

ANA LUCIA DA SILVA PASTORELLI

METODOLOGIA PARA APLICAÇÃO
DE REUSO E REENGENHARIA NOS SOFTWARES DE
AQUISIÇÃO E REDUÇÃO DE DADOS
DE UM TÚNEL DE VENTO

Tese apresentada à Faculdade de Engenharia
do Campus de Guaratinguetá, Universidade
Estadual Paulista, para a obtenção do título
de Doutor em Engenharia Mecânica na área
de Projetos e Materiais.

Orientador: Prof. Dr. Fernando Silva de Araújo Porto

Guaratinguetá
2007

P293m	Pastorelli, Ana Lucia da Silva Metodologia para aplicação de reuso e reengenharia nos softwares de aquisição e redução de dados de um túnel de vento. / Ana Lucia da Silva Pastorelli. - Guaratinguetá : [s. n.], 2007 334f. : il. Inclui apêndice Bibliografia: f. 280-290 Tese (doutorado) – Universidade Estadual Paulista, Faculdade de Engenharia do Campus de Guaratinguetá, 2007. Orientador: Prof. Dr. Fernando Silva de Araújo Porto 1. Projeto de Engenharia I. Título CDU 62.001.63
-------	---

DADOS CURRICULARES

ANA LUCIA DA SILVA PASTORELLI

NASCIMENTO	15.11.1963 – TAUBATÉ / SP
FILIAÇÃO	Rubens Negrini Pastorelli Maria Célia da Silva Pastorelli
1981/1982	Curso de Graduação: Tecnólogo em Processamento de Dados Universidade de Taubaté – UNITAU – Taubaté - SP
1985/1988	Curso de Graduação: Bacharel em Ciências Contábeis Universidade de Taubaté – UNITAU – Taubaté - SP
1993/1998	Curso de Pós-Graduação em Computação, nível de Mestrado, no Instituto Tecnológico da Aeronáutica – ITA – São José dos Campos – SP.
2002/2007	Curso de Pós-Graduação em Engenharia Mecânica, nível de Doutorado na Faculdade de Engenharia do Campus de Guaratinguetá da UNESP.

Dedico de modo especial, à minha filha Luciana, aos meus pais Rubens e Célia e a meu avô Luiz (em memória), que foram os grandes incentivadores deste trabalho.

AGRADECIMENTOS

A DEUS, pela benção e amor para a realização deste trabalho.

A minha filha Luciana Pastorelli pela cumplicidade e confiança de que as dificuldades e limitações poderiam ser superadas.

Aos meus pais Rubens e Célia e irmãos Luiz Fernando, Rubens e Flávio, pelos votos de confiança e incentivo aos estudos.

Ao meu namorado Pedro Sérgio de Oliveira, pelo companheirismo e pela compreensão das longas horas de minha ausência.

À Universidade Estadual Paulista – UNESP, campus da Faculdade de Engenharia de Guaratinguetá - FEG, pela oportunidade de realização deste trabalho.

Ao orientador e professor doutor Fernando Silva de Araújo Porto, por acreditar no meu trabalho, e pelo empenho pessoal, não mensurando esforços para sua realização.

Às funcionárias da Biblioteca do Campus de Guaratinguetá pela dedicação, presteza e principalmente pela vontade de ajudar.

Às secretárias da pós-graduação Regina e Elisa pela dedicação e alegria no atendimento.

Aos especialistas em ensaios em túnel, da Subdivisão Aerodinâmica (ASA-L) do Comando-Geral de Tecnologia Aeroespacial (CTA), pela grande contribuição nas abordagens que fundamentaram este estudo.

Aos colegas e amigos do Instituto de Fomento e Coordenação Industrial (IFI), Divisão de Certificação de Produto Aeroespacial (CPA), pela ajuda na organização das informações constantes neste trabalho.

Aos amigos Ricardo Alvim, Matsuo, Willian Limonge, Takeshi, Maria Luisa, Hering, Sérgio Rebelo, Diego Palma, Barrochelo, Baltazar e Olivério, pela atuação direta ou indireta na condução deste trabalho.

A todos os demais professores, amigos, colegas e funcionários que direta ou indiretamente participaram da realização deste trabalho.

PASTORELLI, A. L. S. **Metodologia para aplicação de reuso e reengenharia nos softwares de aquisição e redução de dados de um túnel de vento**. 2007. 334f. Tese (Doutorado em Engenharia Mecânica) – Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, 2007.

RESUMO

O trabalho tem por objetivo recuperar a usabilidade dos processos de ensaios em túnel de vento por meio de recursos computacionais, visando a atualização e revitalização do sistema de software através de recursos tecnológicos, interfaces, aplicativos de apoio avançados e da implantação de práticas de engenharia e qualidade de software. Neste trabalho de tese fez-se estudo de práticas, métodos, processos e técnicas de engenharia de software de modo a aplicá-las adequadamente no ciclo de desenvolvimento do projeto do sistema de software, considerando o conceito atual de software que abrange os programas de computadores, a documentação associada e os dados necessários para que os softwares de aquisição e redução de dados operem corretamente e continuamente durante a modernização do sistema de medidas do túnel de vento. Foi dado enfoque na melhoria da qualidade dos produtos e dos processos gerenciais e técnicos, considerando as ações das metodologias tradicionais e ágeis adequadas ao escopo e ao grupo de projeto. Neste trabalho, para atingir a abrangência do conceito de software e manter a funcionalidade do sistema de software durante a modernização do túnel de vento é gerada a metodologia para aplicação de reuso e reengenharia nos software de aquisição e redução de dados de um túnel de vento, denominada Metodologia Especial Tradicional e Ágil de aplicação em Projeto Legado de software em Inovação – METAPLI. Com a implantação desta metodologia de projeto de engenharia de software se provê a organização dos processos gerenciais e técnicos, se gera a documentação de todo projeto do sistema de software e se preserva o conhecimento dos processos de engenharia.

PALAVRAS-CHAVE: Metodologia, Engenharia de Software, Túnel de Vento, Aquisição de Dados, Redução de Dados, Reuso, Reengenharia.

PASTORELLI, A. L. S. **Methodology to application reuse and reengineering to data acquisition and reduction software development in a wind tunnel**. 2007. 334f. Thesis (Doctorate in Mechanical Engineering) - Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, 2007.

ABSTRACT

This work focuses on recovering the usability of the wind tunnel tests processes by means of computational resources in order to updating and revitalizing the software system through the technological resources, interfaces, advanced applicative of the aid and the establishing of software engineering and software quality practices. On this thesis' work they were performed studies of practices, methods, processes and software engineering techniques in a way of applying them adequately on the project development cycle of the software system, considering the software current concept which recovers the computer programs, the associated documentation and the necessary data in a way that the data acquisition and reduction software operates correctly and continuously along the wind tunnel measurement system's modernization. The focus was on quality improvement of products and managerial and technical processes, considering the traditional and agile methodologies actions, which are adequate to scope and to project group. On this work, to achieve the software concept wideness, keeping the software system functionality along the wind tunnel modernization, it was generated the methodology in order to apply the reuse and reengineering of data acquisition and reduction software of a wind tunnel, which is denominated "Traditional and Agile Special Methodology for application in Project Legacy of software in Innovation - METAPLI". The implantation of this project methodology of software engineering proves the managerial and technical processes organization, generates the documentation of the whole software system project and preserves the knowledge of engineering processes.

KEYWORDS: Methodology, Software Engineering, Wind Tunnel, Data Acquisition, Data Reduction, Reuse, Reengineering.

LISTA DE FIGURAS

FIGURA 2.1 – Instalações do túnel de vento – TA-2 do CTA (IAE, 2005).....	14
FIGURA 2.2 – Ponte do rio pinheiros e plataforma de petróleo da PETROBRAS (IAE, 2005).....	15
FIGURA 2.3 – Ônibus da Itapemirim e aeronave EMB-145 (IAE, 2005)	15
FIGURA 2.4 – Veículo lançador de satélite e antena parabólica (IAE, 2005)	15
FIGURA 2.5 – Aeronave AMX e aeronave CBA-123 (IAE, 2005)	16
FIGURA 3.1 – Níveis de maturidade do SEI	21
FIGURA 3.2 – Modelo clássico ou modelo em cascata	29
FIGURA 3.3 – Modelo de processo incremental	32
FIGURA 3.4 – Modelo de prototipação	33
FIGURA 3.5 – Modelo de processo em espiral	36
FIGURA 3.6 – Arquitetura geral do RUP	42
FIGURA 3.7 – Modelo genérico de solução de problemas (Narayanan, 2001)	59
FIGURA 4.1 – Metodologia especial tradicional e ágil para aplicação em projeto legado de software em inovação - METAPLI	63
FIGURA 4.2 – Diagrama esquemático da aquisição de dados com a Plataforma HP	68
FIGURA 4.3 – Ciclo de vida de desenvolvimento de Software com ações do RUP	75
FIGURA 4.4 – Fluxo de trabalho RUP do projeto de sistema de software do TA-2	80
FIGURA 4.5 – Comparação dos riscos entre o ciclo iterativo e em cascata (KRUCHTEN, 2000).....	87
FIGURA 4.6 – Modelo geral do diagrama da tartaruga (AFS, 2005).....	88
FIGURA 5.1 – Esforços aerodinâmicos sobre o corpo e ângulos de ataque (REIS, 2000).....	99
FIGURA 5.2 – Desenho esquemático dos dispositivos e instrumentações do TA-2 (IAE, 2005)	100

FIGURA 5.3 – Montagem do modelo aerodinâmico na seção de ensaios do TA-2 (IAE, 2005)	101
FIGURA 5.4 – Sistema de medidas e seus subsistemas.....	105
FIGURA 5.5 – Motor e hélice do TA-2 (IAE, 2005).....	106
FIGURA 5.6 – Balança externa do túnel de vento (IAE, 2005)	110
FIGURA 5.7 – Cruz de calibração do TA-2 (REIS, 2000).....	112
FIGURA 5.8 – Arquitetura geral do subsistema mastro	114
FIGURA 5.9 – Arquitetura geral do subsistema de sensoramento	116
FIGURA 5.10 – Visualização com tinta (IAE, 2005)	118
FIGURA 5.11 – Visualização com fumaça (IAE, 2005)	118
FIGURA 5.12 – Arquitetura típica de um sistema de aquisição de dados.....	120
FIGURA 5.13 – Fluxograma de aplicação de reuso com requisitos estáveis	126
FIGURA 5.14 – Componentes do sistema DAQ e SCXI para aquisição de dados (NATIONAL INSTRUMENTS, 1998).....	133
FIGURA 6.1 – Representação geral dos modelos e artefatos do sistema de software.....	147
FIGURA 6.2 – Diagrama da tartaruga do processo de desenvolvimento do software de aquisição de dados	151
FIGURA 6.3 – Diagrama da tartaruga do processo de aplicação de software na pré-redução de dados	152
FIGURA 6.4 – Diagrama da tartaruga do processo de desenvolvimento do software de redução de dados	153
FIGURA 6.5 – Modelagem gerencial do papel do engenheiro de software	155
FIGURA 6.6 – Ciclos de vida com enfoque em reuso, reengenharia, ações e revisões	156
FIGURA 6.7 – Fluxo de trabalho do projeto de modernização do sistema de software (RATIONAL CORPORATION, 2000).....	160
FIGURA 6.8 – Fluxo de trabalho de soluções de problemas de sistema de software	164
FIGURA 6.9 – Revisor de engenharia de sistema de software	165

FIGURA 6.10 – Fluxo de trabalho de gerenciamento de configuração e controle de mudanças do RUP	167
FIGURA 6.11 – Modelagem de ambiente de desenvolvimento	177
FIGURA 6.12 – Papel do analista de sistema na modelagem de requisitos	181
FIGURA 6.13 – Modelagem do papel e das atividades do especialista em interface	187
FIGURA 6.14 – Atividades e artefatos do arquiteto de instrumentação.....	189
FIGURA 6.15 – Arquitetura de hardware do subproduto SSV e de parte do SBA .	191
FIGURA 6.16 – Arquitetura de hardware para o sistema de aquisição de dados do TA-2	192
FIGURA 6.17 – Arquitetura de hardware para o sistema de controle do TA-2	193
FIGURA 6.18 – Fluxo de trabalho de análise e projeto do RUP	195
FIGURA 6.19 – Papel do arquiteto de software do novo sistema de medidas	198
FIGURA 6.20 – Diagrama de contexto de interfaces do subproduto SLF	201
FIGURA 6.21 – Diagrama de contexto do software de redução de dados do túnel de vento	207
FIGURA 6.22 – DFD0 – Cálculo de redução de dados	208
FIGURA 6.23 – DFD1 – Obter dados de ensaio.....	208
FIGURA 6.24 – DFD2 – Obter dados do objeto ensaiado.....	209
FIGURA 6.25 – DFD3 – Fornecer redução de dados	209
FIGURA 6.26 – Diagrama de contexto do STD	211
FIGURA 6.27 – Arquitetura detalhada do sistema de software.....	215
FIGURA 6.28 – Papel do implementador no TA-2	216
FIGURA 6.29 – Diagrama de estado para os subprodutos SWE em LabView	217
FIGURA 6.30 – Diagrama de estado com botões em LabView	219
FIGURA 6.31 – Declaração de unidade de software de aquisição de <i>scanivalve</i> ...	221
FIGURA 6.32 – Declaração de unidades de software de aquisição de dados e controle do túnel de vento	222
FIGURA 6.33 – Continuação da declaração de unidades do software de aquisição de dados e controle do túnel de vento	223
FIGURA 6.34 – Hierarquia de unidades do subproduto SLF	225

FIGURA 6.35 – Unidade de software para gravar coeficientes de calibração	226
FIGURA 6.36 – Cálculo de coeficientes de tração e compressão.....	226
FIGURA 6.37a – Unidade de cálculo de erro fiducial	227
FIGURA 6.37b – Cálculo de correlação	227
FIGURA 6.38 – Unidade de software de cálculo de ajuste de curva.....	227
FIGURA 6.39 – Unidade de software de leitura dos sinais do multímetro	228
FIGURA 6.40 – Unidade de software de identificação do ensaio	231
FIGURA 6.41 – Diagrama de bloco e funções da unidade identificação de ensaio.....	231
FIGURA 6.42 – Unidade de software para aquisição de dados de <i>scanivalves</i> P1 .	232
FIGURA 6.43 – Resultados de testes de ensaio A0045 e armazenagem de dados..	233
FIGURA 6.44 – Resultados de testes de ensaio A0046 e armazenagem de dados..	233
FIGURA 6.45 – Implementação SBA alto nível balanceado.....	238
FIGURA 6.46 – Implementação SBA baixo nível balanceado.....	238
FIGURA 6.47 – Implementação SBA alto nível desbalanceado	239
FIGURA 6.48 – Implementação SBA baixo nível desbalanceado	239
FIGURA 6.49 – Implementação SBA aquisição, ganho e zero inicial	240
FIGURA 6.50 – Implementação SBA de unidade para gravação de dados.....	240
FIGURA 6.51 – Diagrama das medidas de alto e baixo nível	243
FIGURA 6.52 – Resultado da calibração do <i>pitot</i> com HP.....	248
FIGURA 6.53 – Resultado da calibração do <i>pitot</i> com PC.....	250
FIGURA 6.54 – Correlação entre a calibração do <i>pitot</i> com plataforma HP X PC.	250
FIGURA 6.55 – Painel de acionamento do cálculo de ganho.....	252
FIGURA 6.56 – Painel de cálculo de ganho do SCB.....	252
FIGURA 6.57 – Monitoramento dos dados do subsistema motor e hélice.....	253
FIGURA 6.58 – Painel de tipos de aquisições e calibrações	254
FIGURA 6.59 – Painel de zeragem automática da balança	254
FIGURA 6.60 – Painel de obtenção de dados de ensaios, ZAB e ZQ	255
FIGURA 6.61 – Painel de controle de tipo de ensaio	255
FIGURA 6.62 – Distribuição de R1 para história de tempo de 10 segundos	257
FIGURA 6.63 – Distribuição de R1 para história de tempo de 6 segundos	258

FIGURA 6.64 – Papel do integrador no TA-2	259
FIGURA 6.65 – Reestrutura de dados e código que desconta zero	262
FIGURA 6.66 – Reestruturação de dados modificáveis em campanhas de ensaios	262

LISTA DE TABELAS

TABELA 4.1 – Atividades de engenharia de software executadas no TA-2	70
TABELA 4.2 – Iterações de cada fase do processo unificado	77
TABELA 5.1 – Ambientes operacionais	137
TABELA 5.2 – Ferramentas do ambiente de engenharia de software	138
TABELA 6.1 – Coeficientes obtidos com SLF	229
TABELA 6.2 – <i>Scanivalves</i> usadas no ensaio do modelo NACA0012 do TA-2	234
TABELA 6.3 – Dados de <i>offset</i> de baixo nível com HP e PC	243
TABELA 6.4 – Dados de calibração de baixo nível com HP	244
TABELA 6.5 – Dados de calibração de baixo nível com PC	244
TABELA 6.6 – Dados de <i>offset</i> de alto nível com HP e PC	244
TABELA 6.7 – Dados de calibração de alto nível com HP	244
TABELA 6.8 – Dados de calibração de alto nível com PC	245
TABELA 6.9 – Cálculo do ganho HP e PC	245
TABELA 6.10 – Resultado da aplicação do caso de teste de calibração do <i>pitot</i> com HP	248
TABELA 6.11 – Resultado da aplicação do caso de teste de calibração do <i>pitot</i> com PC	249
TABELA 6.12 – Combinações de grau 1	265
TABELA 6.13 – Cálculo de carga de F1 com software da HP	266
TABELA 6.14 – Valores dos desvios padrão de cargas parcial e total de F1 com HP	267
TABELA 6.15 – Cálculo de cargas de F1 com software do PC	267
TABELA 6.16 – – Valores dos desvios padrão de cargas parcial e total de F1 com PC	268
TABELA 6.17 – Resultados dos testes de obtenção da matriz de calibração.....	270
TABELA 6.18 – Resultados de ensaios com modelo aerodinâmico NACA0012 ...	271

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
AEDC	Arnold Engineering Development Center (Centro de Desenvolvimento de Engenharia de Arnold)
AIAA	American Institutes of Aeronautics and Astronautics
AQAP	Allied Quality Assurance Publications
ANSI	American National Standard Institute
ASA	Divisão de Sistemas Aeronáuticos
ASA-L	Subdivisão de Aerodinâmica
CASE	Computer Aided Software Engineering
CFD	Computational Fluid Dynamics (Dinâmicas de Flúidos Computacional)
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
COTS	Commercial off the Shelf
CTA	Comando-Geral de Tecnologia Aeroespacial (antigo Centro Técnico Aeroespacial)
DAQ	Data Aquisition (Biblioteca de Aquisição de Dados)
DMA	Direct Memory Access
DOD	Department of Defense
DFD	Diagrama de Fluxo de Dados
EMBRAER	Empresa Brasileira de Aeronáutica
ERJ	Jato regional da EMBRAER
ETE	Escola Técnica do Exército
GPIB	General Purpose Interface Bus
GTTC	Ground Test Technical Committee (Comitê Técnico de Teste de Solo)
HP	Hewlett Packard
IAE	Instituto de Aeronáutica e Espaço
IBTWG	Internal Balance Technology Working Group (Grupo de Trabalho de Tecnologia de Balança Interna)

IEC	International Electrotechnical Commission
IEEE	Institute of Electric and Electronic Engineers
INMETRO	Instituto Nacional de Metrologia, Normalização e Qualidade Industrial
IPD	Instituto de Pesquisa e Desenvolvimento
ITA	Instituto Tecnológico de Aeronáutica
ISO	International Organization for Standardization
LA	Laboratório de Aerodinâmica
LabView	Laboratory Virtual Instrument Engineering Workbench
METAPLI	Metodologia Especial Tradicional e Ágil de aplicação em Projeto Legado de software em Inovação
NASA	National Aeronautics and Space Administration
OTAN	Organização do Tratado do Atlântico Norte
PAR	Divisão de Aeronáutica
PAR-L	Subdivisão de Ensaio em Túnel de Vento
PC	Personal Computer
PLA	Divisão de Aerodinâmica
PnP	Plug and Play
PXI	PCI eXtensions for Instruments
RUP	Rational Unified Process
SAP	Software Aplicativo
SAV	Software com aplicativo ARENA para Cálculo de Distribuição de Dados
SATA	Subsonic Aerodynamic Testing Association (Associação Internacional de Testes em Túneis de Vento Subsônicos)
SBA	Software de Aquisição de Dados e Calibração da Balança
SCB	Software de Aquisição de Dados e Controle do Túnel de Vento
SCXI	Signal Conditioning eXtensions for Instrumentation
SEI	Software Engineering Institute
SEL	Software Engineering Laboratory (Laboratório de Engenharia de Software)

SIG	Software Aplicativo de Interface GPIB
SIVAM	Sistema Integrado de Vigilância da Amazônia
SIW	Software Aplicativo de Interface DAQ
SLF	Software de Aquisição de Dados do Laboratório de Força
SOP	Software Operacional
SRD	Software de Redução de Dados do Túnel de Vento
SSV	Software de Aquisição de Dados de <i>scanivalve</i>
STD	Software de Transferência de Dados da HP para o PC
SVE	Software com aplicativo <i>EXCEL</i> para ordenação de dados da <i>scanivalve</i>
SWA	Software de Apoio ao Desenvolvimento
SWE	Software de Ensaios em Túnel
SWI	Software de Interface
SWR	Software de Validação da Redução de Dados
SWV	Software de Validação de Dados Experimentais
TA-1	Túnel Aerodinâmico nº 1
TA-2	Túnel Aerodinâmico nº 2
TTP	Túnel Transônico Piloto
UML	Unified Modeling Language
USA	United States of American (Estados Unidos da América)
VI	Virtual Instruments
VLS	Veículo Lançador de Satélite
XP	eXtreme Programming
ZAB	Correção da variação da atitude do modelo
ZQ	Correção do efeito da pressão dinâmica

SUMÁRIO

1 INTRODUÇÃO	1
1.1 PROJETOS REALIZADOS E MODERNIZAÇÕES DO TA-2	2
1.2 MOTIVAÇÃO DO TRABALHO	2
1.3 OBJETIVOS DO TRABALHO	4
1.4 PERSPECTIVAS DO TRABALHO	5
 2 TÚNEL DE VENTO NA PRÁTICA	 6
2.1 ENSAIOS EM TÚNEL DE VENTO	7
2.2 AMBIENTES DE ENSAIOS EM TÚNEL DE VENTO DO CTA	12
 3 ANÁLISE DE METODOLOGIAS	 19
3.1 METODOLOGIAS E MODELOS DE PROCESSO	19
3.2 AÇÕES DAS METODOLOGIAS TRADICIONAIS.....	28
3.2.1 Modelo clássico ou modelo em cascata	28
3.2.2 Modelo de processo incremental	31
3.2.3 Modelo de processo prototipação	33
3.2.4 Modelo de processo em espiral	35
3.3 AÇÕES DAS METODOLOGIAS ÁGEIS	38
3.3.1 Metodologia eXtreme Programming (XP)	39
3.3.2 Metodologia scrum	40
3.3.3 Metodologia de processo unificado	40
3.4 AMBIENTES DE PROGRAMAÇÃO VISUAL COMERCIAIS	47
3.5 REUSO NO CICLO DE VIDA DE SOFTWARE	50
3.6 REENGENHARIA NA REDUÇÃO DE DADOS	54
 4 METODOLOGIA DE PROJETO DE SOFTWARE.....	 62
4.1 A METAPLI	62
4.2 METODOLOGIA E PROCESSOS APLICADO NO TA-2.....	66
4.3 PLANEJAMENTO DOS NOVOS PROCESSOS VIA METAPLI.....	69
4.3.1 Ações do processo ágil	74
4.3.1.1 Ações para a fase de iniciação.....	78
4.3.1.2 Ações para a fase de elaboração	78
4.3.1.3 Ações para a fase de construção	81
4.3.1.4 Ações para a fase de transição.....	81
4.3.2 Ações do processo tradicional	82
4.4 PLANEJAMENTO DAS AÇÕES GERENCIAIS.....	83
4.5 PLANEJAMENTO DAS AÇÕES TÉCNICAS.....	89
4.5.1 Entregas e aceitações frequentes	93
 5 PROJETO DO NOVO SISTEMA DE MEDIDAS.....	 96
5.1 IDENTIFICAÇÃO DOS SUBSISTEMAS DO TA-2.....	101
5.1.1 Subsistema motor e hélice	106

5.1.2 Subsistema balança	109
5.1.3 Subsistema mastro	110
5.1.3.1 Arquitetura do subsistema mastro	113
5.1.4 Subsistema de sensoramento	115
5.1.4.1 Arquitetura do subsistema sensoramento	115
5.1.5 Subsistema de montagem do modelo aerodinâmico	116
5.1.6 Subsistema de visualização	117
5.1.7 Subsistema de aquisição de dados	119
5.1.8 Subsistema de redução de dados	121
5.2 PROCESSO DE APLICAÇÃO DE REUSO	125
5.3 PROCESSO DE APLICAÇÃO DE REENGENHARIA DE SOFTWARE	127
5.4 REQUISITOS DO SISTEMA DE SOFTWARE DO TA-2	130
5.4.1 Requisitos gerais do sistema de medidas	131
5.4.2 Requisitos de interface homem-máquina	132
5.4.3 Requisitos de interface de comunicação computacional	134
5.5 AMBIENTES DE ENGENHARIA PARA MEDIDAS E CONTROLE DO TA-2	137
5.6 PROCESSO DE ANÁLISE DE DESEMPENHO DOS DADOS	141
5.7 REALINHAMENTO DE DADOS E PROCESSOS	142
5.7.1 Gerência de requisitos variáveis	144
 6 IMPLANTAÇÃO E RESULTADOS DA METAPLI	145
6.1 DOCUMENTAÇÃO GERENCIAL VIA METAPLI	146
6.2 RESULTADO DA MODELAGEM GERENCIAL	150
6.3 DOCUMENTAÇÃO TÉCNICA VIA METAPLI	171
6.4 RESULTADOS DA MODELAGEM DO SISTEMA DE SOFTWARE	172
6.4.1 Modelagem do ambiente de desenvolvimento	175
6.4.2 Modelagem de requisitos, análise e projeto	179
6.4.2.1 Software de aquisição de dados do laboratório de força - SLF	199
6.4.2.2 Software de redução de dados do túnel de vento - SRD	202
6.4.3 Modelagem do projeto detalhado, implementação e testes	213
6.4.3.1 Implementação e testes do subproduto de software de aquisição de dados do laboratório de força – SLF (C1)	224
6.4.3.2 Implementação e testes do subproduto de software de aquisição de dados de <i>scanivalve</i> – SSV (C2)	230
6.4.3.3 Implementação e testes do subproduto de software com aplicativo <i>EXCEL</i> para ordenação de dados de <i>scanivalve</i> – SVE	236
6.4.3.4 Implementação e testes do subproduto de software de aquisição de dados e calibração da balança – SBA (C3)	237
6.4.3.4.1 Caso de teste de cálculo de ganho	242
6.4.3.5 Implementação e testes do subproduto de software de transferência de dados da HP para o PC – STD	246
6.4.3.5.1 Caso de teste de calibração de <i>pitot</i>	247
6.4.3.6 Implementação e testes do subproduto de software de aquisição de dados e controle do túnel de vento – SCB (C4)	251

6.4.3.7 Implementação e testes do software com aplicativo ARENA para cálculo de distribuição de dados – SAV	256
6.4.3.8 Implementação e testes do subproduto de software de redução de dados do túnel de vento – SRD	260
6.4.3.8.1 Caso de teste para obtenção de matriz de calibração	264
 7 DISCUSSÕES DOS RESULTADOS.....	273
8 CONCLUSÕES E TRABALHOS FUTUROS.....	277
 REFERÊNCIAS	280
APÊNDICE A – Principais Normas de Software	291
APÊNDICE B – Materiais do sistema de Medidas do TA-2	294
APÊNDICE C – Glossário de termos	312
APÊNDICE D – Códigos dos Subprodutos de Software	323

CAPÍTULO 1 INTRODUÇÃO

Entre 1950 e 1958 foi construído no Centro Técnico Aeroespacial – CTA, hoje denominado Comando-Geral de Tecnologia Aeroespacial (CTA), o Túnel Aerodinâmico nº 2 – TA-2, no qual eram executados apenas ensaios aeroelásticos, com balanças internas e visualização.

