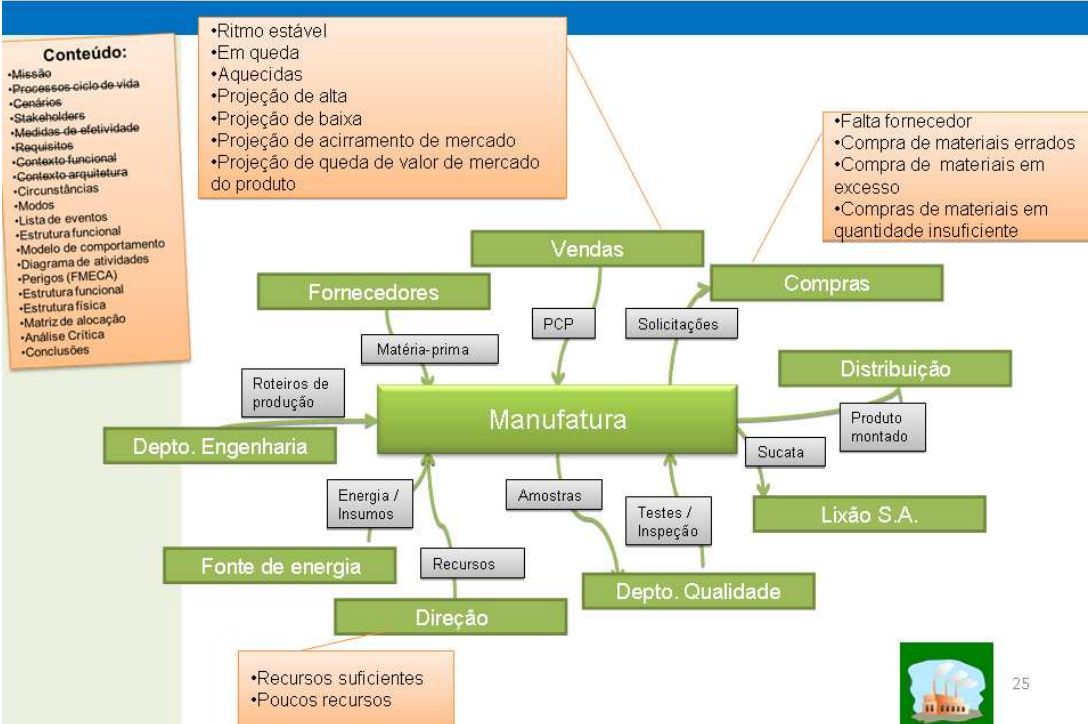


- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões





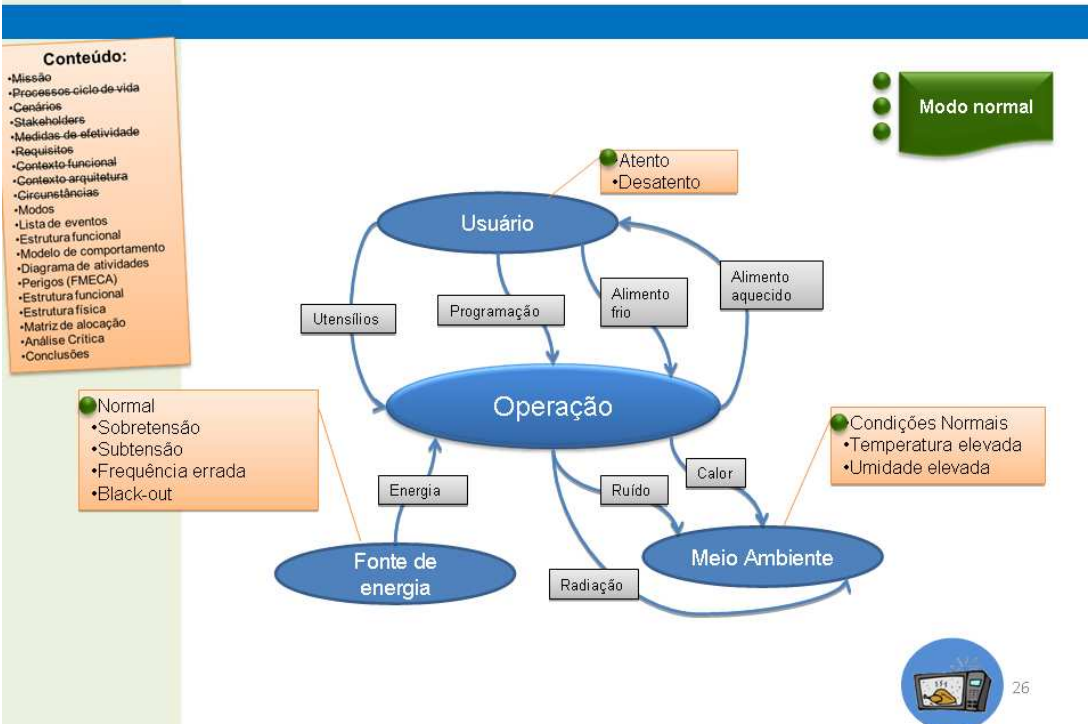
Circunstâncias



25



Modos



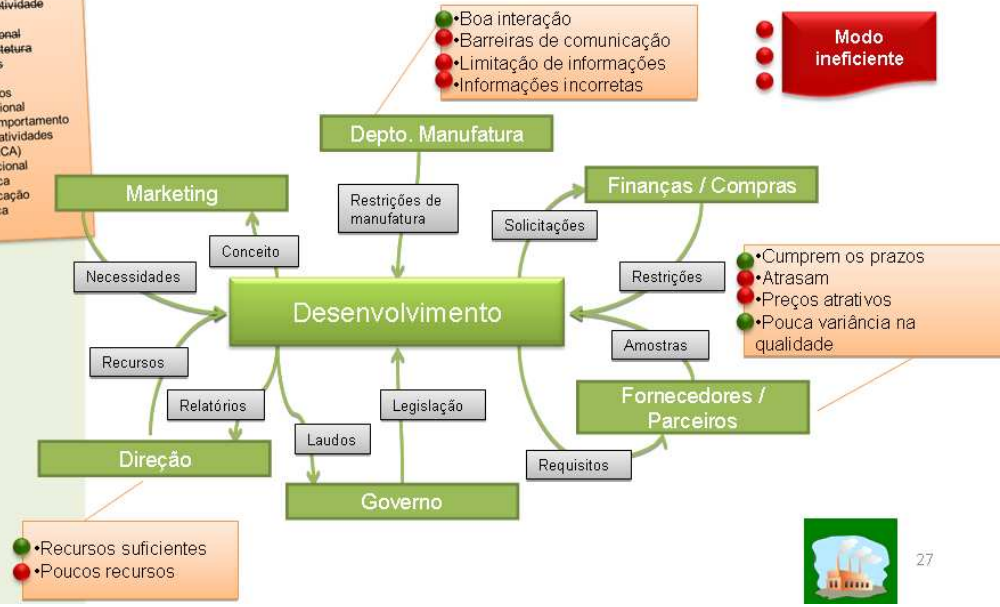
26



Modos



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura funcional
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões



Lista de eventos



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura funcional
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões

Produto em Operação

1. Usuário acessa interior do forno	1. O sistema provê acesso ao interior do forno
2. Usuário fornece informação de modo de preparo	2. O sistema processa informação de modo de preparo
3. Usuário aguarda pedido de confirmação	3. O sistema processa informação de modo de preparo
4. Usuário envia comando de início de preparo	4. O sistema recebe comando de início de preparo
5. Usuário aguarda preparo	5. Sistema executa preparo
6. Usuário recebe informação de término de preparo	6. Sistema provê informação de término de preparo
7. Usuário fecha compartimento	7. Sistema identifica porta fechada





Lista de eventos



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura funcional
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões

Organização de Desenvolvimento

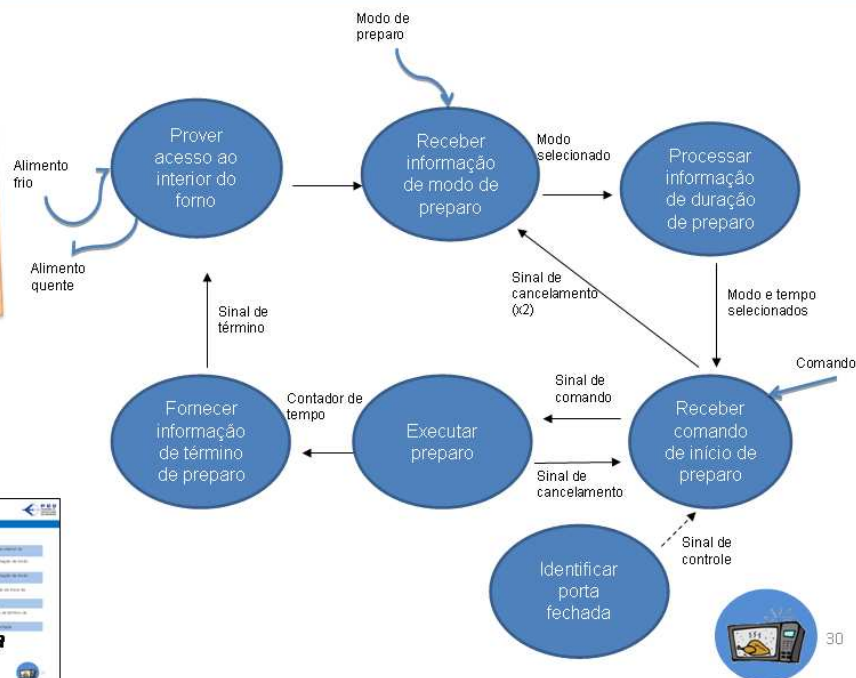
- | | |
|--|--|
| 1. O MKT envia os necessidades dos clientes para a O.D. | 1. A Organização de Desenvolvimento analisa as necessidades dos clientes |
| 2. A manufatura envia feedback de outros produtos | 2. A O.D. define os requisitos do produto |
| 2. Os fornecedores/parceiros enviam informações à O.D. | 2. A O.D. define os requisitos do produto |
| 2. O órgão certificador impõe requisitos de certificação | 2. A O.D. define os requisitos do produto |
| 2. O MKT envia informações para a O.D. | 2. A O.D. define os requisitos do produto |
| 3. Os fornecedores/parceiros enviam informações à O.D. | 3. A O.D. define os requisitos dos sub-sistemas |
| 3. A manufatura envia informações à O.D. | 3. A O.D. define os requisitos dos sub-sistemas |
| 4. Os fornecedores/parceiros enviam informações à O.D. | 4. A O.D. define a arquitetura |
| 4. A manufatura envia informações à O.D. | 4. A O.D. define a arquitetura |
| 5. Os fornecedores/parceiros enviam informações à O.D. | 5. A O.D. realiza a prototipagem e a integração do sistema |
| 5. A manufatura envia informações à O.D. | 5. A O.D. realiza a prototipagem e a integração do sistema |
| 6. O órgão certificador impõe requisitos de certificação | 6. A O.D. certifica o produto |
| 7. Os fornecedores/parceiros enviam informações à O.D. | 7. A O.D. define a manufatura |
| 7. A manufatura envia informações à O.D. | 7. A O.D. define a manufatura |



Estrutura funcional



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura funcional
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões



Lista de eventos

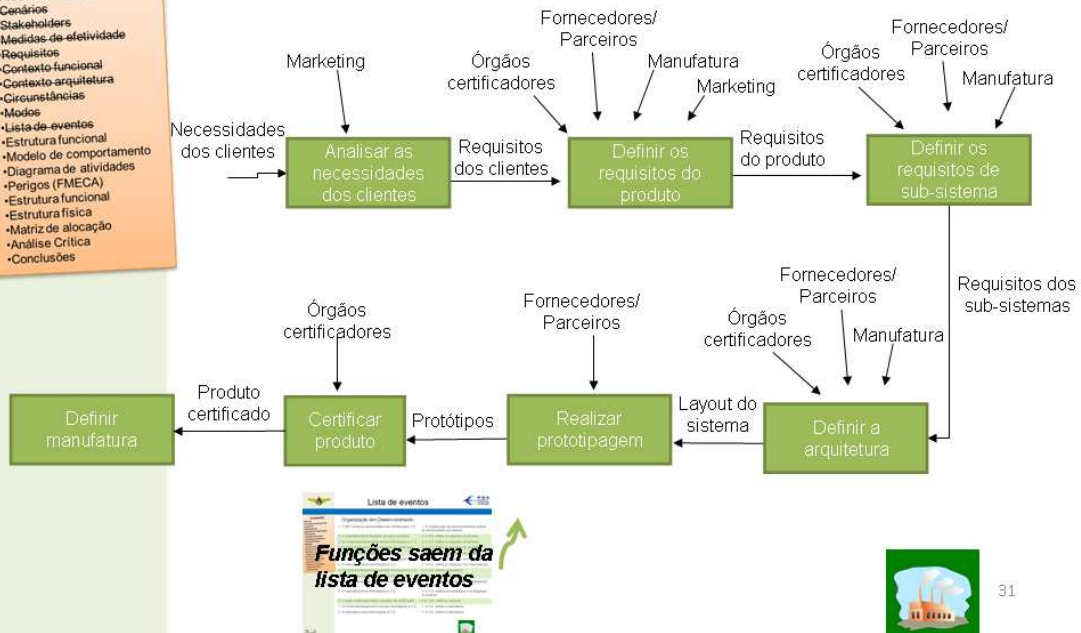
Funções saem da lista de eventos



Estrutura funcional



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões



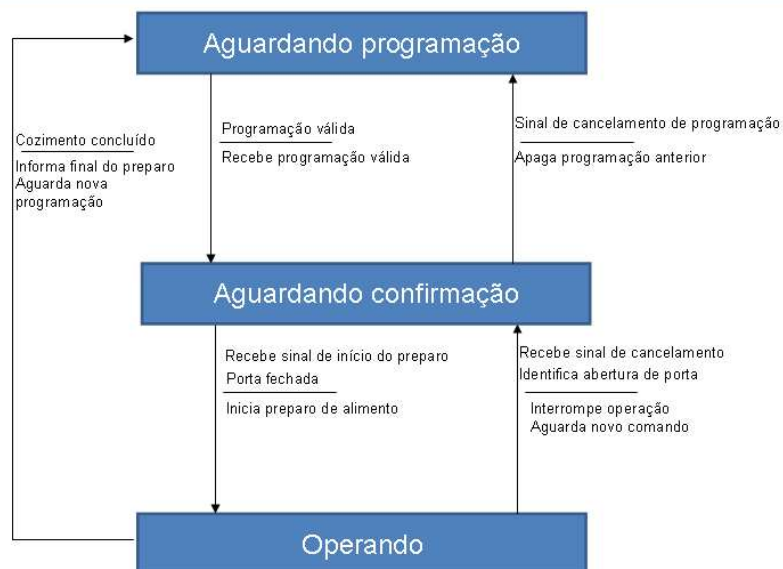
31



Modelo de comportamento



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões



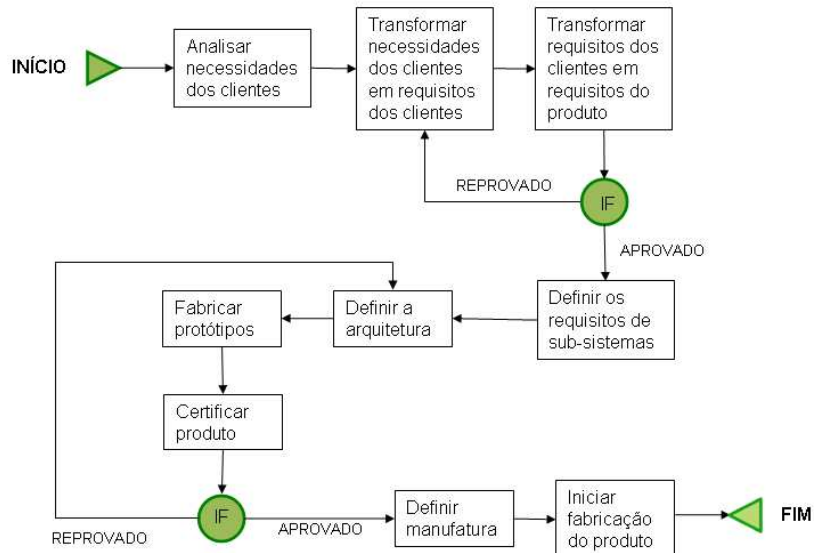
32



Diagrama de atividades



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura funcional
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões



33



Perigos (FMECA)



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura funcional
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões

	TIPO	PERIGO	CONSEQUÊNCIA	GRAV (1-5)	CAUSAS	PROB (1-5)	RISCO	FUNÇÕES
PRODUTO	circunstância	Erro do usuário - acionamento do forno com objetos metálicos em seu interior	Risco de incêndio e danos ao forno	5	efeito fotoelétrico em objetos metálicos	2	10	proteção
	passaporte	Impossibilidade de programação ou programação equivocada	Mal preparo do alimento ou impossibilidade de preparo	4	Falha das teclas	1	4	corretiva
	não função	Equipamento não aquece alimento	Não prepara o alimento	3	Módulo de potência queimada	2	6	deteção
ORGANIZAÇÃO	circunstância	Fornecedor não atende demanda	Prejudica fluxo de produção	2	Falta de peças	4	8	protetiva
	passaporte	Falha de comunicação entre desenvolvimento e manufatura	Dificuldades na manufatura	8	Barreira na comunicação	3	8	preventiva
	não função	Quebra de maquinário da linha de produção	Para linha de produção	3	Máquina quebrada	2	6	preventiva



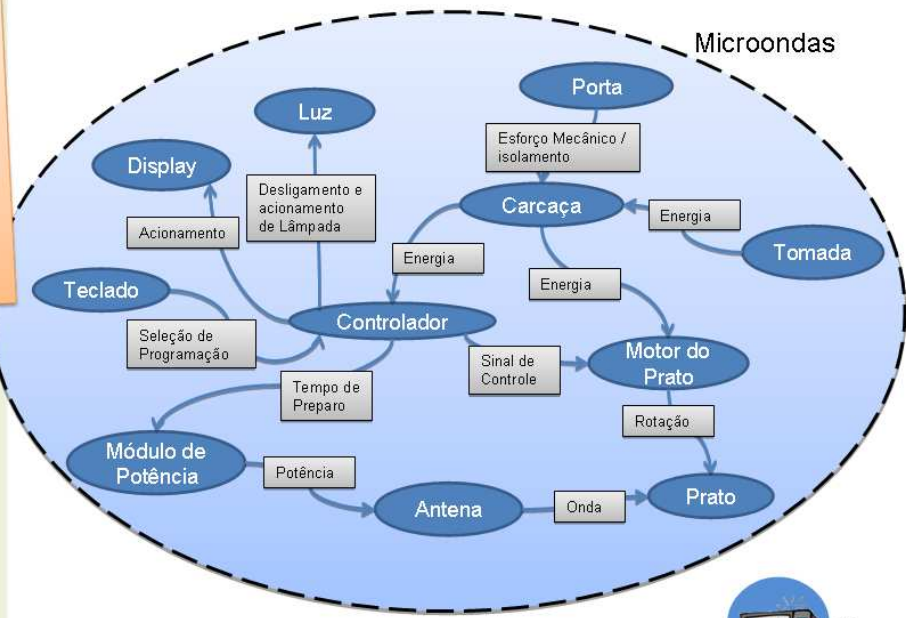
34



Fluxo de arquitetura



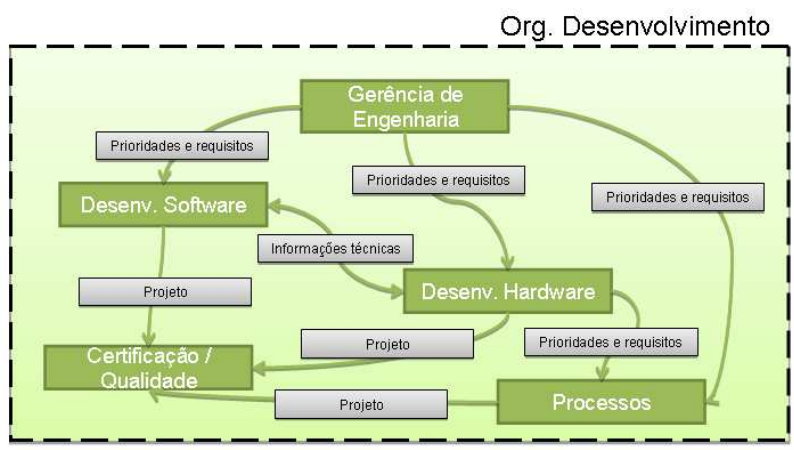
- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões



Fluxo de arquitetura



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões

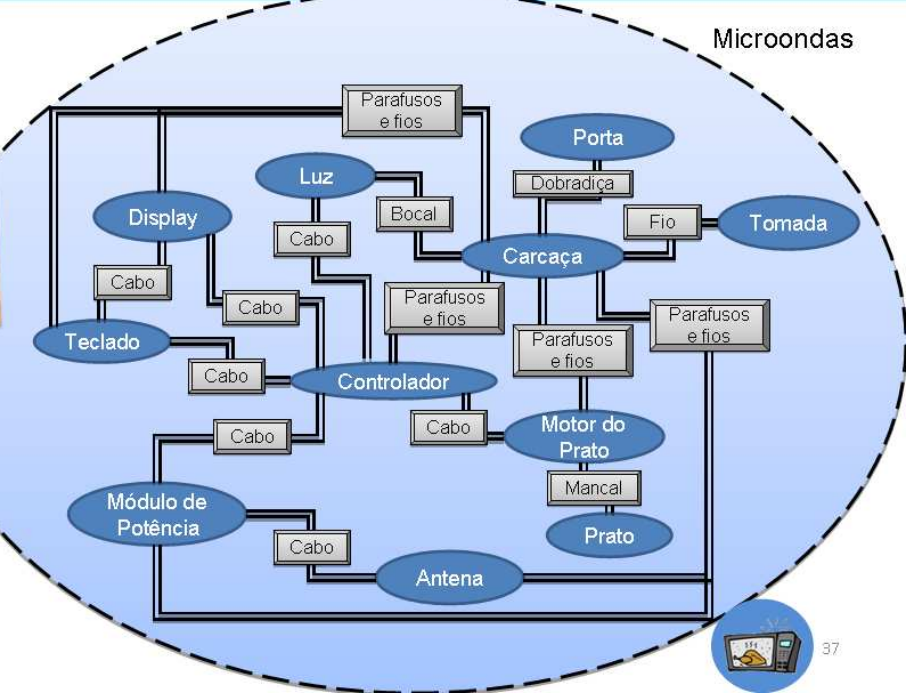




Interconexão de arquitetura



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura funcional
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões



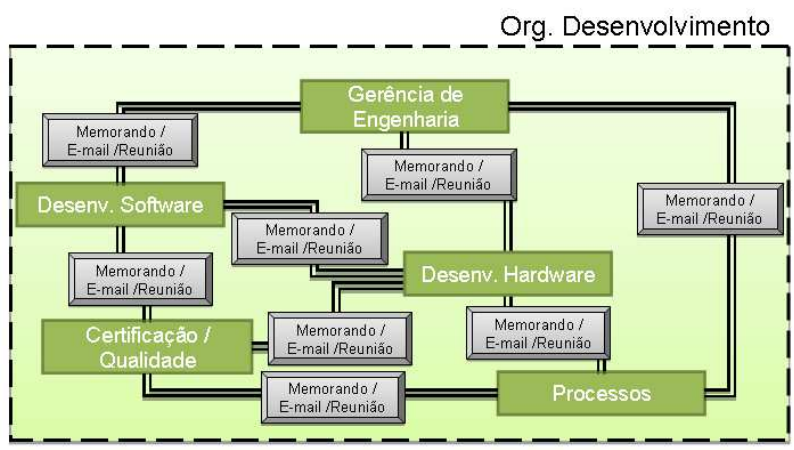
37



Interconexão de arquitetura



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura funcional
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões



38



Matriz de alocação



- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura funcional
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões



Partes dos diagramas de arquitetura



Funções da estrutura funcional

	CONTROLADOR	ANTENA	MÓDULO DE POTENCIA	MOTOR DO PRATO	TECLADO	DISPLAY	CARÇAÇA	PORTA	TOMADA	PRATO	LUZ
Prover acesso ao interior do forno							1	1		1	
Receber informação de modo de preparo	1				1	1			1		
Processar informação de duração de preparo	1								1		
Receber comando de inicio de preparo	1				1	1			1		
Executar preparo	1	1	1	1			1	1	1	1	1
Fornecer informação de término de preparo	1					1			1		1
Identificar porta fechada	1						1	1	1		