Em 1953, com a criação do Instituto de Pesquisa e Desenvolvimento – IPD, foi possível o desenvolvimento e ensaios no TA-2 de diversos projetos aeronáuticos, com recursos de visualização de escoamento, e medições analógicas e não automatizadas, comportando em uma Seção de Ensaios o modelo aerodinâmico em escala. Dentre estes projetos estava o da aeronave “Bandeirantes”, realizado totalmente por integrantes deste Instituto e que deu origem a empresa EMBRAER.

Somente em 1978 teve início a primeira modernização do TA-2, com uso efetivo de uma balança externa, e com sensores eletrônicos em substituição aos sensores analógicos, para medição das cargas aerodinâmicas. Assim, foi desenvolvida uma metodologia para a calibração da balança externa ou principal. A então Subdivisão Aerodinâmica - ASA-L é a única entidade da América Latina que faz parte da Associação Internacional de Túneis de Vento Subsônicos (SATA), sendo associada desde 1977.

Em 1979, com os equipamentos da plataforma Hewlett Packard – HP foi implantado o primeiro Sistema Automático de Aquisição de Dados. Esta modernização foi executada pelos engenheiros e técnicos que trabalhavam naquela época, envolvendo as atividades de operacionalização da balança externa e de implantação do sistema computacional de aquisição de dados e desenvolvimento dos softwares em linguagem BASIC. A partir desta evolução do sistema de medidas com dispositivos eletrônicos e softwares para aquisição e redução de dados, iniciou-se o período de utilização plena do TA-2.

1.1 PROJETOS REALIZADOS E MODERNIZAÇÕES DO TA-2

A partir de sua primeira modernização com equipamentos HP, o TA-2 tornou-se o principal túnel de vento brasileiro, no qual vêm sendo executados diversos ensaios de projetos aerodinâmicos. Neste túnel foram ensaiados, entre outros, os seguintes modelos aerodinâmicos: aeronaves da EMBRAER, tais como EMB-121 Xingu, EMB-120 Brasília, EMB-312 Tucano, CBA-123 Vector, AMX, EMB-145 e EMB-170; foguete VLS; mísseis das empresas AVIBRAS e de antigas empresas como a Órbita e ENGESA; artefatos de defesa, tais como bombas e projéteis; veículos terrestres, como ônibus da ITAPEMIRIM; estruturas, tais como pontes e antenas parabólicas; plataformas da Petrobrás; navios e submarinos da Marinha do Brasil.

Em 1996, com seus equipamentos envelhecidos, sem condições de operar com segurança e sendo constatada a inviabilidade de sua recuperação, foi proposto o início do segundo projeto de modernização do TA-2, com emprego de investimentos visando à obtenção de retornos econômicos com os serviços prestados de ensaios aerodinâmicos em projetos de diversas áreas de aplicação.

Para este segundo projeto de modernização do TA-2 foi proposto este trabalho, que tem por objetivo a atualização dos softwares de aquisição e redução de dados, através de recursos tecnológicos, interfaces e aplicativos de apoio avançados, com enfoques no emprego das atividades de engenharia e qualidade de software. Também abrange a atualização do sistema de medidas e seus subsistemas originais, com a inovação e adequação dos equipamentos operacionais, em substituição aos da HP.

1.2 MOTIVAÇÃO DO TRABALHO

O sistema de software do TA-2 em uso no ano de 1998, quando se iniciou este trabalho, havia sido desenvolvido havia muitos anos, e operava importantes funções do sistema de medidas do túnel de vento nº 2 com a plataforma da Hewlett Packard -HP, porém seu desempenho estava precário e a manutenção dos softwares totalmente

dependente do talento de especialistas que ali trabalhavam. Os especialistas em ensaios em túnel necessitavam de mecanismos computacionais e gerenciais imediatos para recuperação da usabilidade do sistema de medidas e dos softwares que influenciam diretamente nos ensaios aerodinâmicos.

A motivação deste trabalho é recuperar a usabilidade dos processos de ensaios em túnel, mantendo o TA-2 operacional durante a modernização. Assim, se propõe usar diversos recursos computacionais para fazer a atualização e revitalização do sistema de software, empregando práticas de engenharia de software que possam dotar o sistema de medidas de tecnologias viáveis, que através de ações de metodologias resultem na construção de softwares funcionais em conjunto com sistemas de computadores e dispositivos eletrônicos adequados. Durante esta modernização, o CTA poderá executar os ensaios aerodinâmicos até a transição final para o novo sistema de medidas, prosseguindo com a prestação dos serviços de ensaios em túnel de vento, que são essenciais para os projetos aerodinâmicos do país.

Para possibilitar esse trabalho fez-se uso de engenharia de software, por intermédio do estudo de novas práticas, métodos e processos que fossem adequados durante o ciclo de desenvolvimento, considerando o conceito atual de software que abrange os programas de computadores, a documentação associada e os dados necessários para fazer com que os sistemas operem corretamente e continuamente até poder ser substituídos por completo. O enfoque principal dado por este trabalho é buscar recursos tecnológicos para o projeto e desenvolvimento de software, visando aumentar a produtividade, que neste projeto significa prover a continuidade dos serviços prestados pelo TA-2 durante a modernização, com custos reduzidos na implantação dos processos de engenharia e de software nos ensaios aerodinâmicos.

Para atingir esta abrangência do conceito de software criou-se uma metodologia para aplicação de reuso e reengenharia no desenvolvimento dos softwares de aquisição e redução de dados de um túnel de vento, com as ações das metodologias tradicionais e ágeis adequadas aos processos e ao grupo de projeto. Esta metodologia de projeto foi

gerada pelo autor deste trabalho de tese e denominada Metodologia Especial Tradicional e Ágil de aplicação em Projeto Legado de software em Inovação - METAPLI. A metodologia de projeto tem a finalidade de auxiliar os envolvidos no desenvolvimento de software a compreender melhor as atividades desempenhadas, para com isso reduzir o esforço de modernizar o sistema de medidas do TA-2, e produzir os softwares de aquisição e de redução de dados, de modo a garantir através da documentação do projeto a preservação do conhecimento adquirido.

1.3 OBJETIVOS DO TRABALHO

Os objetivos deste trabalho compreendem:

1. Estabelecer uma metodologia simples e representativa para projetos de desenvolvimento de sistema de software.
2. Mostrar como foi definida e implantada esta metodologia (METAPLI), sua abrangência, consistência, utilidade e representatividade para o TA-2.
3. Disponibilizar informações e recursos de forma organizada, que permitam verificar e avaliar a eficácia dos processos aplicados na atualização e revitalização do sistema de medidas de um túnel de vento.

Para atingir os objetivos, a contribuição deste trabalho é composta por:

1. Análise de diversas fontes de informações, as quais se baseiam em metodologias tradicionais e ágeis, processos de desenvolvimento de software, sistema de medidas, ensaios em túnel de vento, reuso e reengenharia de software.

2. Estabelecimento de processos, métodos, práticas, técnicas e capacitações adequadas ao uso e aplicável ao empreendimento.
3. Documentação gerada pela inovadora abordagem gerencial e técnica em todo o ciclo de vida de desenvolvimento de software.
4. Obtenção dos subprodutos de software operacionais durante o desenvolvimento, com dados e resultados representativos dos ensaios em túnel de vento, através da implantação da METAPLI.

1.4 PERSPECTIVAS DO TRABALHO

Este trabalho foi desenvolvido entre 1998 e 2007, com a meta de alcançar um aumento no desempenho do sistema de medidas de um túnel de vento, através de automação assistida por computadores pessoais, e auxiliada por combinações de hardwares e dispositivos eletrônicos mais modernos, que culminou na necessidade de desenvolvimento de software adequado ao contexto do projeto de modernização proposto.

Este trabalho está organizado da seguinte forma: o capítulo atual apresenta a introdução sobre a abrangência, objetivos, perspectivas e contribuições deste trabalho. O capítulo 2 apresenta o túnel de vento na prática e os assuntos pertinentes a este trabalho. O capítulo 3 faz uma abordagem das diversas metodologias. No capítulo 4 é apresentada a metodologia de projeto de software – METAPLI deste trabalho de pesquisa. No capítulo 5 é descrito o projeto do novo sistema de medidas. O capítulo 6 descreve detalhes da implantação e resultados da METAPLI. O capítulo 7 aborda as discussões dos resultados do trabalho. No capítulo 8 são descritos as conclusões e os trabalhos futuros propostos.

CAPÍTULO 2 TÚNEL DE VENTO NA PRÁTICA

Os softwares de aquisição e redução de dados do Túnel Aerodinâmico nº 2 (TA-2) do Comando-Geral de Tecnologia Aeroespacial (CTA) têm sido suportados e mantidos pelos especialistas em ensaios em túnel de vento do CTA desde 1979. Os softwares originais foram desenvolvidos para a Plataforma Hewlett Packard (HP) e vêm apresentando problemas de manutenção.

Esses softwares vêm sendo modificados e têm surgido erros ou “bugs” que comprometem os dados, devido às diversas adaptações para o uso nesta plataforma de hardware. Muitas destas alterações de software têm sido feitas sem a adoção de uma metodologia adequada para responder aos processos de engenharia e gerenciais. Para atender os requisitos da comunidade de usuários, com respeito aos dados pertinentes aos ensaios aerodinâmicos, os softwares, os hardwares, as instrumentações e os dispositivos eletrônicos envolvidos, devem ser avançados e robustos, de modo a apresentar os resultados de ensaios com maior rapidez, confiabilidade, precisão e prover usabilidade.

Neste trabalho é detalhada a modernização do TA-2, através da qual foi implantada a metodologia para aplicação de reuso e reengenharia nos softwares de aquisição e redução de dados de um túnel de vento, denominada Metodologia Especial Tradicional e Ágil de aplicação em Projeto Legado de software em Inovação - METAPLI, que permite prover a abordagem dos processos de engenharia e gerenciais pertinentes, através da pesquisa das ações de outras metodologias e processos de desenvolvimento, existentes na literatura.

Com relação à utilização da metodologia METAPLI, é importante salientar que para o sucesso do trabalho, muitos esforços foram direcionados para documentar o desenvolvimento dos softwares de aquisição e redução de dados, e identificar as capacidades operacionais de um sistema de medidas de túnel de vento. Outros esforços

foram direcionados para entender cada atividade envolvida no ensaio, para obter o domínio de conhecimento de um ensaio aerodinâmico e, através das tecnologias disponíveis, gerar e implementar as funcionalidades nos subprodutos de software, visando garantir a qualidade dos dados intermediários e finais.

2.1 ENSAIOS EM TÚNEL DE VENTO

O American Institutes of Aeronautics and Astronautics (AIAA) criou um grupo composto de renomados especialistas em túnel de vento, oriundos de organismos governamentais e de indústrias da área aeroespacial, como a *National Aeronautics and Space Administration – NASA, Ames Research Center, Lockheed Martin Aeronautics, Boeing Company, Raytheon Missiles*, para descrever as melhores práticas aplicáveis nos ensaios em túnel de vento. Este grupo, denominado *Ground Test Technical Comitee* (GTTC), afirma que as melhores práticas associadas com ensaios de modelos aerodinâmicos em túnel de vento buscam alcançar eficiência, menores custos e minimizar o tempo de ciclo no desenvolvimento de projetos aeroespaciais (AIAA R-092-1-2002, 2002).

Esses centros de referência em pesquisas aeroespaciais consideram os ensaios em túneis de vento componentes importantes no desenvolvimento de veículos aerodinâmicos e suas tecnologias associadas. Afirmam que um ensaio é uma ferramenta essencial para melhorar o desempenho das aeronaves e garantir a segurança das mesmas, de modo a reduzir os riscos que podem ter influência no ciclo de vida dos veículos aéreos.

O GTTC afirma que para o desenvolvimento de veículos aéreos militares podem ser necessários ensaios adicionais, devido às configurações necessárias que tratam, por exemplo, de parte da configuração de teste, carregamento de armamento, e efeitos da propulsão, altamente integrados. Portanto, para os projetos dos veículos militares, os

ensaios em túnel de vento são considerados mais complexos, e requerem instrumentações adicionais para obtenção de dados com qualidade.

Segundo o GTTC, os requisitos de instrumentação variam amplamente, para obtenção dos dados de cada ensaio em túnel de vento, pois são utilizados vários tipos de sensores como *strain gages* para balanças internas, células de carga em balanças externas, transdutores de pressão, inclinômetros, termopares, entre outros. A instalação das instrumentações necessárias e de seus elementos associados como cablagem, condicionadores de sinais, filtros, calibração, redução de dados e apresentação dos dados, podem onerar o orçamento de um ensaio em túnel de vento.

No artigo do GTCC (AIAA R-092-1-2002, 2002), o ensaio em túnel de vento é um componente crítico, de alto custo e de cronograma extenso. Assim sendo, deve existir uma programação de ensaio que focalize a redução de riscos do projeto aerodinâmico, antes de construir e evoluir os veículos aéreos. O programa de ensaio em túnel de vento engloba os investimentos e o tempo gasto para o projeto do modelo aerodinâmico, a fabricação, a preparação do ensaio, o ensaio do modelo e a análise do ensaio, com registro das lições aprendidas.

Há uma grande dificuldade para elaboração e cumprimento de um programa de ensaio, também descrito pelo GTCC em outro artigo (AIAA-R-092-2-2002, 2002), exigindo esforço para que o desempenho do trabalho de ensaio seja feito de maneira consistente, e proporcione a qualidade necessária para garantir o seu sucesso. As dificuldades aumentam ainda mais quando o programa de ensaio for executado em túneis de vento do exterior, pois os serviços de ensaio contratado para aquele projeto específico, devem ter uma relação entre custo, qualidade, quantidade de dados e resposta ao cronograma, estabelecidos corretamente em requisitos do contrato. No caso de um ensaio no exterior, muito do que é programado é caro, e os processos empregados durante o programa de ensaio não estão disponíveis e acessíveis para auxiliar na comprovação da qualidade do projeto aerodinâmico, e caso haja problemas

com os resultados do projeto, ensaios adicionais serão acordados em aditivos do contrato, não previstos no orçamento inicial.

De acordo com o GTCC (AIAA R-092-1-2002, 2002) os dados requeridos para cada ensaio em túnel variam bastante, e geralmente é necessário empregar diversos instrumentos de medição. Assim, diferentes tipos de ensaio requerem instrumentação adicional para correlacionar ou obter dados corretos se comparados com os métodos analíticos, e requerem também instrumentação específica para comprovação de precisão.

Com a evolução eletrônica e computacional, as instrumentações de ensaio em túnel requerem softwares adequadamente implantados, para obtenção de dados dos vários tipos de ensaios que os clientes necessitam para validar seus projetos aerodinâmicos, e de resultados corretos e confiáveis dos ensaios programados. Os diferentes ensaios requerem também detalhes de análise de incerteza (AIAA-S-071-1995, 1995, CAHILL, 1996) para garantir a instrumentação apropriada e a precisão dos dados, visando concluir com êxito as metas de um programa de ensaio.

No artigo do GTCC (AIAA R-092-1-2002, 2002) são encontradas diversas considerações sobre como selecionar um túnel de vento para um projeto específico. É mencionado que, ao executar uma seleção de túnel de vento, os seguintes pontos devem ser observados:

- A dimensão da Seção de Ensaio: para possibilitar estabelecer o tamanho do modelo de ensaio e não comprometer as correções dos dados.
- As condições de operação do túnel: inclui os números de Mach e de Reynolds e a variedade de condições de ensaio do túnel.
- A qualidade do fluxo de túnel: inclui a análise de incerteza dos parâmetros de ensaio e as medições envolvidas.

- Os custos do túnel: incluem custos do ambiente de pré-ensaio, ensaio e pós-ensaio e uma avaliação em relação à produtividade do túnel, relacionadas ao modelo aerodinâmico de ensaio.
- Viabilidade do túnel: inclui as facilidades oferecidas em relação ao modelo de ensaio, ou seja, se este túnel pode ou não realizar o ensaio neste modelo aerodinâmico no tempo requerido.
- A capacidade de aquisição de dados: inclui uma avaliação de quais são as capacidades de aquisição de dados oferecidas, e quais delas necessitam do suporte do cliente no local de ensaio. Casos mais comuns destas necessidades ocorrem na montagem e troca de configuração do modelo aerodinâmico.
- A análise e redução de dados: inclui a capacidade de analisar os dados durante os ensaios e de reduzir esses dados com todas as correções aplicadas, em relação à qualidade do fluxo da seção de ensaio, às interferências do suporte do modelo aerodinâmico, à interferência das paredes, à geometria do modelo de ensaio, e a outras correções pertinentes.
- A instrumentação: inclui a viabilidade de implantar os meios necessários para auxílio ao cumprimento dos requisitos de ensaio do modelo aerodinâmico.
- A experiência: inclui a história de sucesso de executar o tipo de ensaio requerido e o registro de desempenho de ensaios passados e de lições aprendidas.
- O pessoal de apoio: inclui o conhecimento dos especialistas para avaliar e interpretar os resultados dos ensaios e executar os ensaios do tipo daquele requerido.
- O hardware de apoio: inclui o hardware existente e a viabilidade de uso de outros, caso necessário.

- A segurança: inclui os requisitos de segurança das informações fornecidas pelo cliente sobre o modelo aerodinâmico a ser ensaiado e a quem podem ser divulgadas.
- A qualidade da informação e dos dados: inclui a capacidade para produzir os dados, ou seja, a combinação de túnel, modelo, e instrumentação provida, para obter a qualidade dos dados, e o sistema de gerenciamento de informação seguido.
- Os elementos contratuais: incluem as relações formais estabelecidas e o desempenho do ensaio e riscos de execução do contrato, e também as documentações requeridas.

Além deste artigo do GTCC (AIAA R-092-1-2002, 2002), muitos outros trabalhos foram desenvolvidos com respeito a ensaios em túnel de vento. Dentre eles, podemos descrever sobre o trabalho publicado também pelo AIAA (SHIVANANDA et al., 1997), no qual foi enfatizada a importância dos ensaios em túnel de vento na obtenção de dados de forças e momentos, para reduzir as incertezas de um modelo aerodinâmico na configuração de um veículo lançador.

Neste artigo o processo de ensaio de túnel de vento é descrito, tendo início com a escolha do túnel para ensaios, seguido de definição da escala e fabricação do modelo do lançador em suas configurações, a montagem deste modelo no túnel e o uso de balança interna para medir as componentes de forças e momentos.

Os autores Shivananda et al. (1997) mencionam o uso de código de engenharia e métodos de *Computacional Fluid Dynamics* (CFD), para exercitar as características aerodinâmicas do veículo lançador, e comparam os resultados obtidos usando os métodos CFD com os adquiridos em túnel de vento, para validação das previsões aerodinâmicas de configuração do veículo lançador.

Contudo, os resultados obtidos com a aplicação de métodos mais modernos como o CFD, ainda são comparados com os dados obtidos em túnel de vento para validação do projeto aerodinâmico. Algumas vezes, nem todos os métodos novos são validados em todas as suas execuções, podendo se afirmar que os resultados obtidos dos ensaios em túnel de vento ainda são essenciais e confiáveis para determinação das características aerodinâmicas de veículos aéreos.

Sendo assim, os dados adquiridos e processados com relação ao modelo aerodinâmico ensaiado em túnel de vento são de suma importância para o sucesso de um produto aeroespacial e, de alguma forma, devem ser armazenadas as lições aprendidas para uso futuro. Há necessidade da descrição da maneira empregada para obtenção dos dados para comprovar a confiabilidade e possibilitar a rastreabilidade destes, e quais foram as instrumentações, os dispositivos eletrônicos, os softwares e os hardwares, e baseados em que plataformas foram obtidos.

Neste trabalho então, foram envidados esforços para conhecer e documentar os ambientes de engenharia e de software do túnel de vento do CTA com uso da plataforma HP, para identificar as melhorias necessárias no processo de avaliação do projeto de aeronaves. Este conhecimento facilita a inclusão de novas tecnologias no projeto de modernização do sistema de medidas.

2.2 AMBIENTES DE ENSAIOS EM TÚNEL DE VENTO DO CTA

Os diversos tipos de veículos aéreos construídos no Brasil foram ensaiados em túnel de vento do CTA, além de muitos outros projetos aerodinâmicos. Muitas dessas construções e evoluções de projeto, tais como: aeronaves, pontes, navios, plataformas de petróleo, foguetes de sondagem, lançadores de satélites, entre outros produtos, representam um avanço tecnológico para o país, e todos esses elementos sofrem

influências do vento, que precisam ser estudadas para que seus projetos obtenham o sucesso esperado.

O CTA executa os serviços de ensaios em túnel para seus clientes em projetos dessas diversas áreas no TA-2, desde 1960. Desde aquela data, os especialistas vêm praticando diversos tipos de ensaios experimentais, principalmente para avaliação de projetos aeroespaciais de grande importância para o Brasil. Pela sua capacidade operacional, o TA-2 é considerado o mais completo túnel de vento da América Latina.

Os ensaios desses diversos projetos brasileiros podem ser vistos no *site* do Laboratório de Túnel de Vento do Instituto de Aeronáutica e Espaço - IAE (IAE, 2005). Desde sua construção, o TA-2 vem sendo modernizado para abranger as tecnologias requeridas por seus clientes, visando atender os serviços solicitados nos programas de ensaios, sempre almejando garantir a qualidade dos ensaios, dados e resultados.

A utilização da tecnologia nacional, como um túnel de vento do próprio país, de certa forma, permite a empresas nacionais reduzir o tempo de desenvolvimento de seus projetos e da produção em série, no caso de aeronaves. Assim sendo, pode-se citar a utilização do TA-2 em projetos da EMBRAER. Esta empresa utilizou cerca de 4.000 horas de ensaio em túnel de vento do CTA, com seus especialistas participando ativamente nas diversas campanhas ou programas de ensaios da aeronave ERJ-145 de 50 passageiros, em suas diversas configurações. Dentre estas configurações estão as aeronaves do Sistema Integrado de Vigilância da Amazônia (SIVAM) e a aeronave LEGACY.

As instalações do túnel de vento brasileiro TA-2, mostradas na Figura 2.1, foram construídas na década de 1950 e desenvolvidas com apoio técnico da empresa americana Sverdrup. Projetado para operar em regime subsônico pressurizado, ou seja, em baixas velocidades, na faixa de 500 quilômetros por hora, é mais utilizado pela indústria aeroespacial para simular os efeitos do ar sobre a aeronave quando ela está

em movimento, com objetivo de otimizar o projeto de uma aeronave e minimizar situações de risco.



Figura 2.1 - Instalações do túnel de vento – TA-2 do CTA (IAE, 2005).

Os resultados obtidos dos ensaios destas campanhas conjuntas permitiram o aumento de conhecimento de ambas as partes, ou seja, dos especialistas do CTA e da EMBRAER. Posteriormente, através de uma análise criteriosa das lições aprendidas, foi possível a redução do ciclo de produção dessas aeronaves, sendo que a empresa em 1996 gastou cerca de 14 meses para desenvolvimento e montagem dessas aeronaves. Porém, já no ciclo de produção em 2004, conseguiu reduzir a montagem deste mesmo jato para 3,1 meses, com aumento da receita por empregado.

A indústria aeronáutica não é a única usuária do túnel de vento do CTA: indústrias de defesa aérea, construção civil, eletroeletrônica, automobilística, naval, perfuração de petróleo entre outras, também se valeram do túnel do CTA para qualificarem seus projetos, enfatizando nestes casos também a preservação dos segredos nacionais. Exemplos desses projetos, de grande interesse nacional, são mostrados nas figuras de 2.2 a 2.5 (IAE, 2005). Com a modernização do sistema de medidas do TA-2, se almeja continuar provendo aos clientes os serviços requisitados, porém, procurando-se diminuir o tempo dos ensaios e aumentar a confiança nos dados.

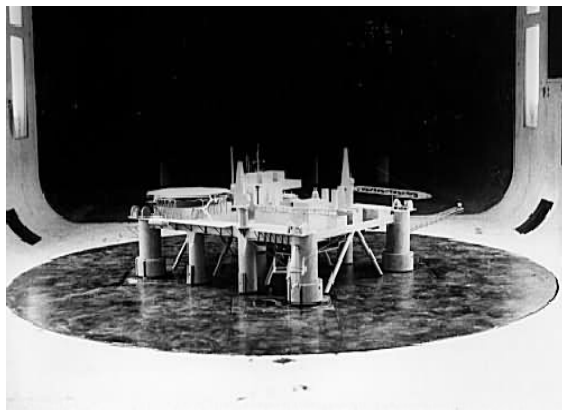


Figura 2.2 - Ponte do rio pinheiros e plataforma de petróleo da PETROBRAS (IAE, 2005).



Figura 2.3 - Ônibus da Itapemirim e aeronave EMB-145 (IAE, 2005).



Figura 2.4 - Veículo lançador de satélite e antena parabólica (IAE, 2005).



Figura 2.5 - Aeronave AMX e aeronave CBA-123 (IAE, 2005).

Segundo o levantamento dos especialistas em ensaios do TA-2, o tempo médio gasto nos ensaios feitos no túnel de vento com equipamentos, instrumentos e softwares interligados na plataforma HP, antes do projeto de modernização, era de 1.000 horas. Com a implantação da METAPLI, foco deste trabalho, pretende-se a redução ou ao menos a permanência desse tempo médio, podendo ser até melhorado. Porém, a sua contribuição maior está em prover a melhoria da qualidade dos serviços prestados, a confiança nos dados e resultados obtidos e o aumento do conhecimento dos envolvidos em relação ao novo sistema de medidas, conhecimentos estes que neste trabalho foram considerados como um aumento de produtividade.

Para se ter uma idéia do uso do TA-2, a EMBRAER, por ser a maior usuária, chegou a fazer em média de 4 a 5 campanhas de ensaios em cada aeronave, ou seja, de 4000 a 5000 horas de túnel, durante a fase de desenvolvimento. Já houve casos, como do projeto AMX, em que esta empresa chegou a ocupar 100% do tempo do túnel do CTA durante o desenvolvimento do referido projeto.

Com o projeto de modernização do TA-2, torna-se necessário desenvolver um novo sistema de medidas que permita a coleta de dados e o seu processamento de modo a caracterizar os efeitos gerados pela influência do ambiente onde o espécime operará dinamicamente, com pouca dependência do operador. Essa evolução de

tecnologia está vinculada à necessidade de concluir os projetos em tempos mínimos, porém com qualidade e confiabilidade.

Assim sendo, houve um esforço para selecionar e implantar práticas eficientes para a reestruturação do sistema de medidas do TA-2, com objetivo de atender ao requisito de redução do tempo de ensaio e de manter o túnel operacional durante o ciclo de vida do projeto descrito neste trabalho. Este projeto abrange a substituição dos equipamentos de medidas e dos softwares de aquisição e redução de dados de forma gradual. Neste contexto, foi criada a METAPLI para a condução do projeto de forma organizada, visando ampliar o conhecimento com relação ao sistema de medidas e à engenharia de software, para possibilitar modernizações futuras.

Para esta modernização focalizou-se primeiramente o trabalho na busca de soluções para minimizar os problemas daqueles equipamentos e softwares que vinham apresentando falhas mais freqüentes, e que comprometiam a qualidade dos dados, devido à vida útil esgotada e à engenharia de software pouco eficiente. Para tanto, foram estudados os sistemas de medidas disponíveis e julgados viáveis tecnológica e financeiramente para o empreendimento, de modo a possibilitar a seleção de dispositivos eletrônicos para adição na nova plataforma PC. As práticas de engenharia, de software e gerenciais pertinentes ao desenvolvimento dos produtos de software foram sendo criadas e implantadas via METAPLI.

Desenvolveu-se um estudo de sistemas de medidas aplicáveis nas áreas aeroespaciais (GASPERS, MUHLSTEIN, 1998) pela NASA, *Boeing Company*, *Lokheed Martin Aeronautics*, e de sistemas de tipo análogo no Brasil (LEMES, 1997), que pudessem ser aplicados e coubessem no orçamento previsto para a modernização, com destaque especial para o conhecimento dos métodos e práticas exercidos pelos especialistas em ensaios no TA-2, que permitissem a execução dos ensaios através de um processo controlado e documentado.

As tecnologias de túneis de vento em escala industrial estão disponíveis em poucos países do mundo como Estados Unidos da América (USA), França, China, Alemanha, Rússia e Holanda. Embora já existindo sistemas de medidas com aplicações em túneis de vento no exterior com qualidade, foi preciso dotar o sistema de medidas do TA-2 de um desempenho operacional integrado, contendo, pelo menos, as seguintes funcionalidades:

- Medidas de forças e momentos
- Medidas de temperatura
- Medidas de vazão
- Sistema de aquisição de dados
- Esquemas para visualização
- Medidas de pressão
- Medidas de velocidade do escoamento
- Sistema de posicionamento
- Sistema de redução de dados

Um sistema de medidas neste porte possui como característica fundamental a integração de outros sistemas ou subsistemas. A ocorrência de falha de qualquer um dos componentes integrados compromete todo este sistema principal, devido a operação conjunta, podendo-se obter dados equivocados do modelo sob ensaio, oferecendo riscos ao projeto de desenvolvimento dos produtos ensaiados no túnel. Analisando este panorama, surgiu então a necessidade de compor as funcionalidades do sistema de software e integrá-lo, de modo a contribuir com este trabalho de pesquisa com a METAPLI, visando cumprir o requisito de aumento da produtividade dos ensaios em túnel de vento, e abranger a qualidade dos dados, dos softwares, dos processos e da documentação no TA-2 do CTA.

Para evoluir o sistema de medidas, estabelecer a instrumentação e demais elementos envolvidos, e também os softwares para obtenção dos dados teóricos, com qualidade e confiáveis, parte da pesquisa foi voltada para a implantação de práticas, que, através da METAPLI, permitissem minimizar falhas nos sistemas de aquisição e de redução dos dados. Assim sendo, a metodologia foi implantada de modo a acomodar mudanças durante o ciclo de vida do software, procurando dirimir as limitações e os problemas da evolução experimental dos ensaios em túnel de vento.

CAPÍTULO 3 ANÁLISE DE METODOLOGIAS

Para desenvolver o projeto do sistema de software deste trabalho, utilizou-se inicialmente a técnica de coleta de dados, conhecida também como pesquisa de campo. Buscou-se através desta pesquisa de campo revisar as informações que iriam auxiliar na atualização e revitalização do sistema de medidas do TA-2. Assim, identificam-se meios tecnológicos que possam oferecer uma série de vantagens a serem exploradas na segunda modernização do sistema de medidas do túnel de vento do CTA.

Esta coleta abrangeu a pesquisa bibliográfica, para definir a fundamentação teórica utilizada, principalmente com relação às práticas de engenharia de software, que através de uma metodologia pertinente, pudessem ser aplicáveis no desenvolvimento dos softwares de aquisição e redução de dados. As ações das metodologias e as vantagens técnicas para a implantação na prática desses novos conceitos, a busca por técnicas que garantam a qualidade de software e por modelos de processo adequados, são os aspectos mais importantes coletados e estão descritos neste capítulo.

3.1 METODOLOGIAS E MODELOS DE PROCESSO

Toda metodologia de projeto de software usa um método de desenvolvimento ou modelo de processo, podendo incluir, além dos processos técnicos, os gerenciais. Com adoção de metodologia e modelos de processo é possível garantir um desenvolvimento de software completo, correto e documentado, que permite associar as tecnologias com os recursos de gerência e técnicos aplicados durante o ciclo de vida do projeto.

Uma metodologia focaliza a construção do software executando diversas atividades com auxílio de recursos gráficos ou diagramáticos, e enfoca também os aspectos não técnicos como gestão, comunicação, motivação das pessoas e equipe. Através de uma metodologia para o desenvolvimento de sistema de software, procura-

se padronizar boas práticas de engenharia de software, que são vinculadas a técnicas específicas, como por exemplo, a análise estruturada de sistemas, a modelagem dos dados, e também padronizar aspectos gerenciais relacionados às atividades a serem desempenhadas no projeto.

Áreas de aplicações diferentes podem ser suportadas por metodologias diferentes. A decisão sobre qual metodologia adotar fica a critério do desenvolvedor, que pode ser uma organização ou grupo de pessoas responsável pelo desenvolvimento do projeto de sistema de software. Entretanto, a escolha de uma metodologia e sua adequação na prática possibilita representar a melhor descrição do projeto e do processo de desenvolvimento do software, e conseqüentemente significam a garantia de qualidade do sistema de software (GRAY, THAYER, 1991).

Há na literatura (PRESSMAN, 1992) metodologias que vêm sendo elaboradas e evoluídas, que em conjunto com modelos de processos e normas, possibilitam a aplicação em diversas áreas de conhecimento. Todas estas metodologias visam melhorar os processos para garantir a qualidade do produto de software (MAIBOR, ANDERSON, DORFMAN, 1991).

Hoje, empresas e organizações governamentais que desenvolvem softwares procuram melhorias tecnológicas e gerenciais para obter maturidade nos seus processos. Implantar e evoluir os processos, as metodologias, e seguir normas são os caminhos para projetar sistemas de software com garantia de sucesso. O Instituto de Engenharia de Software (*Software Engineering Institute – SEI*) da Universidade de *Carnegie-Mellon* avalia as capacitações de fornecedores de software, considerando a melhoria contínua nos processos de software (PAULK et al, 1993).

O SEI atribui um grau de maturidade, segundo os níveis do *Capability Maturity Model-CMM* (PAULK et al, 1993) ou do *Capability Maturity Model Integration-CMMI* (CMMI-a, 2002), reconhecendo que o detentor do laudo desenvolve um projeto de software com qualidade (RATIONAL SOFTWARE, 2000). Segundo o SEI, os

processos de software são classificados em cinco níveis diferentes, conforme a figura 3.1.

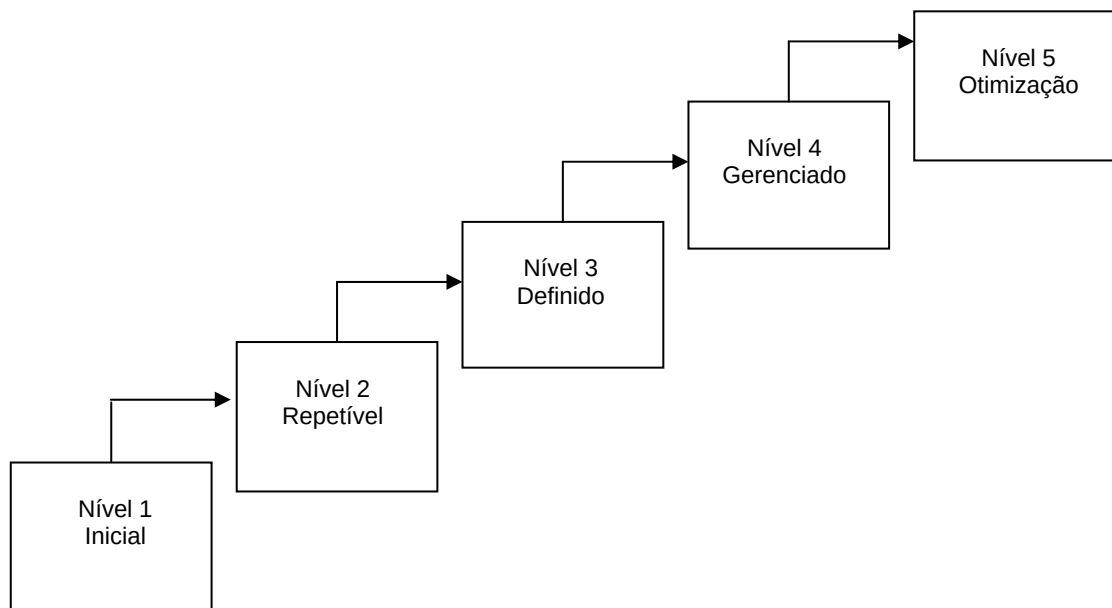


Figura 3.1 - Níveis de maturidade do SEI.

Segundo o CMM, no nível 1, denominado “Inicial”, os procedimentos de gerenciamento ou planos de projeto de softwares de uma organização ainda são pouco eficazes. O software pode ser desenvolvido com algum sucesso pela organização, mas as características de qualidade e o resultado do processo de software serão imprevisíveis. Para o nível 2, denominado “Repetível”, existem procedimentos de gerenciamento formal, garantia da qualidade e controle de configuração implantados pela organização, mas o sucesso do projeto depende de indivíduos da equipe, atuando como uma descrição intuitiva dos processos (RATIONAL SOFTWARE, 2000).

No nível 3, denominado “Definido”, os processos são determinados pela organização e os procedimentos formais estão implantados, de modo a assegurar que estes processos sejam seguidos em todos os projetos de software. São considerados como no nível 4, “Gerenciado”, aqueles processos gerenciados através de um programa formal de coleta de dados quantitativos. Os processos e o produto são medidos e os dados são coletados e fornecidos para as atividades de melhoria de processo. No nível 5, denominado “Otimizado”, a organização está empenhada em

obter melhorias contínuas de processo. Com relação ao modelo de maturidade do SEI, Sommerville (2003) menciona que este é orientado para as metodologias tradicionais e declara o seguinte:

O modelo de maturidade do SEI é uma importante contribuição para a engenharia de software, mas não deve ser tomado como um modelo definitivo de capacitação para todos os processos de software. Esse modelo foi desenvolvido para avaliar as capacitações das empresas que desenvolvem softwares de defesa. Esses são sistemas de software de grande porte, de longo tempo de duração, que têm complexas interfaces com o hardware e outros sistemas de software. Eles são desenvolvidos por grandes equipes de engenheiros e devem seguir os padrões e os procedimentos de desenvolvimento estabelecidos pelo Departamento de Defesa dos Estados Unidos (SOMMERVILLE, 2003, p. 487).

Sobre os níveis de maturidade CMM, Paulk et al (1993) afirmam que há organizações que permanecem no nível 1 por diversos anos, porém, produzem seus softwares específicos com qualidade, e em algumas delas, as normas, os modelos de processo e metodologias são conhecidas, mas poucas implementam software com maturidade.

A realidade das organizações brasileiras é que essa maturidade dos processos somente é alcançada por poucas empresas atuantes na área de software. De acordo com pesquisa do Ministério da Ciência e Tecnologias (MCT, 2002), das empresas que implantaram sistemas de qualidade, a maioria demonstrou estar de acordo com as normas ISO 9001:1994 (NBR ISO 9001, 1994) e ISO 9001:2000 (NBR ISO 9001, 2000), algumas implantaram segundo normas próprias e apenas uma minoria investiu para o reconhecimento do SEI segundo o CMM.

Ainda segundo o MCT (2002) os fatores que contribuem para a permanência das empresas no nível 1 de maturidade dos processos do CMM são:

- A dificuldade de implementar melhorias tecnológicas e de processos.
- Demonstrar e justificar habilidades em diversas áreas de aplicação de software.

- Compatibilizar a gestão da qualidade de software nos diversos níveis hierárquicos da organização.

Com relação às normas ou padrões e o reconhecimento de maturidade dos processos segundo o SEI, pode-se afirmar que os requisitos das normas e padrões existentes são compatíveis e estabelecem correlação com os processos chaves do CMM. Para área de qualidade estas normas são NBR ISO 9000-3 (1991) e ABNT NBR ISO 15100 (2004) da Associação Brasileira de Normas Técnicas (ABNT) e as *Allied Quality Assurance Publications - AQAP* da Organização do Tratado do Atlântico Norte (OTAN). Portanto, caso a organização atenda aos requisitos das normas de qualidade, é possível adotar algumas idéias básicas do modelo SEI.

Antes de tentar classificar processos em níveis de maturidade do CMM, é preciso considerar a importância do processo, da metodologia e das normas para o projeto de software. Na década de 1990, Gray e Thayer (1991) definiram dois tipos genéricos de metodologia de software, que são:

- Enfoque dirigido a documentos.
- Enfoque dirigido a processos.

Esses autores também citam algumas normas de referência nas quais está exemplificado cada tipo genérico de metodologia. Para o enfoque dirigido a documentos é indicada a norma do *Institute of Electrical and Electronic Engineers* (IEEE) denominada *Guide to Software Requirements Specifications* (IEEE STD 830, 1984) e também do *Department of Defense* (DOD) a norma DOD-STD-7935A (1988), e para o enfoque dirigido a processos, a norma DOD-STD-2167A (1988).

Os autores Gray e Thayer (1991) também mencionam as metodologias para uso em determinada fase do projeto de software, como as metodologias estruturada e orientada a objetos e seus desenvolvedores, cujo método de análise, projeto e programação levam seus nomes tais como: método de Yourdon, método de DeMarco

(1989), método de Gane e Sarson (1979), método de Ward e Mellor, método de Shlaer e Mellor, método de Coad e Yourdon e método de Hatley e Pirbhai (1991). Muitos desses métodos foram criados para suprir as necessidades de desenvolvimento de softwares de tempo real da área aeroespacial.

Para a obtenção de qualidade de um produto de software é necessário o conhecimento sobre o que é modelo de processo de software. Este entendimento é descrito por Sommerville (2003) como sendo um conjunto de atividades que são partes dos processos que geram os produtos de software. Na literatura (PRESSMAN, 1992) encontram-se diferentes modelos gerais de desenvolvimento de software, tais como: modelo em cascata, modelo incremental, modelo de prototipação e modelo em espiral. Também se encontram modelos de processos criados especificamente para uso interno de organizações, podendo abranger os processos de qualidade de software. (TAKAHASHI, LIESENBERG, XAVIER, 1990).

Na visão de qualidade (AQAP 150, 1997), os modelos de desenvolvimento de software, ou também conhecidos como modelos de processos, são representações abstratas e simplificadas de uma abordagem sistemática do processo de desenvolvimento de software e que, juntamente com métodos e ferramentas, são os elementos mais importantes do gerenciamento da qualidade (AQAP 159, 1997). Os desenvolvedores desses modelos estruturam os processos em tarefas e atividades lógicas, coordenam e relacionam claramente as atividades de desenvolvimento às atividades de avaliação associadas.

Segundo Pressman (1992) o modelo de processo adotado por uma organização deve descrever claramente todos os processos primários, ou seja, requisitos, projeto, codificação e testes, juntamente com todos os processos organizacionais de apoio, tais como o gerenciamento de projeto, o gerenciamento da qualidade, o gerenciamento de configuração, realizados através do ciclo de vida do software. O uso de normas auxilia a documentar os processos primários e organizacionais de um modelo de processo.

Os modelos de processo, e conseqüentemente, as normas têm evoluído, e vêm melhorando a sua qualidade. Exemplos destas evoluções deram origem a normas tais como: ISO/IEC 12207 (2000), MIL-STD-498 (1994), DOD-STD-2167A (1988), MIL-STD-1521B (1985), RTCA-DO-178B (1992), MIL-STD-2168 (1988) e outras normas específicas usadas por organizações governamentais (ROETZHEIM, 1991). Elas dizem “o que fazer” para ter o projeto de software com qualidade, sem especificar as técnicas, os métodos, o ambiente e a linguagem de programação.

As normas, em sua maioria, são evoluções ou melhorias de normas anteriores. Geralmente as organizações adotam uma norma para o desenvolvimento de seus softwares, e não necessariamente evoluem seus processos a partir do surgimento de uma nova norma, por ser um trabalho um tanto demorado, que implica em uma evolução cultural, e pode afetar sensivelmente a qualidade e a produtividade (PALMA, PASTORELLI, TARANTI, 2005).

A estrutura de uma norma foi concebida para ser flexível, modular e adaptável às necessidades de seu usuário. Para adequação da norma e seu uso juntamente com o processo de desenvolvimento de software escolhido, deve-se entender que ela está fundamentada em dois princípios básicos: modularidade e responsabilidade.

Entenda-se modularidade em termos de processos com mínimo acoplamento e máxima coesão, e responsabilidade em termos de se estabelecer um responsável para cada processo, facilitando a aplicação da norma em projetos onde várias pessoas podem estar legalmente envolvidas. Portanto, uma norma pode ser utilizada com alguma adequação dos modelos de processos existentes.

Existem normas que foram elaboradas para a gestão da qualidade, que têm como objetivo fornecer orientações para a produção de software e demonstração da qualidade, e que podem ser aplicadas por organizações ou empresas desenvolvedoras de software. Essas normas foram elaboradas devido ao crescimento da quantidade de

produtos de software, para os quais hoje existem leis nacionais que os regulamentam (MCT, 2002).

A EMBRAER até recentemente tem desenvolvido projetos de sistema de software fazendo uso das normas DOD-STD-2167A (1988) e RTCA-DO-178B (1992), e atualmente alguns projetos foram iniciados adotando-se a norma MIL-STD-498 (1994). No projeto do Veículo Lançador de Satélite – VLS do CTA (LEMES, 1997), também foi adotada a norma DOD-STD-2167A (1988), e para garantia da qualidade de software a norma IEEE STD 983 (1986).

Apesar de normas e padrões existirem há algum tempo, em software, somente a partir dos anos 1980 a criação e utilização das normas ganhou expressão. Os projetistas de software para a área aeroespacial no Brasil estão migrando seus processos de engenharia de software, aos poucos e de maneira planejada, para normas mais atuais. Devido as dificuldade e complexidade na implementação de novos processos, metodologias e tecnologias, a evolução destes seguindo novas normas de desenvolvimento não avançam em passos muito ousados, para não comprometer a produtividade e a qualidade alcançada dos projetos com uso de normas anteriores. Detalhes de uma coletânea de normas aplicáveis para o projeto de um sistema de software e suas evoluções, podem ser vistos no Apêndice A deste trabalho.

Assim como alguns especialistas têm aperfeiçoado as normas ou padrões aplicáveis em projetos de software, outros têm diversificado também os modelos de processos de software, e outros ainda, têm criado novas metodologias. Algumas metodologias e normas foram criadas para organizar projetos e processos de software específicos. Para estes casos as organizações elaboram e seguem orientações especiais para garantir a qualidade do produto de software, podendo seguir uma ou mais normas, variar de modelos de processos ou adaptar as metodologias tornando-as especiais.

Quando se trata de softwares complexos os desenvolvedores procuram sempre uma forma de apresentar resultados funcionais o mais rapidamente possível. Isso é

possível gerando novas metodologias com enfoque em melhoria dos processos e adaptação das atividades e normas envolvidas. Dessas melhorias surgem projetos de software com qualidade, e com possibilidade de uso dos processos, atividades e documentações em novos projetos, aplicando-se novas normas ou novos processos para dar uma maior visibilidade do desenvolvimento. As metodologias criadas com melhorias de processos são voltadas para diminuir o risco e aumentar a produtividade, sem afetar a potencialidade do software (PASTORELLI, 1998).

Assim, evoluindo a partir dos conceitos sobre metodologias propostos por Gray e Thayer (1991), existem hoje duas linhas de ações aplicadas em desenvolvimento de software, que são:

- As metodologias tradicionais, que procuram enfatizar os processos, planejamentos e documentos.
- As metodologias ágeis que procuram enfatizar as pessoas (BECK, FOWLER, 2001).

Essas últimas são relativamente novas, do final da década de 1990, e baseiam-se em uma série de práticas a serem utilizadas por pequenas equipes, de até 12 pessoas, para o desenvolvimento de projetos pequenos. Ainda não existem casos de sucesso de uso de metodologias ágeis em projetos grandes, críticos e complexos (AGILE MANIFESTO, 2001). No entanto, o uso das ações destas metodologias em projetos grandes e complexos, como a modernização do TA-2, é um dos desafios deste trabalho de pesquisa.

Comparações entre as metodologias tradicionais e as metodologias ágeis têm sido publicadas (SOARES, 2004, 2005), e os detalhes da linha de ação de cada metodologia específica são levados em conta no desenvolvimento deste trabalho. A literatura recomenda o uso das metodologias tradicionais para situações nas quais os requisitos de software são previsíveis e estáveis, e o uso de metodologias ágeis quando estes são mutáveis.

3.2 AÇÕES DAS METODOLOGIAS TRADICIONAIS

As Metodologias de Desenvolvimento de Sistemas (MDS) que surgiram na década de 1970, através das quais se procurava padronizar boas práticas de engenharia de software, eram vinculadas a técnicas específicas, tais como a análise estruturada de sistemas ou modelagem de dados (BELLOQUIM, 2002). O foco excessivo que as empresas colocavam nas atividades de engenharia, em detrimento das atividades de gerenciamento usando estas metodologias, contribuíram para o insucesso de sua aplicação.

Uma das ações principais das metodologias tradicionais são as relações com os modelos de processos, principalmente àquelas metodologias dirigidas a processos. Assim, se fez um estudo dos modelos de processo que vêm surgindo desde a década de 1970. Os modelos de processo abrangem atividades voltadas para o desenvolvimento de software e procuram padronizar as práticas de engenharia de software. O uso destes modelos de processo por organizações e indústrias americanas e européias, no desenvolvimento de softwares aeroespaciais, contribuiu para o sucesso de muitos projetos e a evolução tecnológica de diversos países (PRESSMAN, 1992). Os modelos de processos são abordados a seguir neste trabalho, dos quais o primeiro deles é o modelo em cascata ou modelo clássico.