39

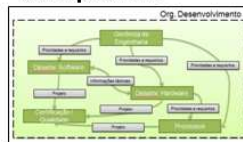


Matriz de alocação

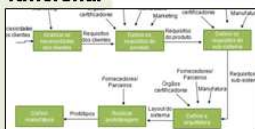


- Conteúdo:**
- Missão
 - Processos ciclo de vida
 - Cenários
 - Stakeholders
 - Medidas de efetividade
 - Requisitos
 - Contexto funcional
 - Contexto arquitetura
 - Circunstâncias
 - Modos
 - Lista de eventos
 - Estrutura funcional
 - Modelo de comportamento
 - Diagrama de atividades
 - Perigos (FMECA)
 - Estrutura funcional
 - Estrutura física
 - Matriz de alocação
 - Análise Crítica
 - Conclusões

Partes dos diagramas de arquitetura



Funções da estrutura funcional



	Gerência de eng	Desenv. Hardware	Desenv. Software	Processos	Certificação / Qualidade
Analisar as necessidades dos clientes	1	1	1		1
Definir os requisitos do produto	1	1	1	1	1
Definir os requisitos do sub-sistema		1	1	1	
Definir a arquitetura		1		1	
Realizar prototipagem		1		1	1
Certificar produto					1
Definir manufatura	1	1		1	1



40



Análise Crítica e Conclusões



Conteúdo:

- Missão
- Processos ciclo de vida
- Cenários
- Stakeholders
- Medidas de efetividade
- Requisitos
- Contexto funcional
- Contexto arquitetura
- Circunstâncias
- Modos
- Lista de eventos
- Estrutura funcional
- Modelo de comportamento
- Diagrama de atividades
- Perigos (FMECA)
- Estrutura funcional
- Estrutura física
- Matriz de alocação
- Análise Crítica e conclusões

Através do modelo proposto foi possível entender o desenvolvimento de um produto utilizando os conceitos modernos de Engenharia de Sistemas. Os conceitos fundamentais foram sendo desenvolvidos no decorrer do trabalho possibilitando uma visão geral de como utilizar o modelo.

Foram encontradas algumas dificuldades na realização do trabalho com devido à pouca literatura disponível sobre o assunto. No entanto, concluiu-se que, na prática, desenvolvendo o trabalho o aprendizado foi muito mais significativo.

Com a metodologia de Engenharia de Sistemas, observamos que há um aumento do entendimento acerca do produto, com maior esforço de desenvolvimento na fase inicial que é compensado por menos retrabalhos nas fases finais.

Seria MUITO útil ter um **glossário dos termos** de DIS (contexto, arquitetura, stakeholder, circunstância, modo, estado, etc) para facilitar o entendimento do fluxo do trabalho



Análise Crítica e Conclusões



Conteúdo:

- Missão
- Processos ciclo de vida
- Cenários
- Stakeholders
- Medidas de efetividade
- Requisitos
- Contexto funcional
- Contexto arquitetura
- Circunstâncias
- Modos
- Lista de eventos
- Estrutura funcional
- Modelo de comportamento
- Diagrama de atividades
- Perigos (FMECA)
- Estrutura funcional
- Estrutura física
- Matriz de alocação
- Análise Crítica e conclusões

Proposta de melhoria: template com simbologia mnemônica para facilitar entendimento do trabalho

É importante diferenciar claramente quando se trata de PRODUTO ou ORGANIZAÇÃO. Fizemos isso através de cores e formatos padronizados ao longo deste trabalho.



Representamos o fluxo entre elementos de PRODUTO ou ORGANIZAÇÃO através de setas padronizadas indicando origem, destino e o que flui



Stakeholder necessariamente refere-se a pessoas ou organização de pessoas. Pode estar relacionado a PRODUTO ou ORGANIZAÇÃO. Representamos da forma mais direta possível



Representamos diagrama de arquitetura através de barras duplas que representam as conexões físicas entre componentes de PRODUTO ou ORGANIZAÇÃO