3.2.1 Modelo clássico ou modelo em cascata

O modelo em cascata foi basicamente o primeiro a ser usado em engenharia de software. É composto de atividades sequenciais e desde a década de 1970 vem sendo aplicado por grandes organizações governamentais do Departamento de Defesa Americano – DOD, e por grandes indústrias desenvolvedoras de software daquele país. Este modelo é derivado de outras engenharias tradicionais, tais como Civil, Naval e Elétrica. O modelo em cascata é um dos exemplos de ações da metodologia tradicional consideradas neste trabalho, e que a literatura aborda como metodologia

dirigida a processos (GRAY, THAYER, 1991). A figura 3.2 ilustra graficamente o modelo clássico ou modelo em cascata.

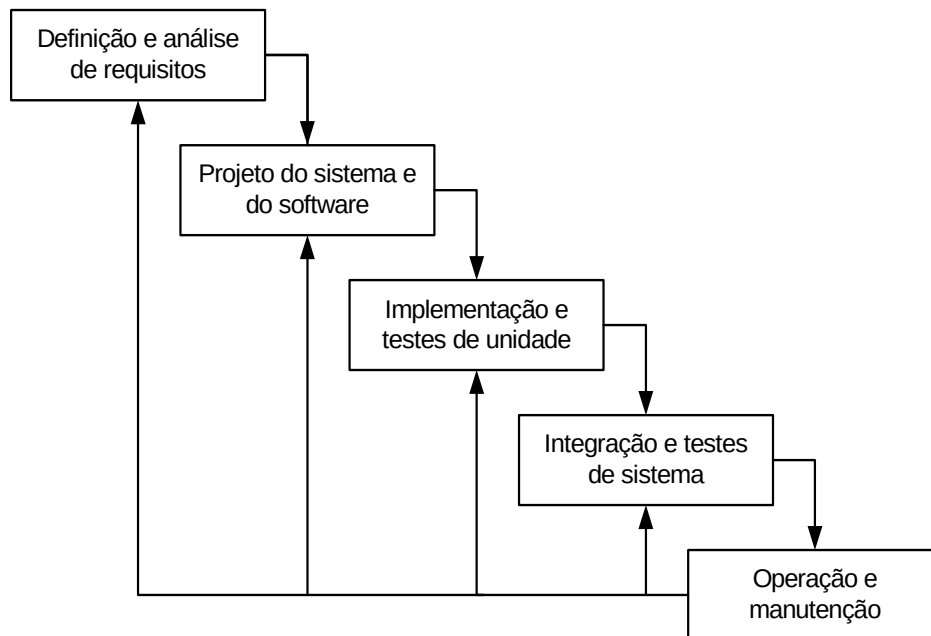


Figura 3.2 - Modelo clássico ou modelo em cascata.

O processo em cascata (LEVINSON, POTAPCZUK, MELLOR, 1999) começa com a fase de definição e análise de requisitos, na qual são identificadas as necessidades do usuário e as capacidades do sistema de software e seus subsistemas, que são documentados através da especificação de requisitos e de interface. Nesta fase os planos de desenvolvimento, de teste, de garantia da qualidade e de gerenciamento de configuração também são documentos que devem ser gerados, podendo ser melhorados nas etapas posteriores.

A etapa seguinte é a de projeto do sistema e do software, a qual envolve a definição de como as capacidades requeridas devem ser projetadas. As documentações geradas nesta fase são o documento de projeto e de interface de projeto, e os planos de teste de integração e de desenvolvimento de componentes de software.

A seguir, na fase de implementação e testes de unidade, o projeto elaborado é codificado na linguagem de programação definida no plano de desenvolvimento,

devendo ser corrigidos os erros de lógica identificados na unidade de software, e registrados de forma apropriada os problemas e as ações corretivas. As documentações desta fase são: o código fonte e objeto, os documentos de procedimento de testes e de relatório de teste de unidade, com o registro de correções praticadas e as identificações estabelecidas no plano de garantia da qualidade e de gerenciamento de configuração. Estes planos são elaborados desde o início do projeto de software e descrevem as atividades de identificação e controle dos componentes de softwares, e as documentações deste projeto e alterações, durante o ciclo de vida de desenvolvimento.

Na etapa de integração e testes de sistema, é verificada a capacidade operacional do sistema de software, que deve estar correta e completa no seu ambiente de operação, e quando erros são encontrados devem ser procedidas as correções e melhorias sucessivas, até que se obtenham resultados aceitáveis. Nesta etapa, os procedimentos e relatórios de teste de aceitação e integração de sistema devem ser documentados, e deve ser gerado ou complementado o documento de descrição de versão de software, além dos relatórios de problemas ou propostas de mudanças com as soluções dadas, completadas e aceitas.

Na etapa de operação e manutenção, o produto de software aceito, entregue e instalado em seu ambiente operacional deve ser monitorado, verificado e seu desempenho validado. Em caso de problemas, estes devem ser corrigidos ou respondidos com mudanças de requisitos, e documentadas as alterações de configurações.

Devem ser completados os manuais de operador de sistema, e de suporte entre outros, bem como o relatório final de verificação e validação de software e os relatórios de mudanças e alterações complementados com as soluções implantadas no produto. Testes executados em nova versão de software devem ser acompanhados pelos envolvidos na proposta de mudança, para qualificação desta versão, com emissão de relatório de resultados e documentação das mudanças nos documentos ou manuais pertinentes, registradas no documento de descrição de versão.

O modelo clássico ou modelo de processo em cascata possui variações de sua representação, todas voltadas para processos, planejamento e documentação do software. As práticas efetuadas visam assegurar a qualidade do software através de eventos de revisão e atividades de verificação e validação, as quais possuem orientações em padrões ou normas, que podem ser estabelecidas para o projeto de sistema de software.

No entanto, para o sucesso com o uso deste modelo, os requisitos do sistema de software devem ser totalmente conhecidos para iniciar o projeto, o que o torna vulnerável quanto a alterações de requisitos durante o ciclo de desenvolvimento. Assim, se faz necessária a formação e implantação de metodologias mais adequadas, especificamente quando os requisitos não são totalmente conhecidos (KRUCHTEN, 2000). Também pode-se usar outros modelos de processos.

Muitas das evoluções de modelos de processo surgiram com intuito de implantar práticas melhoradas de engenharia de software. O modelo de processo incremental (PRESSMAN, 1992), constitui-se uma dessas evoluções.

3.2.2 Modelo de processo incremental

Em 1987, houve uma proposta de variação ou refinamento do modelo de processo em cascata, conhecido como modelo de processo incremental (PRESSMAN, 1992), visando dirimir alguns problemas com as funcionalidades implantadas no software, devido aos novos requisitos ou modificações dos requisitos existentes (FERNANDES, KUGLER, 1990). A figura 3.3 ilustra este modelo incremental.

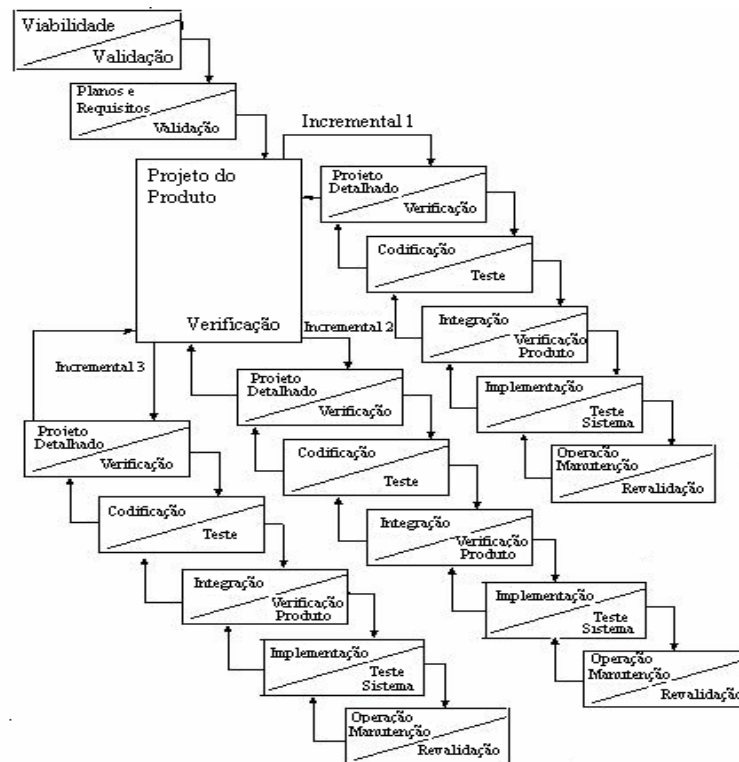


Figura 3.3 - Modelo de processo incremental

O forte deste modelo de processo é que a abordagem incremental gera uma parte do produto de software, à qual através de incrementos são acrescentadas funcionalidades, até que opere de forma integrada ao sistema, com uma participação maior da gerência de configuração. Pressman (1992) considera esta abordagem como apropriada para softwares grandes e tecnicamente complexos, pois permite rapidez na implementação de parte do sistema de software deixando-o em condições de operação.

Um dos problemas do uso deste modelo de processo é que na integração ao sistema maior, os documentos, muitas vezes, acabam por não serem gerados completamente e corretamente, pois as preocupações dos desenvolvedores podem ser direcionadas apenas para a operacionalidade do sistema, uma das atividades do ciclo de vida de um projeto de software que mais atrai a atenção deles.

Uma metodologia de projeto poderia auxiliar na solução deste problema, desde que voltada também para atividades gerenciais. Também pode-se estudar outros

refinamentos do modelo de processo em cascata que foram surgindo, como por exemplo, o modelo de processo prototipação.

3.2.3 Modelo de processo prototipação

O modelo de processo baseado em prototipação tem por objetivo refinar os requisitos a partir do projeto de protótipos de software, facilitando ao detentor do projeto de software expressar os reais requisitos mais claramente, conforme o ciclo de vida mostrado na figura 3.4 (PALMA, PASTORELLI, TARANTI, 2005). Este protótipo é apenas uma versão inicial do sistema de software, cujo projeto pode auxiliar no esclarecimento de possíveis equívocos entre desenvolvedores de software e usuários ou operadores.

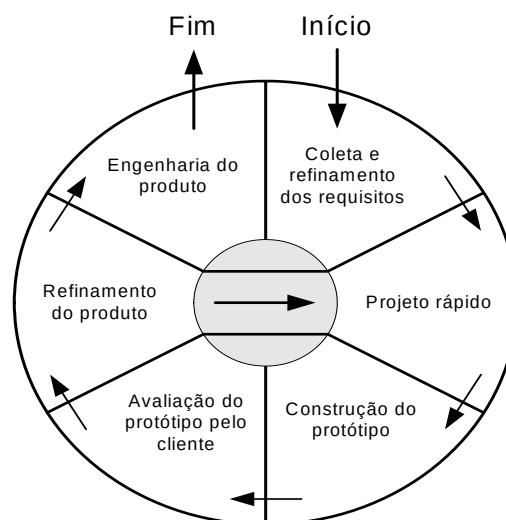


Figura 3.4 - Modelo de prototipação.

A aplicação deste modelo de processo tem como função principal auxiliar no entendimento dos requisitos, sem tanta preocupação com a documentação, o desempenho, a qualidade, a confiabilidade do software e com o gerenciamento de configuração. Um protótipo de software não necessariamente será usado como subproduto do sistema de software final, pois pode ter sido desenvolvido como um software experimental e com enfoque na compreensão dos requisitos dos clientes, e neste caso, é mais viável utilizar apenas o conhecimento com relação aos requisitos e

desenvolver o sistema de software, sem reaproveitar as funcionalidades do código do protótipo.

Este processo começa com a coleta e o refinamento dos requisitos, que irão constituir a versão inicial do documento de especificação dos requisitos. A partir desses requisitos iniciais é realizado um projeto rápido do software ou arquitetura, e constroem-se as interfaces e códigos de um protótipo. Em seguida, este protótipo é avaliado pelo cliente, que fornece as informações e comentários pertinentes ao subproduto. O protótipo é então refinado incorporando modificações conforme as solicitações do cliente e o processo se repete até que o subproduto de software seja aprovado. Com a versão final aprovada, entra-se na fase de engenharia de produto, na qual essa versão congelada do subproduto, denominada protótipo, pode ser melhorada para a demonstração das funcionalidades e interfaces ao cliente.

Segundo Sommerville (2003), hoje em dia o sistema de software com uso de prototipação pode ser compatível com uma abordagem de programação visual. Isso se deve ao uso de ferramentas de apoio ao desenvolvimento de software ou aplicativos comerciais que disponibilizam funções, dados ou componentes de interface, representados por ícones gráficos que podem ser reutilizados, permitindo a construção de forma rápida de protótipos usando esta abordagem. Esse modelo de processo foi concebido para eliminar algumas desvantagens do método clássico. No entanto, o modelo de prototipagem apresenta algumas limitações e questões que devem ser consideradas:

- Um protótipo deve ser construído apenas quando os desenvolvedores podem participar ativamente no projeto com acesso direto ao cliente.
- Deve-se esclarecer aos clientes quanto ao desempenho do protótipo apresentado, ou seja, informar que este não necessariamente será o mesmo entregue como sistema de software final.

- Uma boa análise deve ser conduzida durante todo o processo de prototipagem, e o registro dos problemas e soluções com relação aos requisitos iniciais.
- Mesmo durante a fase de projeto rápido, os documentos devem ser gerados, considerando os problemas levantados e as soluções implementadas no protótipo, que não constavam dos requisitos iniciais, para incluí-los na documentação final.
- Nas concessões do projetista para a obtenção rápida do protótipo, nem sempre é executada uma análise mais cuidadosa dos algoritmos, da linguagem de programação e de outras características operacionais, que de certa forma, podem comprometer a qualidade do produto.
- O sistema de software final não deve ser um protótipo mesmo com as alterações executadas durante o projeto, pois se não houver um gerenciamento das atividades com documentações dos processos, corre-se o risco de se ter um sistema de software mal implementado, sendo que as práticas, métodos, ferramentas e técnicas utilizadas para desenvolver um protótipo, normalmente, são diferentes daquelas utilizadas na implementação de um sistema de software final.

As considerações apresentadas anteriormente levam-nos a uma análise sobre a utilização deste modelo de processo em softwares complexos. Muitas vezes o ciclo de vida de desenvolvimento desses softwares é longo e a participação dos clientes é ínfima. Devido a este modelo de processo não atender por completo as atividades necessárias para obtenção de qualidade de software, ele é recomendado para uso em conjunto com modelos de processos mais elaborados, como o modelo de processo em espiral.

3.2.4 Modelo de processo em espiral

O modelo em espiral, proposto por Boehm (1988), representa o processo de software como uma espiral ao invés de uma seqüência de atividades. Este modelo

considera como uma de suas atividades importantes a influência do risco, como por exemplo, o risco de exceder prazos e orçamentos por não conhecer o sistema de software ou as metodologias contidas em ferramentas de auxílio ao desenvolvimento. Este modelo abrange outros modelos de processo sempre procurando minimizar os riscos através de análise. A figura 3.5 ilustra este modelo de processo.

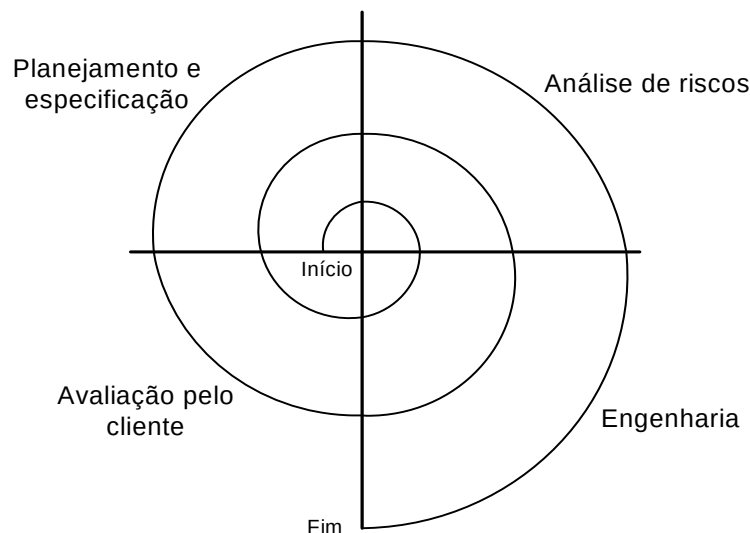


Figura 3.5 - Modelo de processo em espiral

O modelo em espiral é representado por quatro atividades que são:

- **Planejamento e especificação:** na qual se coletam os dados sobre o projeto de software, se determinam os objetivos e as restrições operacionais e de dados do mesmo.
- **Análise de risco:** atividade em que os riscos são previstos e analisados, identificando alternativas para minimizá-los.
- **Engenharia:** aqui são preparadas versões do produto de software através de prototipação ou simulação que serão testadas para apresentação ao cliente.
- **Avaliação pelo cliente:** a versão do produto é avaliada pelo cliente e validado o resultado de engenharia.

Este modelo de processo é apropriado para desenvolvedores que possuem considerável experiência na determinação e análise de risco, pois desse quesito

depende o sucesso do projeto de software. Clientes devem ser convencidos de que essa abordagem evolutiva é controlável e que o custo e o prazo não serão comprometidos. As maneiras de se convencer os clientes de que estes riscos estão sob controle são através de avaliações, análises e demonstrações de reações aos riscos em cada etapa evolutiva. O modelo de processo em espiral é considerado por Sommerville (2003) um processo iterativo que possui similaridade com os processos ágeis, no entanto existe alguma diferença na aplicação da atividade inicial de cada processo.

A diferença mais relevante entre os processos tradicionais e ágeis está na atividade de especificação de requisitos. A primeira atividade dos modelos de processo tradicional é a especificação de software, na qual as funções requeridas pelo sistema e as restrições sobre as alterações e desenvolvimento do sistema de software são estabelecidas. Para processos ágeis o desenvolvimento de software é uma atividade criativa na qual o sistema de software é desenvolvido a partir de um conceito inicial até chegar ao sistema final em operação. A comunidade de engenharia de software ainda exerce a atividade de especificar integralmente os requisitos de software nas etapas iniciais do processo de desenvolvimento, quando estes requisitos são estáveis.

Algumas considerações foram feitas neste trabalho com relação à aplicação de ações de metodologias tradicionais, quando os requisitos não estão totalmente estabelecidos. Algumas vezes são necessárias modificações de requisitos de software em fase adiantada do processo, como por exemplo, na etapa de implementação. Em caso de ocorrência deste fato, a metodologia deve atender aos processos que possam ser afetados.

Esta situação deve ser analisada individualmente, antes do uso de metodologias tradicionais, pois estas podem não permitir práticas que facilitem o processo de modificação de requisitos, em fases avançadas do desenvolvimento. Uma metodologia implantada deve atender às atividades de verificação e validação da solução dada para os novos requisitos de software.

Esses novos requisitos são comprovados através de ensaios do produto de software ou de partes deste software ou subprodutos, ou seja, após os testes individuais com os subprodutos. Estes subprodutos ou produtos são integrados ao sistema de software e ensaiados nos ambientes onde este sistema irá operar, observando todas as interfaces que possam ser afetadas.

A idéia de se especificar totalmente os requisitos de um software antes do início de sua implementação, foi combatida por Brooks (1987), pois alterações nos requisitos de software, muitas vezes, se fazem necessárias em fases subseqüentes do ciclo de vida do desenvolvimento. Segundo o autor, ao adotar um processo tradicional, quanto mais avançado nas etapas de desenvolvimento se procederem as alterações nos requisitos, mais tende-se a aumentar o custo e o risco do projeto, podendo comprometer a qualidade do produto final. Foi com a idéia de solucionar os problemas em relação à especificação de requisitos, que surgiram os movimentos e as metodologias ágeis e seus novos enfoques e ações.

3.3 AÇÕES DAS METODOLOGIAS ÁGEIS

O movimento ágil tem ganhado popularidade nos últimos tempos e tornou-se popular em 2001, quando especialistas em processos de desenvolvimento de software apresentaram novas práticas (BROOKS, 1987). Tem seus princípios fundamentados em boas práticas de gestão e de engenharia de software. Algumas dessas metodologias ágeis enfocam simplesmente a diminuição da quantidade de documentação e de processos, mas consideram como aspectos importantes da agilidade, as seguintes ações:

- O foco nas pessoas e nas iterações.
- A capacidade de indivíduos para tomar decisões rapidamente.
- A adaptação de seus próprios processos.

Alguns exemplos dessas metodologias ágeis são: eXtreme Programming (XP), Scrum, entre outras (SOARES, 2004). Também em 1998, como uma transição das metodologias tradicionais para as ágeis, foram incluídas ações para acrescentar alguns refinamentos ao processo unificado - *Unified Process*, o qual foi transformado em RUP - *Rational Unified Process* (JACOBSON, BOOCH, RUMBAUCH, 1999). O enfoque da maioria das metodologias ágeis está na gestão do processo de desenvolvimento rápido e não linear de produtos de software inovadores.

3.3.1 Metodologia eXtreme programming (XP)

A eXtreme Programming ou XP (BECK, FOWLER, 2001) é uma metodologia ágil para desenvolvimento de software, direcionada a aspectos como: qualidade do código e interação dos desenvolvedores com os itens do planejamento, com respeito às entregas das versões de software. Considera-se durante todo o desenvolvimento a participação do usuário, ou seja, o desenvolvedor deve especificar e projetar o sistema de software com a cooperação dos usuários ou clientes. É estabelecida uma equipe com participantes que constantemente acrescenta ou troca informação sobre o desenvolvimento do software, entre eles e com os usuários. Esta metodologia está baseada em quatro princípios, que são: simplicidade, comunicação, *feedback* e coragem (POLLICE, 2001).

Utilizando a XP (HEDIN, BENDIX, MAGNUSSON, 2003), os requisitos são definidos como uma visão atual, ou seja, sem trabalhar a respeito de “possibilidades” futuras, como meio de aumentar a produtividade. Isto restringe o projeto do sistema de software somente aos requisitos definidos no momento do desenvolvimento do software, não havendo provisão para o futuro. Esta visão para o futuro quando não considerada, pode trazer dificuldades na implementação e um aumento no custo caso seja necessária à atualização no software.

3.3.2 Metodologia *scrum*

A metodologia ágil *Scrum* (SOARES, 2005) tem como objetivo fornecer processos convenientes para o projeto e desenvolvimento de software orientado a objeto. Esta metodologia tem como enfoque estabelecer uma forma de trabalho, para que a equipe produza software flexível para ambientes em constantes mudanças. Na implantação desta metodologia são consideradas as variáveis técnicas e do ambiente, como por exemplo, recursos, tecnologias, requisitos, configuração, entre outros, que podem mudar durante o processo de desenvolvimento de um sistema de software.

A *Scrum* propõe a divisão do desenvolvimento em iterações denominadas *sprints*, cada qual com um período de até 30 dias de duração. A equipe a ser composta para o desenvolvimento de software deve ser pequena, com no máximo 10 pessoas. Esta deve ser constituída por especialistas em projeto, programação, engenharia, e gerência da qualidade, que trabalham para obtenção das funcionalidades do software, definidas como requisitos no início de cada *sprint* (ciclos). O ciclo de vida da metodologia *Scrum* é baseado nas seguintes fases: pré-planejamento, desenvolvimento e pós-planejamento, que podem ser divididas em subfases.

Na fase de pré-planejamento os requisitos são descritos em um documento e priorizados, o qual inclui a definição da equipe de desenvolvimento, as ferramentas os riscos, as necessidades de treinamento e a proposta inicial de arquitetura. O desenvolvimento abrange a definição do ambiente, análise, projeto, implementação e teste. Na fase de pós-planejamento são feitas reuniões para analisar e demonstrar o projeto de software, a qual abrange a integração, testes finais e documentação.

3.3.3 Metodologia de processo unificado

A versão especializada do processo unificado da década de 1960 na qual a empresa *Rational* englobou a visão de processo e produto, com algumas ações evoluídas das metodologias tradicionais é o chamado *Rational Unified Process* (RUP).

Trata-se de uma metodologia na qual o processo de desenvolvimento de software é representado como uma estrutura genérica, que através da adição ou supressão de atividades pode ser adaptada às necessidades do projeto de software de uma organização (RATIONAL CORPORATION, 2000). Os princípios de um processo unificado são:

- Dirigido a Casos de Uso.
- Centrado na Arquitetura.
- Iterativo e Incremental.

Os Casos de Uso são utilizados para guiar o desenvolvimento durante todo o ciclo de vida do sistema de software. Trata-se da representação da seqüência de ações desempenhadas por responsáveis específicos ou pelo sistema de software, e dos resultados obtidos das ações de cada um deles. São adequados na captura de requisitos, análise, projeto e implementação, por expressar desde as perspectivas do usuário de sistema até o produto de software final, ou seja, os requisitos e suas decomposições até a rastreabilidade dos requisitos nas diversas representações dos modelos.

A arquitetura expressa a visão comum que todos os membros da equipe e parceiros devem compartilhar, visando obter um sistema adequadamente robusto, flexível, expansível e de custo viável. Essa arquitetura é especificada em termos de visões de modelo, a partir das atividades ou disciplinas, mais conhecidas como *workflows* ou fluxo de trabalho (PROBASCO, 2000). É na descrição e representação dessa arquitetura que são definidas e identificadas as possibilidades de reuso de componentes, a serem inseridos no sistema de software em construção.

O RUP (RATIONAL CORPORATION, 2000) foi considerado neste trabalho como um processo focado nas ações de uma metodologia ágil, por possibilitar refinamentos dos requisitos de software nas iterações e incrementos, permitir a obtenção e o controle das documentações pertinentes, e descrever práticas com enfoques nas ações de pessoas, cuja arquitetura genérica é mostrada na figura 3.6.

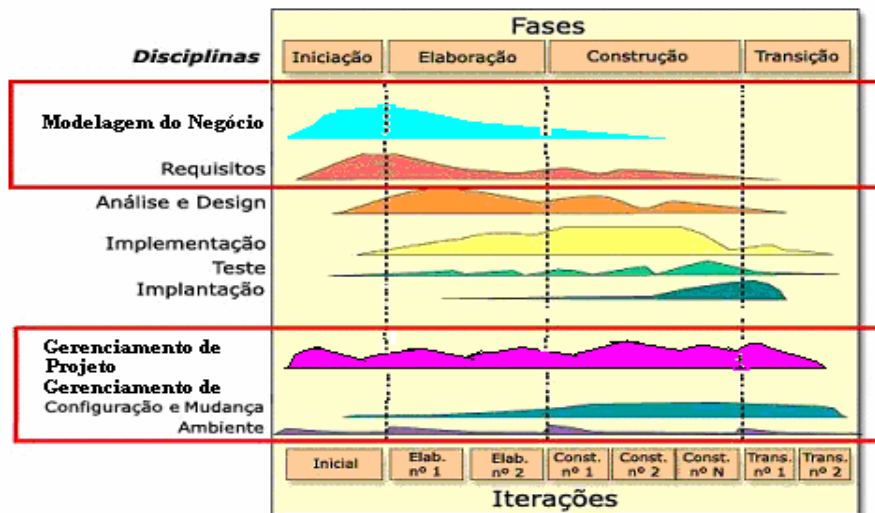


Figura 3.6 - Arquitetura geral do RUP.

Segundo a publicação (RATIONAL CORPORATION, 2000), o processo unificado RUP tem como finalidade garantir a produção de software de alta qualidade, atendendo as necessidades dos usuários dentro de um cronograma e um orçamento previsíveis. Assim sendo, esta metodologia trata das linhas de ações enfatizando pessoas, processos, planejamentos e documentos, nas devidas proporções práticas aplicáveis aos projetos de software, dada a sua flexibilidade.

As características do processo de ser iterativo e incremental são que resultam nas versões do sistema de software, que podem ser liberadas para uso interno ou externo da organização, conforme o projeto. Na modelagem das atividades deste processo, os problemas conhecidos devem ser dirimidos e, a cada iteração, e uma funcionalidade específica do sistema de software pode ser implementada (RATIONAL CORPORATION, 2000). Os riscos devem ser avaliados e analisados, propondo alternativas planejadas para minimizá-los.

As fases desse processo unificado são:

- **Iniciação ou Concepção:** fase na qual é especificada a visão do projeto de desenvolvimento de software.
- **Elaboração:** fase na qual são especificados os recursos necessários e detalhadas as atividades requeridas para o produto.

- **Construção:** fase na qual é produzida uma versão completamente operacional do produto de software.
- **Transição:** fase na qual são executadas as atividades relacionadas à entrega do produto de software aos respectivos usuários.

Essas fases podem ser divididas em iterações que contemplam os fluxos de trabalho, produzindo um mini-projeto, resultando em um incremento no qual a versão do sistema de software conterà uma funcionalidade melhorada ou adicionada, em relação à versão de software elaborada na iteração anterior. O enfoque iterativo e incremental do processo de desenvolvimento unificado provê a avaliação dos riscos ligados aos requisitos, práticas, tecnologias e políticas (KUNTZMANN, KRUCHTEN, 2003). Ao serem desenvolvidos sistemas de software utilizando metodologias ágeis (POLLICE, 2001), presume-se que as seguintes práticas fizeram parte do projeto:

- Programação em par.
- Posse coletiva do software.
- Padrões de Codificação.
- Integração Contínua.
- Planejamento incremental, começando com um plano global e evoluindo ao longo de todo projeto.
- Testes escritos, codificados e testados em conjunto pelos detentores do projeto, seus parceiros e os clientes.
- Gerenciamento de configuração.

A programação em par é o processo utilizado quando grupos de dois ou mais programadores compartilham um único computador para desenvolver o código, *drivers* ou unidades de software, enquanto o outro revisa os códigos quanto à correta implementação e entendimento dos requisitos, ao mesmo tempo (MARTIN, 2000).

Segundo Soares (2005), o desafio das metodologias ágeis é usá-las em grandes empresas e equipes, e encontrar meios de eliminar alguns pontos fracos como a falta

da atividade específica para análise de risco, referindo-se às metodologias XP e Scrum. No caso particular da metodologia RUP focada como ágil, a análise de risco está incluída, dirimindo neste caso um dos elementos que este autor (SOARES, 2005) considerou como ponto fraco.

As propostas gerais do RUP disponíveis na literatura (RATIONAL CORPORATION, 2000) relativas aos fluxos de trabalho, papéis e atividades, servem de base para definição da modelagem técnica e gerencial, como práticas a serem adotadas na metodologia deste trabalho. Além destas práticas do RUP, aquelas referentes ao planejamento das atividades, à programação segundo as metodologias ágeis Scrum e XP, e às diagramações das metodologias tradicionais, também são avaliadas como práticas para compor a metodologia deste trabalho – METAPLI. Usando estas práticas no projeto de modernização do sistema de software do TA-2, a METAPLI, pode ser considerada uma metodologia de projeto inovadora.

Segundo o CMMI (CMMI-b, 2002) as melhorias inovadoras para desenvolvimento de um projeto de software são consideradas tipicamente alterações maiores, que representam uma mudança do modelo de ciclo de vida, podendo também incluir alterações nos produtos para apoio, automação ou melhoria de processos, como, por exemplo, o uso de produtos *Commercial off-the-shelf* - COTS para apoio aos processos. Nesta publicação são citadas como exemplos de melhorias inovadoras:

- Avanços no computador e produtos de hardware.
- Novas ferramentas de apoio.
- Novas técnicas, metodologias, processos ou modelos de ciclo de vida.
- Novos padrões de interface.
- Novos componentes reutilizáveis.
- Novas técnicas de gerenciamento.
- Novas técnicas de melhoria da qualidade.
- Novos processos de engenharia.

Pretende-se com este trabalho de pesquisa prover um projeto de modernização do sistema de medidas do TA-2, com praticamente todas as melhorias inovadoras mencionadas pelo CMMI (CMMI-b, 2002). Para tornar possíveis estas melhorias, se criou a METAPLI, com a abrangência de práticas viáveis, para o projeto de inovação dos softwares de aquisição e de redução de dados.

Com relação aos exemplos dados pelo CMMI (CMMI-b, 2002), no projeto de modernização deste trabalho, as melhorias inovadoras estariam na substituição da plataforma de HP pela plataforma PC, e também na seleção de dispositivos eletrônicos disponíveis comercialmente. Este avanço no computador e produtos de hardware conduz ao processo de avaliações de ferramentas e aplicativos disponíveis no mercado, para apoio ao projeto dos novos softwares de aquisição e redução de dados, visando atividades voltadas para componentes reutilizáveis e uso de padrões de interface.

Para apoio ao processo com uso de melhorias inovadoras, e para a condução do projeto dos novos softwares neste trabalho de pesquisa, faz-se necessário conhecer, além das abordagens das metodologias, dos modelos de processos e das normas, algumas técnicas de modelagem disponíveis em ferramenta CASE - *Computer Aided Software Engineering* (MCMANUS, 1992), para auxiliar no gerenciamento e melhoria da qualidade dos softwares.

Em se tratando de metodologias, várias abordagens de modelagem vêm sendo utilizadas em desenvolvimento de software por pelo menos 20 anos. Algumas dessas abordagens podem ser verificadas em ferramentas CASE que contemplam, por exemplo, as modelagens de metodologias estruturadas e orientadas a objetos ou os conhecidos métodos dos autores citados por Gray e Thayer (1991). Nestas modelagens se enfatiza a representação diagramática e gráfica do processo, o que permite a interação com os usuários durante o ciclo de desenvolvimento.

Segundo Zarrella (1990), as ferramentas CASE permitem prover produtividade do desenvolvimento de software, redução de custo de manutenção e aumento na

qualidade do produto. Estão disponíveis no mercado várias ferramentas que são integradas e que facilitam o desenvolvimento de um projeto de software, mas há diferentes interpretações para o termo “integrado”. Dentre estas interpretações está a que considera integração como o ciclo de vida de desenvolvimento totalmente auxiliado ou suportado pela ferramenta CASE.

Ainda segundo este mesmo autor (ZARRELLA, 1990), usando a interpretação acima, em uma análise de recursos dessas ferramentas, pode-se verificar que muitas ferramentas não suportam interfaces e formatos de controle externo de forma automática, que são requeridos por fluxo de dados ou processos inter-ferramentas. Assim diferentes níveis de integração de produtos e padronizações influenciam na decisão de adoção ou não de ferramentas CASE no desenvolvimento do software.

Zarrella (1990) recomenda também considerar o ambiente operacional (variantes do sistema UNIX e variantes do sistema *Windows*), o ambiente computacional de base, e também as ferramentas e utilitários adicionais que compõem o ambiente de desenvolvimento de software. Neste sentido, as limitações de integração das ferramentas CASE devem ser avaliadas neste trabalho, com considerações em relação aos ambientes de programação visuais comerciais e também aos enfoques pertinentes ao desenvolvimento dos produtos de software, que influenciam na decisão de adotar o enfoque voltado às funções do software ou orientados a objetos.

Ainda com relação às ferramentas CASE voltadas ao suporte para as metodologias estruturadas ou orientadas a objetos, seus recursos disponíveis requerem do usuário conhecimento do processo e do produto de software, para seguir rigidamente as etapas e regras prescritas. Por isso, torna-se um investimento caro, quando o usuário desenvolvedor de software não conhecer suficientemente o sistema de software, as atividades do ciclo de vida e as modelagens das metodologias, aplicáveis em cada fase do ciclo. Apesar do desenvolvimento de software que utiliza ferramentas CASE possuir qualidade elevada, o investimento em capacitação,

equipamentos e recursos humanos demandam um custo elevado e um risco em relação à implementação desta nova tecnologia no desenvolvimento dos softwares.

Além do risco financeiro, outro exemplo de risco pode estar ligado à disponibilidade de recursos da ferramenta CASE, para transformar uma arquitetura especificada do software, em códigos fonte na linguagem de programação adotada como ambiente de desenvolvimento. A ferramenta CASE pode não suportar a transformação automática das representações do projeto em códigos nas linguagens de programação visuais, como também os ambientes de programação visuais podem, por exemplo, não disponibilizar recursos de orientação a objetos. Faz-se necessário neste trabalho, um estudo das ferramentas CASE e de ambientes de programação visual comerciais, bem como a avaliação da integração destes ambientes, para análise de viabilidade de suas aplicações no projeto dos softwares do TA-2.

3.4 AMBIENTES DE PROGRAMAÇÃO VISUAL COMERCIAIS

O uso de ambientes de programação visual é uma prática comum em simulações de engenharia com aplicações em áreas como aeroespaciais, automotivas e náuticas. Esses são tipicamente ambientes de programação em diagrama de blocos que integram ferramentas de modelagem, de análises e gráficas em um único ambiente ou aplicativo comercial (ROBINS et al, 1998).

Existem disponíveis comercialmente muitas ferramentas de programação em ambientes gráficos, que visam simplificar a tarefa de quem desenvolve aplicações que manipulam dados de instrumentos e exercem simulações, através das chamadas linguagens de programação visual, também conhecidas como linguagem “G”, que vêm de “Graphics”, cujo uso tem a mesma potencialidade que as linguagens de programação tradicionais ou linguagens textuais, como FORTRAN, PASCAL, ADA, BASIC e C.

Os ambientes gráficos de programação maximizam a qualidade do software e possibilitam menores gastos com documentação, provendo os seguintes benefícios:

- Caixas de ferramentas viáveis comercialmente.
- Ferramentas de análises que podem ser integradas em outros ambientes.
- Permitem ter códigos em outras linguagens textuais.
- Reuso tanto de bibliotecas quanto de componentes.
- Familiaridade rápida com a programação gráfica.
- Testes de módulos e componentes mais completos e revisados.

ROBINS et al (1998) mencionam exemplos de alguns ambientes de programação visual comercial, disponíveis no mercado e consideradas ferramentas populares como MATLAB[®]/SIMULINK[®] da MathWorks, o MATRIX_xTM/System BuildTM da Integrated Systems, a Easy 5TM da Boeing Aerospace, a ACSLTM da Mitchell and Gauthier Associates, com as quais se elaboram os componentes dos subsistemas logicamente organizados que são parte de sistemas complexos.

Uma ferramenta de programação visual para desenvolvimento de software deve possuir mecanismos para compor os componentes ou unidades em sistemas de software, os quais possibilitem usar recursos de controle e comunicação com demais aplicativos, e interações com usuário, de modo a permitir testes das funcionalidades e capacidade de implementação rápida. Deve ser feita uma avaliação dos recursos disponibilizados pelas ferramentas de programação visuais, bem como, uma análise do uso destas ferramentas na instrumentação de seus laboratórios.

Esta nova tecnologia, que se destina à instrumentação de laboratórios de ensaios, foi utilizada pela EMBRAER e implementada no laboratório de ensaios de trem de pouso, conhecido como RIG Hidráulico. Utilizando-se do ambiente de programação visual comercial da ferramenta *Laboratory Virtual Instrument Engineering Workbench* - *LabView* e de placas de aquisição da empresa *National Instruments*, foi desenvolvido o sistema de software para aquisição de dados e implementado para ensaios de vida em trem de pouso.

Esta empresa, fazendo uso dos recursos do aplicativo *LabView*, gerou softwares com capacidades para executar as funções de aquisição de dados dos sensores, de emissão de comando aos atuadores hidráulicos e de contagem automática de ciclos de ensaio. Muitas outras operações também foram implementadas utilizando-se deste ambiente de programação visual comercial, com relativo sucesso nos ensaios dos trens de pouso das aeronaves projetadas pela empresa.

Com a revisão da literatura em relação às ferramentas e ambientes de programação que possam ser aplicáveis no desenvolvimento de software, este trabalho de tese está voltado para verificação de possibilidades de reuso de bibliotecas e componentes fornecidos nos pacotes comerciais. Também na engenharia de software utilizando as facilidades proporcionadas por uma linguagem de programação visual, reengenharia com uso de linguagens textuais, arquitetura orientada por fluxo de dados. Os estudos dos equipamentos usados no sistema de medidas da plataforma HP, e dos programas existentes em BASIC codificados no final da década de 1970 para operação do TA-2, servirão de base para avaliação dos recursos oferecidos pela ferramenta *LabView* da *National Instruments* e de outras tecnologias.

Pretende-se com uso de novas tecnologias via METAPLI, minimizar os erros de programação e de projeto de software, obter documentação do projeto, ampliar o conhecimento sobre ensaios em túnel de vento e facilitar a obtenção dos dados e resultados, na modernização do sistema de medidas do TA-2. Se os recursos disponíveis, por exemplo, na tecnologia de ambiente de programação visual forem usados de forma integrada, e os dados necessários interpretados corretamente, é possível a compreensão das fontes de erro dos ensaios aerodinâmicos, facilitando ainda mais a elaboração do software com possibilidade de criar uma boa apresentação dos dados, para interface com usuário em termos de telas, campos, botões e menus.

Devido à necessidade de substituir os equipamentos interligados na plataforma HP, que vêm apresentando problemas com manutenção, muitos subprodutos de software podem ser gerados usando *LabView* em conjunto com demais tecnologias.

Esta ferramenta permite o uso de interfaces convencionais com os dispositivos eletrônicos interligados numa plataforma PC, e de certa forma, uma visão para o reuso de componentes das interfaces, disponíveis em bibliotecas.

Outro exemplo de uso desta ferramenta *LabView* na produção do software de aquisição de dados, estaria na representação da situação do escoamento através de instrumentos virtuais, planilha de dados, grandezas do modelo, e interfaces. Estas funções integradas iriam substituir o software existente de aquisição de dados feito em linguagem de programação BASIC, podendo facilitar o entendimento com a representação dos códigos em forma gráfica na documentação do projeto. Estas são as vantagens a serem obtidas através do uso de uma linguagem gráfica.

Esta mesma ferramenta *LabView* e a técnica de reuso devem ser avaliadas para aplicação no projeto do software de redução de dados. O resultado desta análise viabilizará ou não as aplicações na engenharia deste software no TA-2. Deve ser avaliado o grau de complexidade do projeto de software, pois, na redução de dados são executados todo tratamento específico dos dados de ensaios aerodinâmicos. Além disso, os projetos dos novos softwares de aquisição e de redução de dados teriam início ao mesmo tempo, para os quais a atenção estaria voltada para a modelagem dos dados gerados, sem prejuízo para a execução de ensaios em túnel.

Deve haver um equilíbrio entre o custo, a qualidade e o tempo de desenvolvimento, para possibilitar a implantação de novas tecnologias. Assim, este trabalho tem seus objetivos voltados para uso de técnicas de engenharia de software, as quais possibilitem projetar o novo sistema de medidas com equilíbrio de tempo e recursos. Essas facilidades são buscadas nos enfoques em reuso e reengenharia de software.

3.5 REUSO NO CICLO DE VIDA DE SOFTWARE

O ciclo de vida de um software com enfoque em reuso é recomendado pela NASA (NASA SEL 81-305, 1992), através da organização responsável em investigar

a efetividade das tecnologias de engenharia de software, quando aplicadas no desenvolvimento de aplicativos desta organização, no caso, o *Software Engineering Laboratory* (SEL).

O SEL teve como objetivo adequar o ciclo de vida de desenvolvimento de software aos diversos projetos de vários tamanhos e complexidade, ampliando o princípio de reuso de experiências existentes, recomendando o uso para o progresso em qualquer área, pois segundo o artigo NASA SEL 81-305 (1992), a prática do reuso elimina o chamado “reinventar a roda”, reduzindo custos e provendo confiabilidade e produtividade do projeto de sistemas de software.

O artigo menciona que durante todo o ciclo de vida de desenvolvimento de software pode-se obter benefícios com o reuso. Citam-se alguns exemplos de reuso no ciclo de vida, tais como: reutilizar formatos, conceitos chaves, e funcionalidades de alto nível, na fase de definição de requisitos. Houve a afirmação de que se o software não for originalmente projetado para reuso, é mais difícil de incorporá-lo em um novo sistema do que o software explicitamente projetado para reuso.

Em um outro artigo da NASA (MADDEN, 2004) foram descritos os esforços para implantar programas sistemáticos de reuso para desenvolvimento de código fonte de aplicação em veículos lançadores, em suas simulações. Esses programas enfatizam o planejamento e desenvolvimento de recursos de software reutilizáveis para uso futuro. Assim, reuso de software pode ser uma técnica aplicável em diversos tipos de software, dentre eles os de simulação.

O autor Madden (2004) menciona a necessidade de analisar como o reuso afeta o desempenho organizacional e, qual o peso de impacto nas decisões de gerenciamento, no programa de reuso. Enfatiza-se que as medidas de reuso provêm a base para examinar os efeitos do reuso na produtividade, qualidade, custo e tempo para entrega dos produtos.

O especialista da NASA Madden (2004) afirma que a técnica de identificação de componentes para o reuso deve ser realizada no desenvolvimento de software, e consiste em identificar a origem de cada componente, antes de colocá-lo sob o gerenciamento de configuração. Essa técnica não precisa ser automatizada, podendo funcionar na prática se ela for introduzida desde o início do processo de desenvolvimento de software. O autor recomenda que a organização deve procurar investir no esforço manual para identificar a possibilidade de reuso de software.

Para este especialista (MADDEN, 2004) o desenvolvimento de um software com reuso, pode ser traçado através da seleção explícita de componentes pelo desenvolvedor. No entanto, este desenvolvedor pode não estar habilitado para separar partes apropriadas para uso, de partes não apropriadas, para serem reusadas na estrutura de um produto. Este fato pode implicar no reuso de um componente no produto completo, sem que este componente seja composto adequadamente para esta finalidade.

Assim, o trabalho descrito no artigo (MADDEN, 2004) menciona que as técnicas para identificar recursos reutilizáveis na simulação de veículos aeroespaciais, são voltadas para identificação e seleção manual feita pelo próprio desenvolvedor. Os componentes para reuso são identificados e selecionados durante o desenvolvimento do software de simulação que foi implementado em linguagem de programação C++, com estrutura orientada a objeto para tempo real, como parte do programa sistemático de reuso implantado nesta organização da NASA.

Segundo Sommerville (2003), o desenvolvimento de software orientado a reuso é a abordagem na qual todo esforço é direcionado para a identificação e integração de componentes reutilizáveis, durante o processo de desenvolvimento de sistemas de software, sem a necessidade de proceder o desenvolvimento a partir do zero.

A pesquisa sobre elementos para reuso pode ser feita em aplicativos comerciais, por exemplo, começando com as bibliotecas de software disponíveis, nas quais os

componentes de softwares com características de usabilidade estão armazenados. Pressman (1992) descreve que o reuso abrange algoritmos e estrutura de dados, e que muitos componentes reutilizáveis estão contidos em bibliotecas que encapsulam os dados e o processamento aplicado aos dados, permitindo criar novas aplicações a partir de partes reutilizáveis.

Os sistemas *COTS*, também conhecidos como sistemas comerciais de prateleira, mencionados em diversas bibliografias (PRESSMAN, 1992; SOMMERVILLE, 2003) podem ser considerados como sistemas para reuso. Na área aeroespacial, os sistemas *COTS* são soluções confiáveis quando há aplicações em diversas aeronaves, e quando há pouco ou nenhum registro de problemas durante a vida útil de seus sistemas de software.

Devido ao nível crítico dos sistemas de software inseridos em aeronaves, existe uma sistemática de registro de problemas, ou seja, quando ocorrem problemas com a aplicação desses sistemas em alguma aeronave, estes problemas são registrados e divulgados a todos os clientes que adquiriram o sistema *COTS*. Dependendo da gravidade do problema, muitas vezes, as aeronaves são mantidas em solo até que os problemas sejam sanados. Então, evoluções dos sistemas são executadas e validadas através de ensaios pelos detentores do projeto em função dos problemas detectados e relatados pelos clientes, contribuindo para a melhoria dos sistemas de software *COTS*.

Pode-se afirmar que o reuso não é novidade nos sistemas aeroespaciais, a NASA (ANGIER, KELLEY, 1991) vêm utilizando esta técnica no programa da aeronave *Space Shuttle*. O reuso vem sendo aplicado há muitas décadas em softwares embarcados em aeronaves, pois, na oferta de tais sistemas observa-se que muitos desses foram desenvolvidos utilizando-se bibliotecas científicas em linguagem FORTRAN, ADA, *Very-High-Level Languages* – VHLL (WILLIAMS, 1991). Um sistema de software de navegação aérea, por exemplo, pode ser composto de algoritmos bem definidos na linguagem de programação FORTRAN, e estar

disponíveis no mercado em equipamentos, com reuso deste software embarcado em aeronaves específicas.

O desenvolvimento no Brasil de softwares embarcados em aeronaves com característica de reuso ainda tem um domínio de aplicação limitado (MCT, 2002), ou seja, ainda é pouco conhecido o desenvolvimento de sistema de software nacional embarcado em aeronave, que é considerado como sistema COTS disponível para o mercado internacional. As empresas aeronáuticas brasileiras fazem aquisições comerciais desses sistemas COTS para integração em suas aeronaves. Os softwares desses sistemas adquiridos são considerados “caixas pretas”, ou seja, de propriedade dos projetistas dos sistemas de software, não sendo autorizada qualquer alteração neste sistema de software.

Considerando que reuso é muito utilizado na área aeroespacial em projetos de software e simulações, procurou-se estudar as técnicas de identificação de partes reutilizáveis disponíveis em ferramentas de apoio ao desenvolvimento de software. As facilidades de reutilização de componentes de software possibilitariam o desenvolvimento rápido dos produtos de software, e conseqüentemente a disponibilidade dos serviços de ensaios em túnel em menos tempo.

3.6 REENGENHARIA NA REDUÇÃO DE DADOS

A reengenharia de software consiste em reimplementar sistemas ou subsistemas, o que envolve redocumentar, organizar e reestruturar ou traduzir o sistema de software para uma linguagem de programação mais moderna e modificar e atualizar a estrutura de valores dos dados (SOMMERVILLE, 2003). Com aplicação de reengenharia a funcionalidade do software não é modificada e, normalmente, os métodos de engenharia usados para o sistema de software permanecem os mesmos. Em geral, as limitações inerentes ao sistema legado são mantidas para o novo porque a funcionalidade do software permanece inalterada, no entanto, há melhoria significativa

na performance, devido a modernização do hardware, ao uso de outra linguagem de programação, e as reestruturações do projeto e dados.

Pressman (1992) considera a reengenharia como a manutenção preventiva de software, na qual se irá prover uma melhor base para melhorias futuras. Ele menciona também que este termo é comumente usado para caracterizar mudanças de hardware e de outros sistemas físicos.

Segundo este mesmo autor Pressman (1992), muitos componentes de software podem requerer significativa reengenharia, para se tornar competitivos no mercado. A reengenharia para este autor é considerada de baixo custo e uma alternativa para o que chamou de “manutenção de software”. Pressman (1992) recomenda o uso de ferramentas CASE avançadas para a reengenharia, entretanto, a opção para uma aplicação efetiva só deve ser feita após uma análise criteriosa do custo do esforço humano com o aprendizado da ferramenta. Neste trabalho, foi dado enfoque na análise do esforço humano, pois, na formação da equipe de projeto, poucos são especialistas em engenharia de software, e o custo do projeto para a compra, o aprendizado e o uso efetivo da ferramenta CASE para reengenharia, pode se tornar elevado.

Baseado nas declarações de Pressman (1992) relativas a busca de baixo custo de um projeto de sistema de software é proposta a análise da técnica de reengenharia para aplicação no software de redução de dados. A reengenharia para este software pode ser uma solução possível que merece ser estudada, uma vez que pode-se avaliar as melhorias nas estruturas dos dados e do projeto, considerando a urgência na recuperação dos dados e resultados da plataforma HP. A criação de um novo sistema de software para redução de dados pode ser mais dispendiosa, uma vez que os métodos e processos de engenharia não seriam alterados em função da implantação dos novos equipamentos e software de aquisição de dados.

A reengenharia de sistemas de software prolonga sua vida útil e é eficaz em termos de custo e melhoria da estrutura, além de gerar uma nova documentação, que

torna o novo sistema de software de fácil compreensão. Confirmando assim que os custos da reengenharia seriam significativamente menores do que os custos de desenvolvimento de um sistema de software de redução de dados, pois, não seriam alteradas suas funcionalidades.

Para execução de uma reengenharia do software, especificamente na redução de dados do TA-2, é necessário conhecer as funcionalidades dos softwares existentes. Isto implica em conhecer os sistemas de medidas e subsistemas usados nos túneis de vento, para poder documentar as atividades de engenharia envolvidas, implementar corretamente as funções e gerar testes adequados para validação dos produtos de software.

Para auxiliar no entendimento dos subsistemas que compõem o sistema de medidas e os processos de engenharia envolvidos, foram feitas pesquisas com respeito às práticas exercitadas em outros túneis de vento (PAYNE, 1999), principalmente em relação ao processo de calibração de sistemas de medidas de múltiplos componentes. Estas pesquisas tiveram o objetivo de aumentar a compreensão dos especialistas e desenvolvedores dos softwares do TA-2, sobre os sistemas de medidas de outros túneis de vento, para facilitar a implantação de processos de engenharia, de software e gerenciais similares.

Um dos artigos dessas pesquisas trata das práticas aplicadas pelo Centro de Desenvolvimento de Engenharia de Arnold – AEDC (CAHILL, 2001), no qual seus especialistas desenvolveram e implementaram um novo software para reduzir os dados de forças e momentos em túneis de vento, incorporando as maiorias das práticas recomendadas pela *American Institute of Aeronautics and Astronautics / Ground Testing Technical Committee* - AIAA/GTTC, através do Grupo de Trabalho de Tecnologia de Balança – IBTWG. Dentre essas práticas estão os processos para obtenção da matriz de calibração, de nomenclatura comum entre os especialistas em redução de dados.

Essas práticas são recomendadas pelo GTTC em outro artigo que trata de calibração e uso de balanças com aplicação para ensaios em túnel de vento (AIAA-R-091-2002, 2002) do qual o grupo de trabalho IBTWG fez parte. Neste artigo os especialistas consideram o sensor *Strain-Gage* como principal instrumento para medir carregamentos aerodinâmicos em ensaios de modelos, e descrevem o processo de calibração pertinente.

Foi observado neste trabalho de pesquisa, que devido à similaridade com os processos de engenharia aplicados pelos especialistas do TA-2, muitas atividades desse processo de calibração tratados no artigo, podem ser aplicadas também para uma balança externa, como a existente no TA-2. Assim, foi documentado o processo de calibração da balança externa do TA-2 com células de carga, com base na similaridade da calibração da balança interna com sensores *Strain-Gage*, complementando a documentação do sistema de medidas com as demais atividades desse processo.

Com relação a essas atividades praticadas em um ensaio em túnel, foi estudado mais profundamente este artigo do AIAA (AIAA-R-091-2002, 2002), em que seus autores descrevem sobre a importância da aplicação das práticas usadas para medida dos carregamentos aerodinâmicos em túnel de vento, com o uso da instrumentação *strain-gage* em balanças internas. Esse instrumento é aplicado nas diversas atividades dos ensaios em túnel, tais como: no projeto dos modelos aerodinâmicos, na fabricação destes modelos e na calibração do túnel de vento para ensaio destes modelos.

Para uso dos processos de engenharia de um ensaio em túnel é necessário descrever detalhadamente cada etapa do processo, e seguir atenciosamente as várias técnicas usadas no pré-ensaio e durante o ensaio com acompanhamento via software. O processo de calibração de uma balança é o principal processo de engenharia, e um erro técnico afeta a precisão dos carregamentos medidos pela balança e armazenados via software, sendo que parte deste processo é feita de modo manual, porém deve ter conformidade com a armazenagem automatizada dos dados.

Outro exemplo de processo de engenharia que pode ser considerado é o de calibração da instrumentação ou de um dispositivo eletrônico usado no ensaio, que também pode ser automatizada. Este processo, descrito e disseminado ao grupo técnico, possibilita a execução provendo a garantia da precisão dos dados armazenados de forma computacional, e, conseqüentemente, a determinação correta dos coeficientes aerodinâmicos através do software de redução de dados de acordo com o fenômeno físico ocorrido no ensaio.

Também podem ser acrescentados processos de análise usando aplicativos de software comerciais antes da redução dos dados, para garantir a qualidade dos dados advindos da etapa de aquisição, podendo evitar a progressão da incerteza de informações com respeito aos processos de engenharia.

Neste trabalho então, foram pesquisados alguns processos de calibração aplicáveis aos tipos de balanças de um túnel de vento. Para calibração de balança interna estudou-se o processo de calibração recomendado pelo AIAA (AIAA-R-091-2002, 2002). Para calibração de balança externa, estudou-se o método de calibração de sistemas de medidas de múltiplos componentes (NOGUEIRA, 1980). O estudo deste processo e método de calibração serve de base para o entendimento, identificação e documentação das funcionalidades do software de redução de dados, e auxiliarão na reengenharia deste.

Para este trabalho de pesquisa, no âmbito de inovação tecnológica investiu-se no aumento da qualidade e na capacidade de realização de ensaio em túnel de vento, por considerar uma tecnologia indispensável ao desenvolvimento confiável e com grau de sigilo almejado para os projetos aerodinâmicos nacionais. Com a implantação de novos equipamentos, instrumentos e dispositivos eletrônicos, faz-se necessário uma avaliação quanto ao uso de ferramenta de programação visual, que possibilite desenvolver o software do sistema de medidas, e forneça um suporte para reduzir o tempo de ensaios em túnel de vento, e conseqüentemente o tempo de desenvolvimento de projetos.

Novas tecnologias aplicadas em conjunto com ações de metodologias facilitariam a organização dos processos de engenharia e de software, no desenvolvimento do projeto de sistema complexo de software, como do túnel de vento. Neste trabalho de pesquisa são analisadas as tecnologias e consideradas as adequadas ao projeto em questão.

O foco das iniciativas de melhoria nos processos de engenharia e de gerência de um projeto de software vem sendo o grande desafio das organizações de software. Ao implantar metodologias adequadas no desenvolvimento de softwares, possibilita a obtenção de melhoria da qualidade, a integração das equipes de desenvolvimento, a gestão do conhecimento, as demonstrações dos requisitos do sistema de software, entre outros objetivos, os quais integrados contribuem para o sucesso do produto de software final.

Uma das propostas deste trabalho, para a utilização plena do sistema de medidas do túnel de vento, é a implantação de práticas via METAPLI. Esta metodologia foi criada baseada em práticas de engenharia de software, focando o amadurecimento dos processos técnicos e gerenciais para o uso em outros projetos. No entanto, existe uma aderência aos processos de soluções de problemas (NARAYANAN, 2001) e conceitos de gestão de conhecimento (NONAKA, TAKEUCHI, 1997).

Para Narayanan (2001) no nível de uma firma individual, mudanças tecnológicas podem ser descritas como um processo de solução de problemas, contendo os estágios de reconhecimento de problema, seleção de tecnologia, desenvolvimento da solução e implementação/comercialização. Esta proposta de solução de problema é apresentada no modelo genérico da figura 3.7.

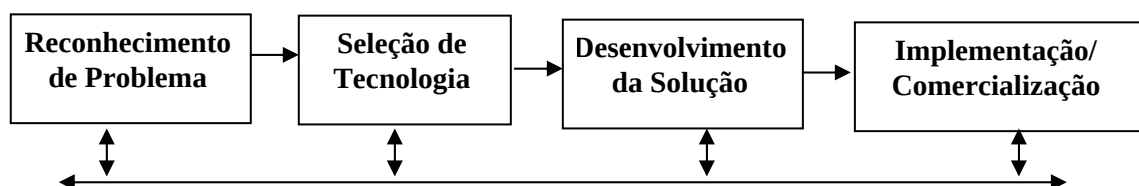


Figura 3.7 - Modelo genérico de solução de problemas (NARAYANAN, 2001).

A proposta de solução de problema apresentada por Narayanan (2001) é considerada genérica, e as ações da METAPLI são aderentes à proposta deste autor. A proposta de Narayanan (2001) é caracterizada por uma seqüência de etapas de processo em que um conjunto de metas deve ser cumprido, como soluções de problemas para transformação de um sistema. As ações da METAPLI foram detalhadas de forma inovadora para a concepção da metodologia de projeto deste trabalho, pois se faz uso das ações das metodologias tradicionais e ágeis descritas na literatura, de forma híbrida e na prática em sistemas de software grandes e complexos.

Os conceitos de gestão de conhecimento (NONAKA, TAKEUCHI, 1997) também são considerados neste trabalho. Os enfoques dados por estes autores para inovação tecnológica também são aderentes aos empregados via METAPLI. O objetivo da metodologia é prover em cada etapa ou macro-funções o detalhamento de várias propriedades gerenciais e técnicas, com as descrições das práticas para gerar e documentar os acontecimentos do projeto de modernização do sistema de medidas do TA-2.

A metodologia também levará em conta os fatores externos que influenciam na qualidade do produto, como a novidade de uma aplicação. Esta novidade tem levado a comunidade de engenharia de software a discutir a qualidade de software, com o objetivo de aumentar a produtividade de suas organizações. Muitos dos novos conceitos e abordagens estão voltados para a qualidade do processo de desenvolvimento de software sem que prejudiquem a qualidade do produto.

Estes conceitos e abordagens em sua maioria estão voltados para a liberação rápida de subprodutos deste sistema de software, desde que corretos, completos e com a documentação segundo normas de software. A escolha de uma nova tecnologia, que requeira muito tempo para capacitação de pessoal, por exemplo, pode não ser um bom caminho a trilhar, podendo comprometer o ciclo de vida de um software, o qual é representado através de metodologias que contemplam os modelos de processo.

A qualidade do projeto de software na década de 1970 foi encarada por alguns desenvolvedores como entrega rápida do código executável, levando-os a criar e implementar processos e atividades de formas variadas, não havendo uma uniformidade ou um consenso para o processo de geração de produtos de software. Segundo Yourdon (1992), cada organização deve implementar diferentes prioridades para os enfoques de produtividade e qualidade, e para deixar de usar atividades informais e/ou intuitivas, pode investir em: melhores linguagens de programação, pessoal, ferramentas para automatizar o processo, prototipação, técnicas estruturadas, metodologias orientadas a objetos, reuso e reengenharia de software.

Partindo da idéia de buscar a qualidade de software, foi desenvolvida e implantada a metodologia deste trabalho, juntamente com técnicas, métodos e práticas de engenharia de software e gerenciais, para representar o desenvolvimento dos softwares do sistema de medidas de um túnel de vento. Para evitar a pouca qualidade de software que ocorre devido ao emprego de atividades informais, foram direcionados esforços para contornar problemas como processos improvisados, dependência de talentos individuais e práticas de engenharia de software não padronizadas.

Em um projeto de sistema de software com qualidade é possível a identificação das tecnologias e de modelos de serviços. No Brasil, as empresas ainda se encontram em fase intermediária, almejando que os produtos e serviços de software ganhem mercado internacional, e possam ser oferecidos e desenvolvidos com garantia de sucesso e reconhecimento de maturidade (MCT, 2002).

Para isso, a implantação de práticas através de uma metodologia adequada é fundamental, objetivando prover toda a documentação dos processos de engenharia, de software e gerenciais, e ampliar o conhecimento de membros da equipe. A fim de atender essa necessidade foi criada a METAPLI, com o enfoque em qualidade já mencionado anteriormente, e implantada no desenvolvimento dos softwares de aquisição e redução de dados do TA-2.

CAPÍTULO 4 METODOLOGIA DE PROJETO DE SOFTWARE

Para o projeto de modernização do TA-2 foram usados os recursos de engenharia, de software e gerenciais que permitissem atingir as expectativas de qualidade, de produtividade e de confiabilidade dos dados e resultados dos ensaios do sistema de medidas do TA-2 modernizado. Assim, a METAPLI foi desenvolvida considerando as atividades requeridas no projeto, as seqüências destas atividades, os ambientes de desenvolvimento, as informações e análises requeridas em cada fase do desenvolvimento, os critérios de qualidade de cada fase e os tipos de pessoas designadas para tomar certas decisões no projeto dos softwares.

Os materiais, métodos, processos, técnicas e práticas utilizadas neste trabalho foram levantados em informações históricas dos ensaios no TA-2, em descrições de bibliografias pertinentes e em informações colhidas de algumas indústrias do ramo aeroespacial que automatizaram seus sistemas de medidas, através da utilização de equipamentos, de dispositivos eletrônicos, de software, entre outros recursos, para medir os fenômenos físicos que ocorrem em ensaios laboratoriais ou em campo.

4.1 A METAPLI

A METAPLI é constituída de ações das metodologias tradicionais e ágeis voltadas para solução de problemas práticos de projetos de software grandes e complexos. Estas ações são aplicadas ao projeto dos novos softwares de aquisição e de redução de dados, para evolução do sistema de medidas do TA-2. As ações da metodologia visam à qualidade da engenharia de sistema de software durante o projeto de modernização. As práticas estão voltadas para os aspectos de reuso, reengenharia, tecnologias que abordam a qualidade de software, e projetos nos quais os softwares desempenham o papel principal.

Todo sistema de software precisa ser modelado como um conjunto de componentes e de relações entre esses componentes. Isso normalmente é representado como um diagrama de blocos e suas interconexões. Esse esquema foi utilizado tanto para representar os sistemas de medidas e seus subsistemas, como para representar a METAPLI e suas práticas, conforme indica a figura 4.1 gerada pela autora deste trabalho de tese.

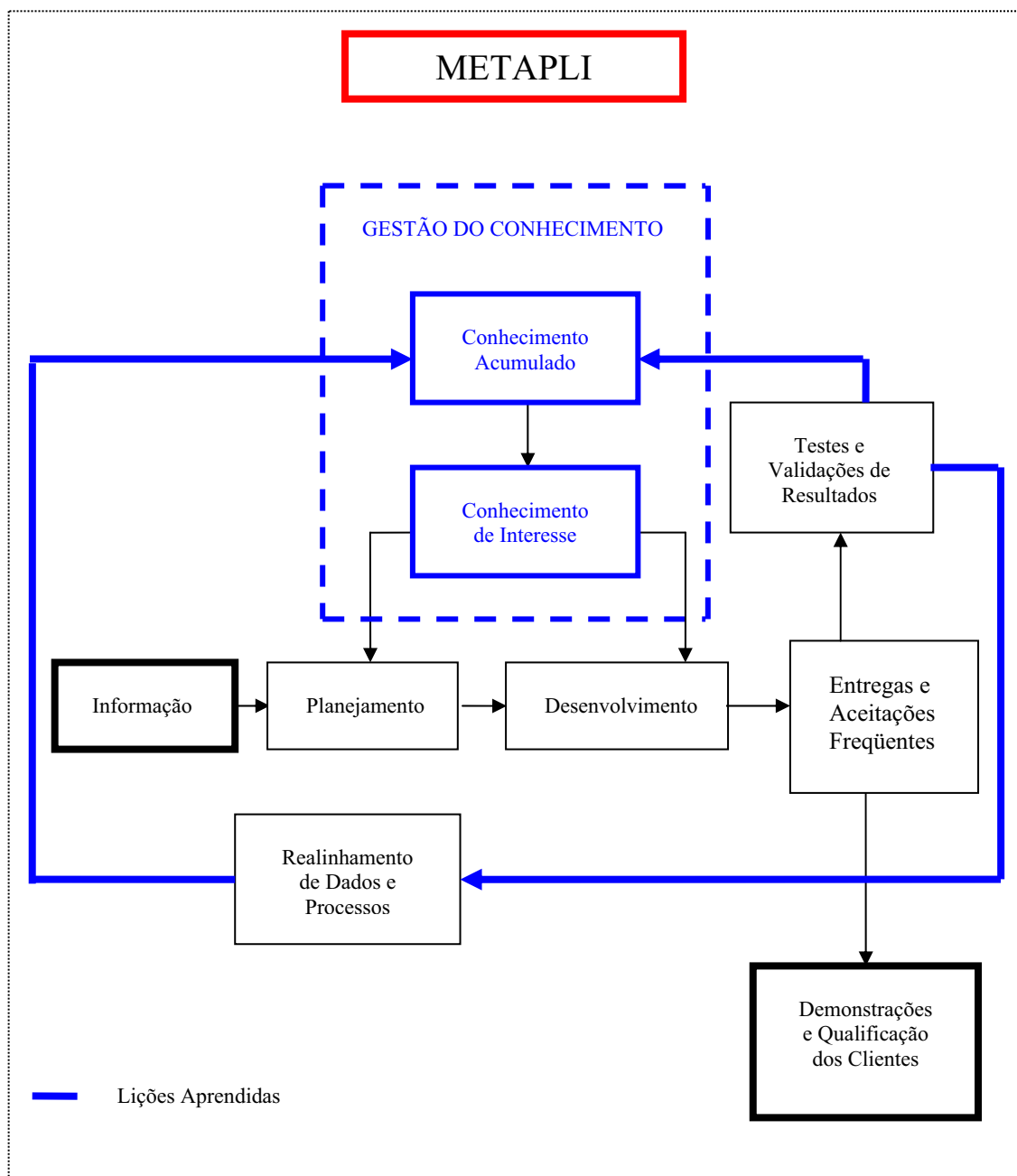


Figura 4.1 - Metodologia especial tradicional e ágil de aplicação em projeto legado de software em inovação – METAPLI.

As práticas da METAPLI incluem macro-funções gerenciais e técnicas para o projeto de revitalização e atualização dos softwares de aquisição e redução de dados do TA-2. As macro-funções incluem também a melhoria dos processos, a definição da arquitetura geral dos subsistemas e do sistema de medidas, e a integração desses subsistemas ao sistema principal com novos softwares operacionais. Esta metodologia se baseia nas seguintes práticas:

1. **Planejamento:** consiste no levantamento das necessidades dos clientes com respeito aos ensaios em túnel e na decisão conjunta do que é necessário ser feito em cada ciclo de desenvolvimento, estabelecendo o que poderá ser criado, aperfeiçoado e/ou melhorado posteriormente, em relação aos processos técnicos e gerenciais para o desenvolvimento dos subprodutos de softwares, os quais serão integrados ao produto final no projeto de modernização do sistema de medidas do TA-2.
2. **Desenvolvimento:** consiste na identificação e implantação de variáveis técnicas e do ambiente de desenvolvimento de software, e execução de meios para o gerenciamento da qualidade e de configuração dos subprodutos e do produto final. Também consiste no exercício de um controle contínuo e da verificação da qualidade da documentação gerada em cada ciclo de desenvolvimento, o qual engloba as atividades de requisitos, análise, projeto, implementação e testes de cada subproduto até a obtenção do produto final.
3. **Entregas e Aceitações Frequentes:** consiste no estabelecimento de critérios para a aceitação dos subprodutos de software, na determinação de marcos para entregas e na identificação e documentação das iterações e de fatores de riscos. Trata-se de definir como será a avaliação do atendimento aos requisitos especificados e do conhecimento atingido pelo pessoal envolvido no desenvolvimento do sistema de software.

4. **Testes e Validações de Resultados:** consiste na execução de testes, com a participação dos envolvidos no desenvolvimento, nos subprodutos de software entregues e integrados ao sistema de medidas do TA-2, com enfoque na validação dos processos de desenvolvimento de software e de engenharia, que devem ser comprovados através da documentação gerada e dos dados obtidos após processamento, em confronto com os requisitos estabelecidos para os subprodutos e para o produto final.
5. **Realinhamento dos Dados e Processos:** focaliza o aperfeiçoamento dos processos de desenvolvimento de software e de engenharia durante o ciclo de vida do projeto do sistema de medidas, assim como a verificação da conformidade com os modelos de processos estabelecidos para os softwares de aquisição e redução de dados. Focaliza também a simplificação da estrutura dos dados sem perder as funcionalidades das unidades de software e do subproduto em operação integrada ao sistema de medidas.
6. **Demonstrações e Qualificação dos Clientes:** focaliza a integração, testes e documentação final dos processos de engenharia, de software e gerenciais do sistema de software, com a demonstração das funcionalidades dos softwares para a qualificação dos clientes, para o emprego efetivo nos ensaios dos modelos aerodinâmicos.

As abordagens gerenciais e técnicas da METAPLI aplicadas ao sistema de software do túnel de vento foram modeladas para as necessidades do grupo de projeto de software deste trabalho de pesquisa. Esta metodologia provê sua fundamentação nas práticas de engenharia identificadas e descritas na literatura, com enfoque na gestão de processos que auxiliam nas soluções de problemas e oferecem oportunidades de aplicação em projetos futuros.

A METAPLI foi gerada de modo a englobar os processos de desenvolvimento de sistema de software e as práticas pertinentes. Esta implantação também abrange

diversas atividades, com tecnologias que possibilitam prover os enfoques adequados para a criação dos softwares de aquisição e de redução de dados, voltadas para uma plataforma ajustada ao ambiente de desenvolvimento.

A ação de entrada para a METAPLI é a coleta de informações sobre as atividades de engenharia de software do TA-2. Esta coleta engloba o entendimento do sistema de medidas e dos softwares existentes, o estudo das tecnologias disponíveis no mercado, o levantamento das pessoas envolvidas no projeto e suas especialidades, um cenário da produtividade, disponibilidade e manutenibilidade do sistema de software, um levantamento das experiências com uso do sistema de medidas da HP e uma análise dos requisitos específicos dos clientes.

As coletas destas informações, a princípio, foram feitas na literatura e em campo com o acompanhamento de ensaios, observação dos procedimentos e processos empregados, e em relatos destes ensaios realizados com os softwares de aquisição e redução de dados do TA-2.

4.2 METODOLOGIA E PROCESSOS APLICADO NO TA-2

A metodologia adotada originalmente pelos desenvolvedores de software do túnel de vento do CTA, no fim da década de 1970, era assistida por práticas voltadas aos aspectos técnicos como concepção, desenvolvimento e testes destes produtos de software. A nova metodologia proposta, a METAPLI, contempla os aspectos técnicos e equaciona os aspectos gerenciais em paralelo, através de práticas que minimizam os erros e maximizam a rapidez para desenvolvimento do projeto dos softwares de aquisição e redução de dados.

Para desenvolver os novos softwares de forma correta e completa houve a necessidade de se obter as informações e o entendimento sobre o sistema de medidas existente, uma vez que este seria mantido em operação até ser possível a obtenção dos

dados e resultados de ensaios, para comparação com os obtidos com os novos softwares. Por isso se preocupou em descrever o sistema de medidas existente para a plataforma HP, que foi usada até ser possível a sua substituição.

O sistema de medidas existente no TA-2 foi todo projetado para plataforma HP, com softwares de aquisição e redução de dados em linguagem de programação BASIC, os quais eram alterados em cada experimento a ser ensaiado. Seus equipamentos ou instrumentos de medidas eram interconectados ao computador HP para execução das funcionalidades operacionais deste sistema de medidas e vinham apresentando panes.

Este sistema de medidas da plataforma HP deveria ser desativado, mas não o conhecimento sobre seu funcionamento, que embora fosse fragmentado, devido às diversas manutenções efetuadas nos softwares e nos hardwares, seria necessário entender suas funcionalidades para a adoção de recursos tecnológicos viáveis, possibilitando também a documentação do projeto de modernização.

A documentação do sistema de medidas e de seus subsistemas neste trabalho demonstra a visão macro das funções do operador, do hardware e do software, para efeito de comparação dos resultados entre ambas as plataformas HP e PC. A relação dos equipamentos da plataforma HP para aquisição de dados e dos softwares em linguagem BASIC é apresentada no Apêndice B deste trabalho.

Com este levantamento se obteve a informação de que o maior problema de cada software em linguagem BASIC é o processo de manutenção deste, no qual eram feitos os ajustes dos dados de cada projeto aerodinâmico, bem como os ajustes de configurações dos equipamentos HP, para sincronizar as leituras corretamente. Isso implicava em possibilidades de erros do especialista na alteração dos softwares, alteração esta pouco documentada, sendo que na realidade gerava-se uma nova versão do software para cada programa ou campanha de ensaio.

Tais erros eram difíceis de serem detectados, pois existiam diversos processos de engenharia envolvidos, e várias manipulações dos dados de aquisição na execução do software durante o ensaio. Por exemplo, os dados eram processados e armazenados com as médias das leituras e dessas já descontados os zeros iniciais de cada dado real. Este processamento dos dados na aquisição via software dificultava ainda mais o entendimento dos fenômenos físicos, quando os dados eram usados pelo software de redução de dados para obtenção dos resultados de calibração ou de ensaios aerodinâmicos.

Para se ter uma visão dos equipamentos de aquisição de dados da plataforma HP, um diagrama esquemático simplificado é apresentado na figura 4.2. O desenho esquemático utilizado representa a cadeia de medidas da chamada Seção de Controle, que tem os equipamentos da HP interligados conforme modelagem apresentada em outros trabalhos abordando o túnel de vento (REIS, 2000).

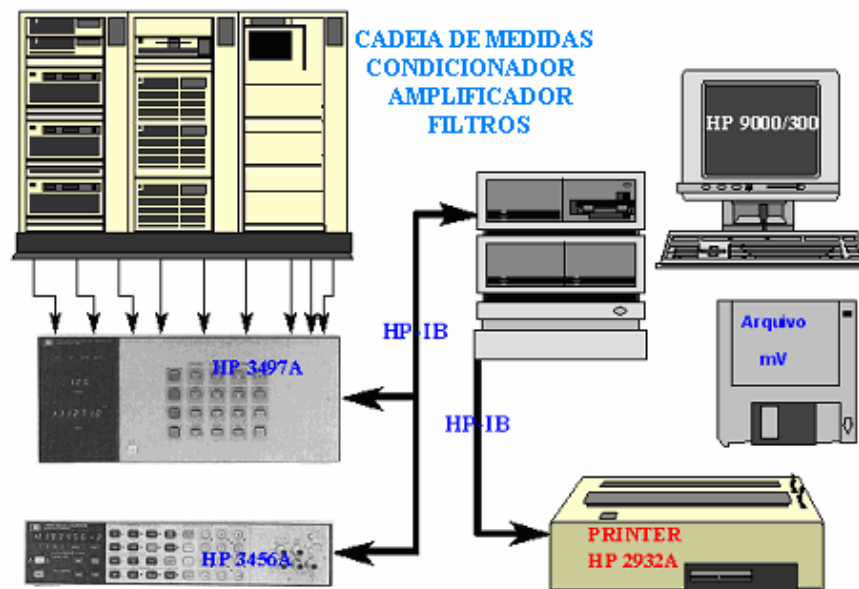


Figura 4.2 - Diagrama esquemático da aquisição de dados com a plataforma HP.

Para este sistema de medidas antigo, utilizava-se o computador HP 9000/300, o cabo de comunicação HP-IB e a impressora HP 2932A. Além destes instrumentos, também se empregavam diversos softwares mencionados anteriormente (Apêndice B), para obtenção dos resultados dos ensaios em túnel. Os softwares de aquisição de dados

continham comandos para os instrumentos executar funções, tais como: o cálculo da média dos dados de leitura através do multímetro, as seleções dos canais de leitura do *scanner*, e o sincronismo das leituras dos instrumentos do TA-2.

Sabe-se que quanto maior o porte e a complexidade do sistema de software a ser projetado, mais aumenta a necessidade de planejar melhor o desenvolvimento e a aceitação do produto, podendo acarretar variações no tamanho da equipe de desenvolvimento. O planejamento, então, é a primeira prática da METAPLI.

4.3 PLANEJAMENTO DOS NOVOS PROCESSOS VIA METAPLI

Na primeira prática da METAPLI, o planejamento, fez-se uma avaliação da qualidade e da importância do sistema de software existente. Esta começa com um plano de avaliação que faz uso de algumas técnicas, para diagnosticar o sistema de medidas existente e direcionar esforços para a melhoria da qualidade, podendo considerar inicialmente os aspectos técnicos do sistema de software modernizado, que são:

- Descrever o sistema de medidas existente.
- Definir o novo sistema de medidas.
- Identificar os subsistemas.
- Definir as interfaces.
- Identificar os dados e arquivos.
- Gerenciar o escopo do sistema de software.
- Refinar a definição do sistema de medidas.
- Gerenciar requisitos variáveis.

Um documento denominado Plano de Avaliações foi elaborado para inicialmente descrever o levantamento da experiência e familiaridade dos especialistas do TA-2, na implementação de modelos de processo. Foi avaliado o processo de desenvolvimento

de software adotado para o sistema de medidas anterior, buscando informações para a viabilidade de reaplicá-lo no projeto dos novos softwares de aquisição e redução de dados.

Para caracterizar as atividades de engenharia de software desempenhadas pelos especialistas do TA-2 no desenvolvimento dos softwares existentes, foi feito um diagnóstico do grau de aplicação das etapas de um processo de software, coletando dados através de entrevistas com os envolvidos, quanto à aplicação de cada atividade de um processo no sistema de software. A consolidação e a validação dos dados coletados das observações, derivados de evidências ouvidas e vistas em alguns ensaios acompanhados, são apresentadas na tabela 4.1.

Tabela 4.1 - Atividades de engenharia de software executadas no TA-2

Atividade	Execução (%)
Definição de Requisitos	1%
Análise de Requisitos	1%
Projeto do Sistema e de Software	2%
Codificação ou Implementação	6%
Teste de Integração	2%
Operação e Manutenção	80%
Garantia da Qualidade	5%
Gerenciamento de Configuração ou Controle de Mudanças	2%
Gerenciamento de Projeto	1%

A tabela 4.1 demonstra que 80% das atividades executadas pelos especialistas do TA-2 se concentrava na operação e manutenção dos softwares, o que seria aceitável se não houvesse novos requisitos dos clientes a serem atendidos. Quando há novos requisitos ou qualquer alteração destes deve ser refeita a documentação de software, o que implicaria na alteração de documentos e representações geradas em todas as atividades do processo de engenharia de software. Para os conceitos de engenharia de

software atuais, a manutenção de software é considerada apenas em casos excepcionais.

Através dessa avaliação constatou-se que os processos do túnel de vento em relação à engenharia de software e a maturidade, segundo o CMM (PAULK et al., 1993), estava no nível 1, considerado inicial. A maioria dos esforços dos especialistas se concentrava na atividade de manutenção e operação dos softwares existentes, tendo pouco domínio de conhecimento das demais atividades. No entanto, foi possível através da implantação da METAPLI gerar o Plano de Avaliação, e passar por uma etapa de auto-avaliação para poder iniciar um processo de melhoria que se efetiva com este trabalho.

O objetivo da manutenção de software é preservar a integridade do sistema de medidas, ou por necessidade de melhorias, ou para adaptações. Toda vez que o produto de software precisa de modificações, o processo de desenvolvimento de software deve ser invocado, para se efetuar e completar corretamente as modificações, para que não haja perda de informações do sistema de medidas. A manutenção de software é uma atividade intensa em conhecimento, e se não houver domínio do conhecimento do sistema verificam-se atrasos no cronograma de ensaios, e outras implicações como a possibilidade de introdução de novos erros ou “bugs” durante a manutenção.

Com essa avaliação foi possível observar que o modelo de processo seguido anteriormente era informal, havendo pouca preocupação com documentação, uso de metodologia e de processos de um projeto de software. Por se tratar de um projeto de sistema de software complexo com pouca documentação a respeito, concluiu-se que seria inviável adotar o mesmo processo, devido a este concentrar as atividades apenas na etapa de manutenção e operação do ciclo de vida do desenvolvimento de software.

Não cabe discutir se este processo adotado para a época foi um fato positivo ou negativo, mas a decisão para este trabalho é a de não utilizá-lo por considerá-lo um

modelo de processo limitado, que produziu um sistema de medidas carente de documentação e de qualidade com o passar do tempo. Devido a essas limitações para o sistema de software existente, uma parte importante dos ensaios aerodinâmicos era entendida por poucos integrantes da equipe do túnel.

Esta carência de conhecimento na área de engenharia de software e com domínio de poucos integrantes, prejudicou inclusive a prática de alterar o código e ajustar ou inserir os diversos parâmetros exigidos no ensaio, ou seja, ficou difícil proceder a modificação do código fonte e torná-lo logicamente adequado para o ensaio do modelo aerodinâmico, pois, qualquer modificação no sistema de software era dependente do especialista e da forma como este codificava para proceder à manutenção.

Foi então adotada a técnica de coleta de informações com a criação de um pequeno grupo voluntário cujos integrantes usavam os recursos de informática preferidos, para obter materiais, para serem apresentados em reuniões de trabalho e palestras, com abordagens de assuntos sobre qualidade e engenharia de software, com disponibilidade para colher sugestões em relação a soluções de problemas, basicamente de maneira formal, que possibilitou transformar os processos informais em documentações. Assim foi possível melhorar a familiaridade da equipe com os modelos de processos existentes, e direcionar os esforços dos especialistas do TA-2, para a obtenção de um entendimento sobre os processos de desenvolvimento de software e normas aplicáveis.

Também se considerou importante na METAPLI a implantação de gestão de conhecimento para preservação do capital intelectual, caracterizando um risco caso não houvesse investimentos no aumento das capacidades individuais, conhecimento, habilidades e experiência dos especialistas em ensaios em túnel de vento. Então se envidaram esforços para incentivar os integrantes a obter o conhecimento através de acesso livre à Internet, publicações de trabalhos técnicos, cursos, congressos e afins.

Com o plano de avaliações houve a conscientização dos especialistas do TA-2 sobre a importância da implantação da METAPLI e foi possível levantar o estado do sistema de software anterior. Com a identificação dos problemas se formulou um conjunto de melhorias e aperfeiçoamentos, tendo como meta o uso de modelo de processo adequado ao desenvolvimento de software e a inclusão de novas atividades de engenharia de software, para se desenvolver o novo projeto. Foi dada ênfase na decisão quanto ao não uso do mesmo modelo de processo de desenvolvimento anterior para o novo projeto, informando os seus impactos com relação aos erros de software ou de dados.

Com a análise das tarefas dos usuários aplicadas na manutenção e operação dos softwares existentes para plataforma HP, foi possível elaborar uma distribuição dos esforços de trabalho para a equipe de engenharia de software. Também foi possível considerar parcerias para auxiliar no trabalho, dentro das atividades envolvidas de engenharia de software, tanto no projeto dos novos softwares quanto nos softwares existentes, até a sua desativação por completo.

Portanto, a ação via práticas da METAPLI para um modelo de processo de software mais eficaz, ocorreria com a transição das atividades informais para as ações das metodologias tradicionais e ágeis, que implicou em uma mudança de cultura bastante complexa dos envolvidos no projeto de modernização.

Ainda como parte do plano de avaliação, Sommerville (2003) recomenda avaliar um sistema de software sob duas perspectivas: o valor do negócio ou valor estratégico, e a qualidade do sistema. Como já mencionado anteriormente, o sistema de software do TA-2 vem prestando uma importante contribuição aos ensaios aerodinâmicos, que são os serviços fornecidos pelo túnel de vento a seus clientes.

Sob a perspectiva do valor estratégico, este sistema de software é de suma importância, ou seja, apresenta um alto valor. Quanto à qualidade do sistema de software existente, esta foi considerada baixa, devido à ausência de um modelo de

processo, aos erros introduzidos na manutenção e operação e outros elementos. No entanto, os softwares de aquisição e redução devem ser implantados. Então, o resultado apresentado da avaliação sob as perspectivas segundo este autor (SOMMERVILLE, 2003), sugere que o sistema de software seja transformado ou substituído.

Diante dos problemas citados com a manutenção dos softwares de aquisição e de redução de dados, foi preciso decidir sobre o que implementar. Esta decisão foi tomada entre as duas propostas de solução, ou seja, transformar ou substituir os softwares por novos, considerando a opinião dos especialistas quanto à modernização do sistema de medidas e à importância dos valores dos softwares, optou-se pela reengenharia para transformar o software de redução de dados, e pela substituição por um novo software de aquisição de dados com enfoque no reuso de possíveis partes dos subprodutos desenvolvidos, durante implantação do modelo de processo escolhido.

Os processos do sistema de software foram definidos através das ações da METAPLI, também na prática de planejamento. No desenvolvimento destes sistemas de software se adotaram estratégias de uma metodologia iterativa e também de uma estruturada. A primeira para o desenvolvimento do software de aquisição de dados e a segunda para o software de redução de dados.

4.3.1 Ações do processo ágil

Para desenvolver o software de aquisição de dados do TA-2, a estratégia iterativa é aplicada com padrão de iteração de ciclo de vida evolutivo. Utilizaram-se então as ações da metodologia *Rational Unified Process -RUP*, que neste trabalho está sendo considerada como uma metodologia ágil. Dentre estas ações estão as estratégias de processo que abrangem a determinação dos requisitos, análise, projeto ou *design*, implementação e teste (RATIONAL CORPORATION, 2000).

Com um sistema de software desse porte, definir antecipadamente todos os requisitos não é uma tarefa fácil, portanto, a estratégia adotada foi aumentar a visibilidade do projeto do sistema de medidas progressivamente, que implica em obter subprodutos que vão sendo refinados a cada ciclo de liberações parciais sucessivos. Dessa forma, as ações do processo unificado, para desenvolver o software de aquisição de dados iterativamente, seriam adequadas. Como parte do planejamento do desenvolvimento do sistema de software, o processo na prática possui a estrutura representada em forma gráfica, conforme mostra a figura 4.3.

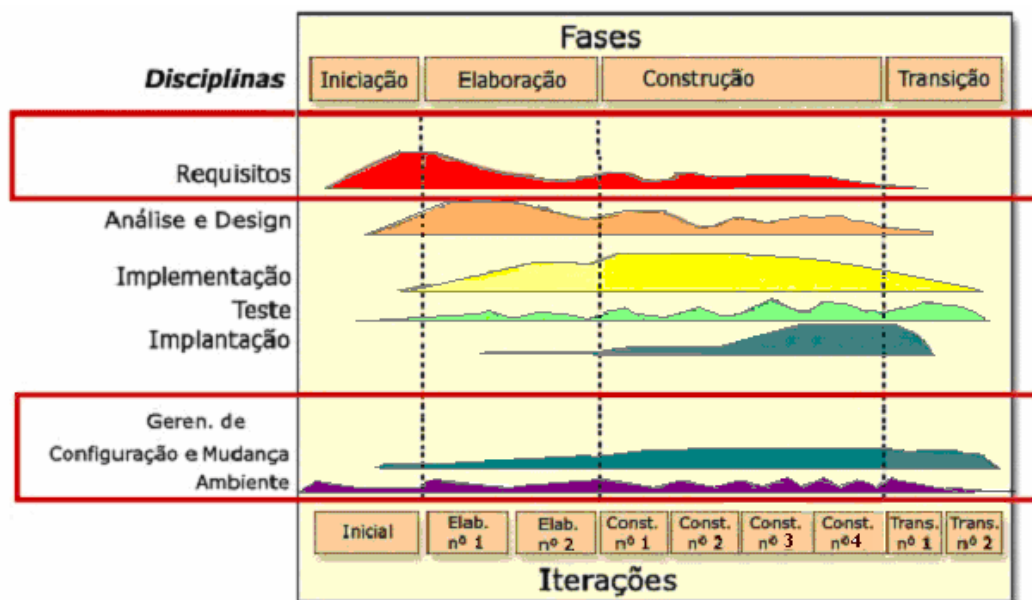


Figura 4.3 - Ciclo de vida de desenvolvimento de software com ações do RUP.

Este processo é dividido em quatro fases seqüenciais que são:

- **Iniciação:** nesta fase define-se o escopo do projeto, as atividades e os responsáveis por estas atividades no sistema, e alguns documentos para auxílio à gerência de um projeto.
- **Elaboração:** nesta fase as atividades são desenvolvidas voltadas ao domínio do problema e à arquitetura do projeto do sistema de software, dando ênfase

a um plano de desenvolvimento detalhado. É considerada a fase mais crítica de um processo unificado.

- Construção: é a fase na qual o produto ou os subprodutos de software são obtidos e toda a documentação do projeto do sistema de software deve estar revisada, para que os produtos ou subprodutos possam ser entregues ao usuário final, com uma documentação representativa. É considerada a fase principal do processo unificado.
- Transição: nesta fase a versão final do sistema de software é entregue ao usuário, sendo que, para validá-lo, adota-se a estratégia de entregar uma versão preliminar para o usuário, e proceder à identificação de problemas, que devem ser todos corrigidos nesta mesma fase. Deve haver ainda uma avaliação final, na qual, em função dos resultados obtidos, pode-se iniciar um novo ciclo de desenvolvimento, devido à necessidade de se melhorar o sistema de software obtido.

O processo unificado consiste em incorporar as várias atividades ou disciplinas descritas em várias iterações. Nesta mesma figura, em sua parte inferior, observam-se as iterações de cada fase segundo a estratégia evolutiva. A cada iteração no desenvolvimento do software do TA-2 passa-se pelas atividades previstas ou disciplinas do modelo de processo, resultando na entrega de um produto de software executável de aquisição de dados, subconjunto do produto final em desenvolvimento, que cresce por incrementos, para se tornar o sistema de medidas final.

O RUP é caracterizado como um processo que pode ser adaptado para as necessidades da organização. No caso do software de aquisição de dados do TA-2, algumas disciplinas não eram necessárias, e as iterações foram definidas conforme a extensão necessária para a conclusão dos subprodutos de software, de modo que os envolvidos utilizassem um sistema de medidas mais simples, até a evolução para o mais complexo.

Para uma organização com enfoque em pesquisa e desenvolvimento, a modelagem do negócio é um pouco diferente de uma empresa comercial. Apesar de se tratar de um projeto de modernização complexo, com similaridades aos desenvolvidos em uma indústria, os serviços de ensaios em túnel constituem o único negócio do TA-2, não existindo outro produto. Portanto, a disciplina de modelagem do negócio não foi considerada com a abrangência que as práticas do processo unificado estabelece, por se tratar de uma organização governamental de pesquisa, sem focar a disputa de mercado e a obtenção de lucros.

Isto não significa que as práticas para o desenvolvimento dos softwares aplicadas neste projeto de modernização não possam ser praticadas por indústrias que comercializam produtos de software. Apenas significa que, neste trabalho, não se exploraram intensamente as práticas de modelagem do negócio devido ao tipo de organização. Também não foram consideradas em sua total abrangência as atividades de gerenciamento de projeto como práticas gerenciais deste trabalho. No projeto de modernização do TA-2, estas atividades de gerenciamento de projeto foram executadas pela empresa solicitante do financiamento, não cabendo, neste caso, aos integrantes do TA-2.

As iterações necessárias para cada fase do processo unificado foram decididas pelos envolvidos, com base nas necessidades de alguns ensaios pré-programados, e descritas no plano de desenvolvimento. Estas iterações são descritas na tabela 4.2.

Tabela 4.2 - Iterações de cada fase do processo unificado.

FASE	Nº DE ITERAÇÕES
Iniciação	1
Elaboração	2
Construção	4
Transição	2

Os tempos das iterações não apresentaram durações definidas, conforme defendido pelos integrantes do movimento ágil (AGILE MANIFESTO, 2001). Cabe lembrar que as metodologias ágeis são recentes e ainda não houve o desenvolvimento de um software grande e complexo, com o uso destas. Assim, não foi possível definir os tempos das iterações, mas se estabeleceram os critérios de avaliação, para a que cada iteração fosse incrementada.

4.3.1.1 Ações para a fase de iniciação

Seguindo a METAPLI na prática de planejamento, as ações para a fase de iniciação na abordagem iterativa do processo unificado devem ser executadas. Essas ações tratam do levantamento das atividades com relação aos aspectos de projeto de sistema de software. Dentre estes aspectos, tem-se: as ferramentas ou aplicativos comerciais utilizáveis, os softwares desenvolvidos internamente, aqueles adquiridos prontos para serem usados, as habilidades das pessoas, os softwares de prateleira ou COTS, entre outros.

Não somente na fase de iniciação, mas também nas demais fases, todas as iterações são planejadas e controladas, e as atividades exercidas que visam a verificação dos aspectos de projeto de sistema de software, permitem uma integração de software quase contínua a cada iteração. Na fase de iniciação para a qual foi prevista uma única iteração, foram definidos modelos aplicáveis ao desenvolvimento do sistema de software, os quais foram sendo complementados nas fases seguintes.

4.3.1.2 Ações para a fase de elaboração

As ações desta fase são praticadas após se obter o entendimento sobre o sistema de medidas do TA-2 e seus subsistemas. As descrições do sistema de software em detalhes são obtidas na fase de elaboração, uma vez que, nesta fase se definem os elementos de hardware e dispositivos eletrônicos e suas interligações com o

computador pessoal, e também os componentes dos softwares para a aquisição e redução de dados. Também são descritas as aplicações de reuso e de reengenharia de software e suas respectivas abordagens.

A intenção em considerar duas iterações nesta etapa é diminuir o risco, considerado alto pelos especialistas do TA-2, em relação às capacidades dos desenvolvedores com respeito à ferramenta de apoio ao desenvolvimento *LabView*, e aos recursos computacionais voltados para aquisição de dados da fabricante *National Instruments*.

As atividades desempenhadas nas fases de iniciação e de elaboração estão voltadas para o domínio do problema que é o sistema de medidas. Cabe então, na fase de elaboração um detalhamento do projeto do sistema de medidas, definindo um refinamento para o plano de desenvolvimento dos softwares de aquisição e de redução de dados.

Nesta etapa são definidos os requisitos dos subprodutos, sendo que o primeiro deles a ser desenvolvido, tinha como objetivo treinar os desenvolvedores para a familiaridade com a ferramenta de apoio ao desenvolvimento de software *LabView*. Também nesta etapa os requisitos gerais para o sistema de medidas e de interface homem máquina foram detalhados, os quais devem ser comprovados pelos subprodutos de softwares de aquisição de dados. A técnica de treinamento é conhecida como “*on the job training*” e consiste na definição detalhada das atividades e operacionalidades que o subproduto de software usará e sua implementação pelos desenvolvedores sempre em parceria.

Durante a fase de Elaboração, a ênfase recai em definir o sistema e refinar a definição do sistema de software. Nesta fase foram planejadas duas iterações, sendo que em cada uma delas serão detalhadas ainda mais as operacionalidades e atividades.

Independente da fase, as gestões do escopo do sistema e dos requisitos variáveis são detalhamentos feitos continuamente ao longo do projeto de software. Segundo ações do processo RUP (RATIONAL CORPORATION, 2002), o fluxo de trabalho apresentado na figura 4.4, a seguir, ilustra a aplicação de partes da fase de iniciação e da fase de elaboração, do novo projeto de sistema de software do TA-2, em similaridade com a proposta geral do RUP.

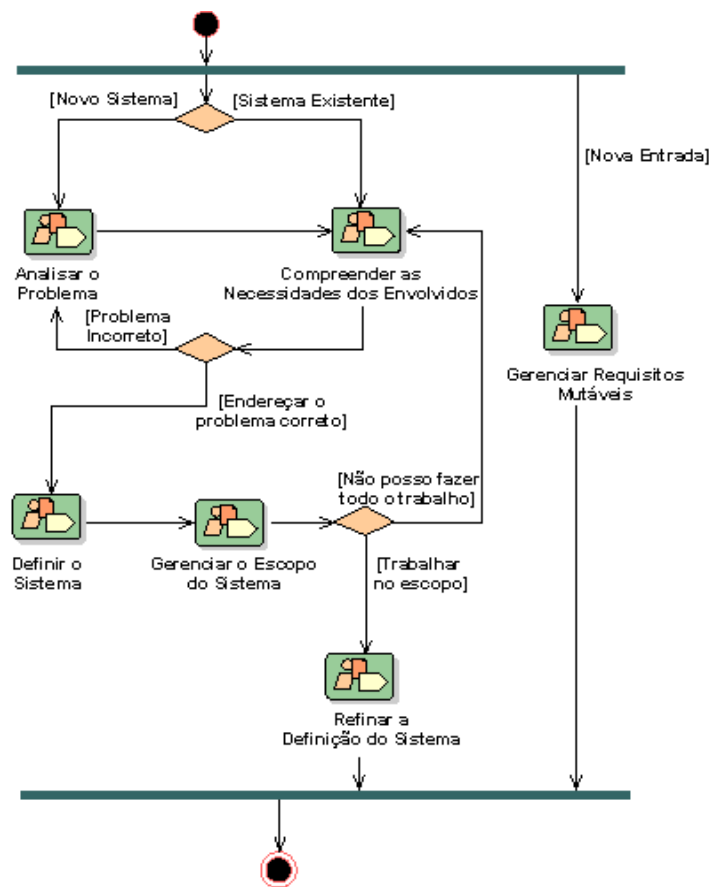


Figura 4.4 - Fluxo de trabalho RUP do projeto de sistema de software do TA-2

O que difere as ações do novo projeto de sistema de software do TA-2 da proposta geral do RUP são as atividades e artefatos a serem adotados para este projeto específico. As ações do processo unificado RUP foram delineadas para organizações em que os processos de engenharia de software estão sistematizados. No caso deste trabalho, os processos de engenharia de software estão sendo ajustados para este projeto de modernização, o qual é considerado complexo devido ao grau de inovações, o que implica na necessidade de atividades e artefatos específicos.

4.3.1.3 Ações para a fase de construção

Na fase de construção as ações correspondem a múltiplos ciclos de análise e projeto, implementação e teste de subprodutos de softwares e do produto final, definidos de acordo com as necessidades dos clientes de ensaios em túnel de vento. Inicia-se a evolução do sistema de software pelo subproduto para calibração de células de carga no laboratório de força. Os requisitos de software foram parcialmente definidos, e as atividades de engenharia para tornar automatizado o processo de calibração foram planejadas e organizadas, proporcionando um treinamento com respeito ao ambiente de desenvolvimento adotado.

As ações desta fase correspondem também à diagramação para visualizar, implementar e testar os subprodutos de softwares que foram definidos. Estes subprodutos correspondem aos construídos em cada iteração para operacionalizar os ensaios em túnel. Neste trabalho foram definidos subprodutos para medição via sensores e para controle de eventos durante os ensaios aerodinâmicos, visando a aquisição de dados dos sensores instalados na seção de ensaios e monitoramento dos eventos importantes dos ensaios, para posterior processamentos pelo software de redução de dados. Também são partes destas construções os subprodutos experimentais que tratam os dados de ensaios, para melhoria do processamento, e que facilitam a obtenção de resultados precisos e corretos.

4.3.1.4 Ações para a fase de transição

Para fase de transição as ações são voltadas para aprovação do produto final de software também chamada de implantação. Nesta fase as ações são voltadas para a implantação, testes de aceitação, instalação e operação do sistema de medidas final integrado, submetidos à aprovação dos clientes. São previstas duas iterações devido à primeira iteração se tratar da integração com aprovação dos especialistas em ensaios em túnel, considerados os clientes internos. A segunda iteração envolve a aprovação

dos clientes externos, ou seja, aqueles que contratam os serviços de ensaios em túnel de vento para seus projetos aerodinâmicos.

4.3.2 Ações do processo tradicional

Baseado na literatura e nas coletas de informações nas indústrias do ramo aeroespacial, com respeito às práticas das metodologias tradicionais e ágeis, notou-se o uso da norma DOD-STD-2167A (1988), com as adequações necessárias. Para o projeto de software do TA-2, não se estabeleceu, a princípio, uma rigorosidade quanto ao uso da norma, apenas a recomendação que o conteúdo estabelecido em cada documento seria de alguma forma abordado, e os títulos dos documentos seguiriam este padrão.

Esta norma é mais adequada para ações do processo tradicional (MAIBOR, ARDERSON, DORFMAN, 1991) e a escolha se deve aos relatos de experiências por outros especialistas do CTA. Um exemplo é o uso da DOD-STD-2167A (1988) no desenvolvimento do software do VLS (LEMES, 1997), no qual se utilizou o modelo de processo em cascata juntamente com o processo de prototipação (PASTORELLI, 1998). O uso desta norma também foi feito pela EMBRAER no desenvolvimento de seus softwares. A diferença de documentos entre esta norma, em relação a suas evoluções ou normas anteriores, também faz parte do Apêndice A, deste trabalho. Os documentos gerados no desenvolvimento dos softwares são citados no decorrer deste trabalho.

Neste trabalho, também se considera a ação do modelo cascata quando se referencia as revisões do projeto do sistema de software. Os marcos de revisão são estabelecidos no processo de gerenciamento de configuração, em que são tratadas as aprovações ou não dos documentos de requisitos, projetos e testes, bem como das versões dos subprodutos. O modelo em cascata provê uma estrutura comum que pode ser vista como parte do processo unificado, e na representação deste projeto abrange as

atividades de requisitos, análise e projeto, implementação e teste. Ao complementar cada atividade se obtém os documentos que serão submetidos à revisão formal, relacionada à norma MIL-STD-1521B (1985).

O modelo em cascata é baseado em suposições oriundas de processos convencionais, sendo que neste trabalho, o enfoque estruturado foi considerado como uma ação dos processos tradicionais, e foi usado para modelagem detalhada do sistema de software. Estas modelagens refletem o entendimento do sistema de software, e contribuem para a identificação dos controles e dados a serem processados. Essas representações do sistema de software são refletidas no uso de diagramas de contexto e diagramas de fluxo de dados.

4.4 PLANEJAMENTO DAS AÇÕES GERENCIAIS

Devido às restrições gerenciais e técnicas originárias do processo implantado no sistema de medidas existente, foi adotada a estratégia de abordar na METAPLI as transformações realizadas por pessoas. Foram então incluídas as atividades consideradas importantes na representação de um projeto de software, que abrange a aceitação dos processos, ferramentas de apoio ao desenvolvimento, dispositivos eletrônicos e, até mesmo, o reconhecimento dos problemas que poderiam gerar novas restrições ao novo projeto. Um resumo das práticas da METAPLI para o planejamento da fase de iniciação do processo unificado, em relação aos aspectos gerenciais, consiste no seguinte:

- Convencer a equipe a aceitar e usar novos processos, tecnologias e ferramentas de software.
- Analisar e dirimir os problemas do processo de desenvolvimento de software anterior.
- Compreender as necessidades dos envolvidos.

Os problemas devido ao uso de um processo informal foram levantados durante o acompanhamento de ensaios, com o modelo aerodinâmico de um novo avião da EMBRAER. Houve avaliação de todos os subsistemas envolvidos com o uso da plataforma HP, integrados aos demais equipamentos e dispositivos de medidas e de controle. A metodologia implementada concentrou-se em listar os problemas das áreas mais críticas e importantes do sistema de medidas anterior, para encontrar soluções adequadas no processo de desenvolvimento do novo sistema.

Os estudos dos programas em BASIC existentes deram início às atividades de modernização do sistema de medidas do TA-2, nos quais foram cuidadosamente identificados os problemas gerenciais e técnicos, através de inspeções de código, análise de documentação, testes funcionais dos dados principais e inspeção visual detalhada do sistema de medidas e dispositivos. Foram então levantados os casos de problemas ou discrepâncias daquele desenvolvimento, para que estes problemas fossem dirimidos no desenvolvimento dos novos softwares. A lista obtida com respeito aos problemas é:

- Gerenciamento de requisitos insuficiente.
- Arquitetura de software e hardware deficientes.
- Inconsistências não detectadas em requisitos, projetos e implementações.
- Violação de padrões de programação.
- Número insuficiente de testes para detecção de erros no sistema.
- Execução de mudanças descontroladas.
- Integração de programas deficiente.
- Comunicação ambígua e imprecisa.
- Número insuficiente de pessoas para desenvolvimento.
- Configurações do sistema ou subsistema não registradas.

Estes problemas foram alocados, inicialmente, na lista de risco, a qual era discutida em cada reunião com os envolvidos procurando tornar visível para todos integrantes, qual a influência de cada fator, estimando o quanto cada elemento é

responsável por problemas no desenvolvimento de software. Esta técnica para apontar problemas deixa claros os objetivos de torná-los sob controle durante o processo de desenvolvimento do novo sistema de software.

Para cada item apontado na lista de risco foram computados alguns pesos, e efetuada uma classificação em grave, moderado e menor, numa avaliação dos processos com a plataforma HP e dos softwares existentes. Assim sendo, para o novo projeto de software todos os problemas foram trabalhados pela equipe, para dirimir, o quanto antes, a probabilidade de sua ocorrência. As práticas na implantação da METAPLI para dirimir cada um dos problemas apontados, são voltadas para os meios de verificação da qualidade e de gerenciamento de configuração, aplicados durante o ciclo de vida do projeto do sistema de software.

Numa avaliação do sistema de medidas anterior se detectou que este projeto teria sido desenvolvido baseado em técnica orientada a funções. Sendo que a transformação dos processos de engenharia em um projeto orientado a objetos, dependeria de uma rígida mudança cultural na codificação dos softwares. Então se fez análises de ferramentas CASE disponíveis comercialmente, para que estas servissem de apoio nesta transformação de processos. Porém, devido a uma restrição de orçamento para aquisição e uso da ferramenta LabView, que na época de início deste trabalho, ainda não disponibilizava comercialmente processos orientados a objetos a um custo viável, a utilização de ferramentas CASE para o desenvolvimento e a documentação do sistema de software se tornou inviável.

Outro fator, não menos importante, também inviabilizou o uso de ferramentas CASE. Trata-se do recurso de linguagem padrão de modelagem, denominado *Unified Modelling Language- UML*, que é a principal ferramenta para programação orientada a objetos. Este recurso das ferramentas CASE disponíveis comercialmente, não provê geração de códigos em LabView, portanto não seria viável descrever e visualizar modelos e artefatos com esta linguagem padrão de diagramação, para modelar objetos que não poderiam ser transformados em códigos do ambiente de programação visual

estabelecido no projeto. Além de fatores como o custo elevado e de seu uso requerer um domínio de conhecimento de metodologias, modelos de processo e normas para modelar, representar e documentar corretamente o projeto de engenharia de software.

Os processos de engenharia analisados durante os ensaios com os softwares existentes foram os principais meios usados, para iniciar uma abordagem gerencial no novo sistema de software, a qual servira de base para melhorias do novo projeto de sistema de software. Essa inter-relação entre as atividades de engenharia de software e de qualidade foi melhor abordada como parte dos processos gerenciais, os quais, as práticas de implantação da METAPLI, apresentam como uma modelagem de forma diagramática.

A aplicação prática do planejamento via METAPLI teve como base as diversas avaliações sobre os processos dos softwares da plataforma HP, e as ações dos processos ágeis e tradicionais envolvendo os planos das atividades de modelagem, detalhadas para o projeto do sistema de software. Assim, a METAPLI foi gerada e implantada para prover a transição suave de um processo informal para um processo formal ou sistemático. A transição dos processos ocorreu, de modo que, para o projeto do software de aquisição de dados se implantariam as atividades com os enfoques incremental e iterativo com ações do processo unificado para sua representação, e se aplicariam as atividades de reuso.

Para o projeto do software de redução de dados se aplicariam as atividades de reengenharia, e seriam considerados as representações do sistema de software com enfoque estruturado como ações dos processos tradicionais, usando modelo de processo em cascata. A reengenharia usando ações do processo cascata permitiria que boa parte da especificação do sistema de software fosse completada e bem sucedida. Em termos gerenciais para os projetos do software de aquisição e redução de dados seriam consideradas as revisões formais, representações comuns de atividades gerenciais de apoio ao desenvolvimento e a norma estabelecida para documentação.

No projeto de modernização do TA-2, adotou-se o ciclo de vida iterativo devido à possibilidade de listar, o quanto antes, os riscos envolvidos que podem afetar todo o desenvolvimento do sistema de software, abordagem esta que é similar à do modelo de processo em espiral (BOEHN, 1988). Portanto, se difere do ciclo de vida em cascata (PRESSMAN, 1992), pois numa abordagem iterativa, cada iteração permite a produção de um software executável e a identificação dos riscos, o que com o ciclo em cascata só é possível na integração do software ao final do ciclo, conforme pode ser visto na figura 4.5.

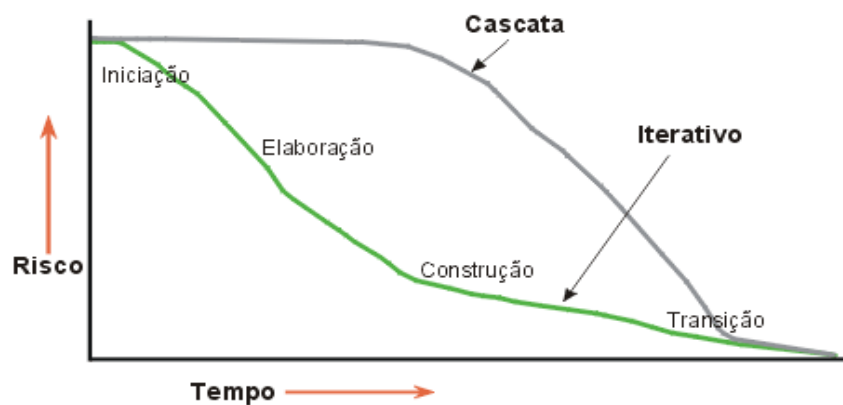


Figura 4.5 - Comparação dos riscos entre o ciclo iterativo e em cascata (KRUCHTEN, 2000).

Mesmo assim, uma equipe de desenvolvimento experiente não consegue prever todos os riscos, alguns deles não sendo descobertos até que se proceda a integração do sistema. No entanto, a vantagem da abordagem iterativa é que os riscos são reduzidos, na medida em que a integração ocorre progressivamente e, portanto, mais cedo neste ciclo.

A cada iteração desde a fase de iniciação, uma análise de riscos no projeto de desenvolvimento do sistema de medidas do TA-2 é efetuada. A lista de riscos é obtida e são verificados em reuniões com todos os envolvidos, e delineadas as ações para dirimir cada risco identificado.

A reunião para avaliação de risco se inicia com uma análise qualitativa para definir o grau do risco, classificando-os como, por exemplo, em alto, médio ou baixo, e caso seja necessário, complementa-se com a análise quantitativa por meio de

técnicas de demonstração de parte do subproduto ou da chamada prototipação. Os riscos são priorizados de acordo com seu efeito potencial no projeto de modernização do sistema de medidas.

Para abordar os processos gerenciais também foram usados os Diagramas da Tartaruga adaptados da área de produção, usados em auditorias internas (AFS, 2005), nos quais se podem definir os aspectos voltados para qualidade de software, envolvidos nos processos de desenvolvimento dos softwares de aquisição e redução de dados. Através desses diagramas foram definidas as modelagens dos softwares em relação à qualidade e à gestão, incluindo uma visão sistemática para a qualidade de software. Um modelo geral do Diagrama da Tartaruga é mostrado na figura 4.6.

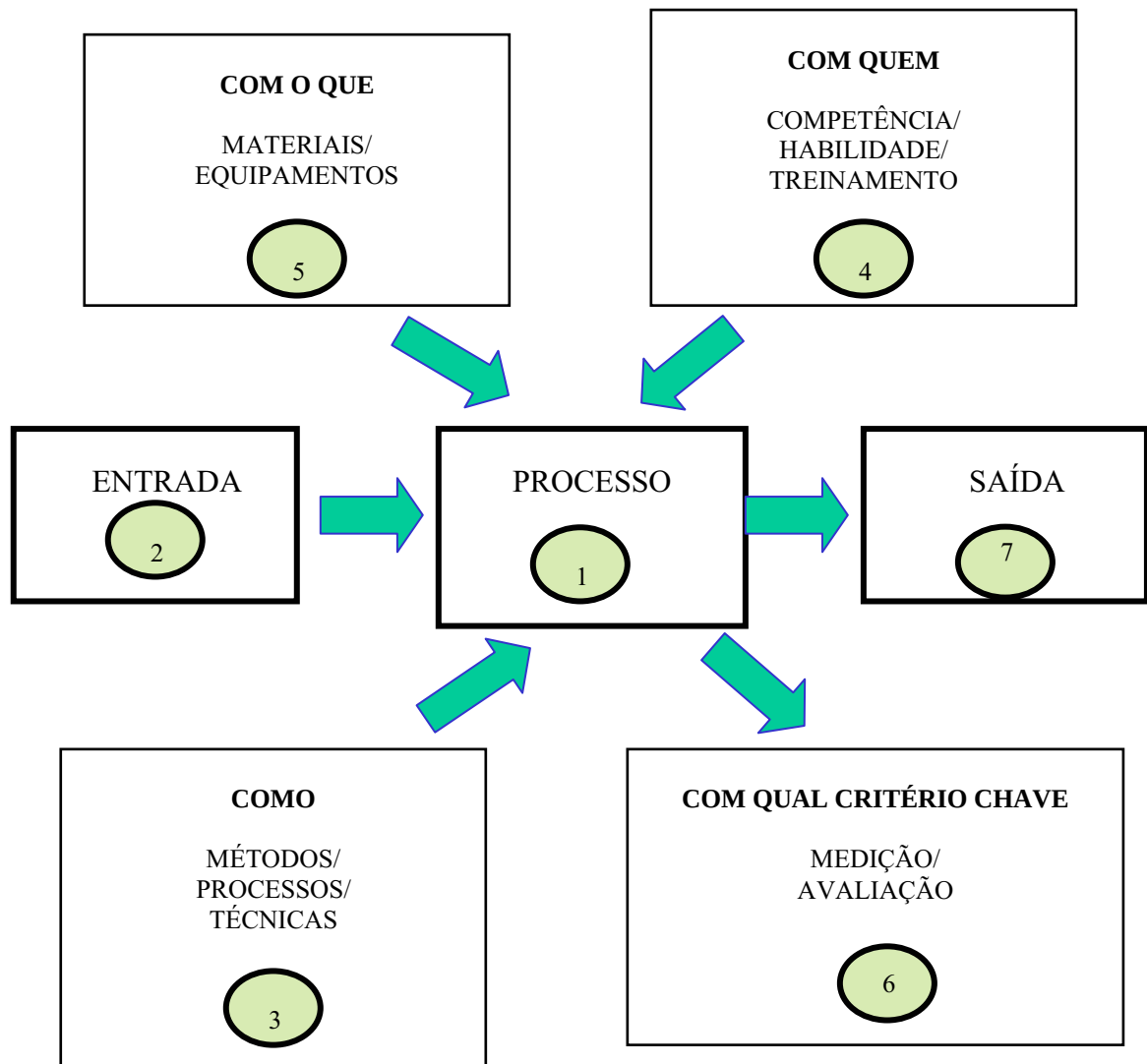


Figura 4.6 - Modelo geral do diagrama da tartaruga (AFS, 2005)

Neste trabalho, o diagrama da tartaruga foi adotado para a definição dos processos de realização dos novos softwares, visando estabelecer itens que direcionem os processos, para que as entradas requeridas sejam conduzidas para obtenção das saídas estabelecidas em cada processo. O uso dos diagramas neste trabalho surgiu de consulta na Internet, com respeito a normas ISO aplicadas em indústrias automobilísticas, com destaque para a ISO/TS 16949:2002 (AFS, 2005).

Em linhas gerais esses diagramas contemplam os aspectos dos processos de desenvolvimento e de qualidade de cada software, tais como, a definição do processo de software a ser desenvolvido, detalhes das informações de entrada e saída do processo, “com que” materiais e equipamentos, “com quem” em termos de pessoas, habilidades, competências e treinamento, “como” em relação aos métodos, processos e técnicas e “com qual critério chave” para medição e avaliação.

Durante as atividades do processo de desenvolvimento do projeto de modernização do sistema de medidas do TA-2, usando as ações do processo unificado e do processo em cascata, os elementos considerados principais foram as disciplinas, os papéis ou responsabilidades, as atividades, os artefatos, os diagramas de fluxo de dados, os fluxogramas e os fluxos de trabalho. Os detalhes de alguns aspectos gerenciais fazem parte do fluxo de trabalho, diagramas de tartaruga e são mostrados em uma ordem lógica e seqüencial, em que seriam aplicados no novo projeto de sistema de software.

4.5 PLANEJAMENTO DAS AÇÕES TÉCNICAS

O plano de desenvolvimento de software para estabelecer o novo sistema de medidas foi definido atendendo a um ciclo de vida de software com todas as atividades, desde a idéia do que se quer desenvolver até a entrega ao cliente, e também pós entrega. O ciclo de vida do desenvolvimento de software segundo o processo unificado de estratégia evolutiva, é apropriado ao software de aquisição de dados

porque os requisitos do software podem ser determinados a partir do domínio do conhecimento sobre o sistema de medidas do TA-2.

Esse domínio de conhecimento pode ser obtido à medida que se evolui no ciclo de desenvolvimento do sistema de software. Abrange o conhecimento acumulado e o conhecimento de interesse da METAPLI, e suporta requisitos parcialmente definidos para produzir resultados de cada subproduto de software. Estes conhecimentos podem ser obtidos em documentos gerados em cada fase do ciclo de vida e com a implantação de novas tecnologias. As ações técnicas para implantar tecnologias inovadoras são caracterizadas por novos equipamentos ou dispositivos eletrônicos, instrumentações requeridas para a campanha ou programa de ensaio em túnel e a elaboração de softwares associados para operação do sistema de medidas.

Como exemplo de inovação técnica citam-se as substituições das conexões HP-IB dos instrumentos de medidas, com as quais se acionavam funções de até 15 dispositivos, e que eram executadas pelos programas em BASIC para comunicação com os equipamentos da HP (multímetro, scanner, impressora) usados pelo sistema existente. Esta mesma função foi implementada usando reuso de funções de bibliotecas da ferramenta LabView, disponibilizadas no mercado através de *drivers*. Esta ferramenta de programação visual provê também outras bibliotecas com funções apropriadas para a aquisição de dados de instrumentações.

Também com as informações referentes aos ambientes de apoio ao desenvolvimento e de programação visual, foi possível interligar dispositivos da *National Instruments* ao computador pessoal, e prover via software uma interface gráfica e textual para apresentação dos dados e resultados dos ensaios aerodinâmicos.

O software de aquisição de dados é um dos elementos chave dos ensaios aerodinâmicos, sendo responsável por administrar todas as atividades de ensaio e calibração do TA-2, além de proceder à realização de cálculos, monitoração e controle dos instrumentos pelo computador. Com a implementação de funcionalidades

similares com uso de equipamentos e software na plataforma PC, é possível adquirir dados via comunicação digital, o que requer um interfaceamento adequado para a transmissão e recepção dos sinais digitais.

Um instrumento real genérico é dotado de componentes como sensor ou atuador, transdutor, painel de controle e medição e painel de conexões. Esses não são os únicos componentes de um instrumento real, podendo existir outros como os circuitos eletrônicos. Como as funcionalidades desses instrumentos reais do TA-2 permaneceriam no novo projeto de software, se estudou um conjunto de funções adequadas para a substituição dos instrumentos da plataforma HP. Esta instrumentação consiste de um sistema de controle ou medida formado por um computador pessoal acrescido das funções (controle ou medida) providas por softwares, e de dispositivos eletrônicos colocados em comunicação conforme executados pelos instrumentos.

Na modernização do software de aquisição de dados com uso do *Labview*, todas as funções de medidas e controle seriam implementadas via software como uma instrumentação virtual. O computador pessoal seria usado tanto para operar o instrumento quanto para conduzir o experimento. Interfaces seriam desenvolvidas no software de aquisição de dados, para prover ao operador a mesma funcionalidade que um instrumento real apresentaria ao usuário em seus painéis.

Contudo, haveria a necessidade de pessoas especializadas em desenvolver softwares usando *LabView*, ou ampliar a equipe do TA-2 que era pequena contando com apenas 12 pessoas na época, e apenas dois especialistas na área de engenharia de software. Então, o próximo passo da metodologia foi um levantamento de possíveis equipes desenvolvedoras de software, com certa experiência em engenharia de software e garantia da qualidade, para desenvolver os softwares do novo sistema de medidas em parceria, usando as ações da METAPLI.

Para avaliar cada equipe a ser escolhida, e trabalhar com pessoas envolvidas no projeto de maneira eficaz, utilizou-se a técnica de etnografia (SOMMERVILLE, 2003)

que consiste em observar o ambiente de trabalho onde o sistema será inserido e anotar as tarefas reais em que os participantes serão envolvidos. Assim, foi possível derivar as informações da maneira como as pessoas realmente trabalham, revelando importantes detalhes do processo que poderiam ser trabalhados em conjunto, até mesmo em locais distantes, porém com a gestão comum dos subprodutos de software requeridos.

Com a escolha das parcerias e com o levantamento das práticas, técnicas, processos e métodos usados para o sistema de software antigo, foi possível selecionar os itens de engenharia e gerenciais aplicáveis para o desenvolvimento e a documentação dos softwares em seus ambientes operacionais, considerando os aplicativos e dispositivos eletrônicos adquiridos ou indicados pelas análises realizadas e demonstrados aos especialistas do TA-2, começando por documentar os materiais do sistema de medidas antigo.

Com base nas informações coletadas foram estabelecidas: as inovações no processo de desenvolvimento de software, as ferramentas ou aplicativos de apoio ao desenvolvimento, os dispositivos de medidas e conseqüentemente as novas ações a serem implantadas na prática via METAPLI. Esta metodologia especial tornou-se uma necessidade para a melhoria da qualidade do sistema de medidas renovado e reequipado.

A metodologia foi elaborada para abranger os aspectos considerados importantes para a engenharia de software, agregando práticas com um enfoque voltado para atualizações futuras. O passo inicial foi convencer a equipe de que tais práticas, juntamente com modelos de processo e demais atividades, poderiam gerar um sistema de medidas novo que fosse bem sucedido e documentado, podendo facilitar o entendimento dos fenômenos físicos ocorridos durante o ensaio em túnel, e relatar aos clientes dados e resultados confiáveis.

Os dispositivos eletrônicos e softwares envolvidos no sistema de medidas existente foram levantados e relacionados ainda na coleta de informações. Trata-se daqueles mais relevantes para o software de aquisição e redução de dados, substituídos ou não. O levantamento dos elementos do sistema existente foi feito para melhorar o entendimento das funcionalidades implementadas pelos softwares em BASIC, principalmente para o caso da reengenharia.

Também foi possível documentar os novos processos de engenharia e gerenciais, abordando as condições operacionais identificadas do sistema de medidas e de seus subsistemas, usando a modelagem diagramática. Assim, através da aplicação das técnicas descritas em planos, que serão seguidas durante o ciclo de vida, é possível quantificar o risco tanto gerencial quanto técnico, além de auxiliar na correta descrição de funcionamento das partes do subproduto de software e dos subsistemas envolvidos. Numa avaliação da usabilidade do subproduto, pode surgir uma inclusão de partes demonstradas como reuso, contribuindo assim com sucesso do desenvolvimento do software.

4.5.1 Entregas e aceitações freqüentes

Na implantação da METAPLI, são incluídas atividades nos planos de avaliações e de desenvolvimento, referentes aos processos para entregas e avaliações dos softwares. Entre estes processos estão as atividades gerenciais e técnicas para exercer a avaliação e aceitação de software, assim que considerados operacionalmente viáveis. Diminuindo assim, os riscos de obter softwares incorretos e incompletos quando da entrega do produto de software final.

Segundo as metodologias tradicionais estas entregas e aceitações acontecem somente no final do ciclo de vida do desenvolvimento, mais especificamente na etapa de teste onde a identificação dos problemas pode ser tardia. Procurou-se adotar a estratégia de identificar os problemas com antecedência, que as metodologias ágeis

provêm com a abordagem incremental e iterativa, minimizando a intensidade de sua ocorrência. Com isso, procurou-se garantir um alto nível de qualidade no desenvolvimento do projeto do novo sistema de medidas, que abrange a evolução eletrônica e os softwares.

Com ações da METAPLI podem ser citados diversos documentos em que são definidos processos e atividades gerenciais e técnicas, para prover a qualidade dos subprodutos a serem entregues e aceitos. Dentre estes documentos podemos citar: o Plano de Desenvolvimento, o Plano de Garantia da Qualidade, o Plano de Gerenciamento de Configuração e o Plano de Teste, e os documentos para os processos de requisitos, análise e projetos, implementação e testes estabelecidos durante o desenvolvimento.

Uma vez definidas como serão as práticas do desenvolvimento, todo o sistema de medidas, seus subsistemas e atividades de engenharia do TA-2 serão detalhados nos tópicos seguintes deste trabalho. Também em capítulos posteriores estão as abordagens dos planos, para os quais, a cada iteração do processo unificado e fases do processo cascata, as documentações serão atualizadas com as evoluções e/ou serão procedidas as revisões pertinentes.

Quando se trata de software as mudanças podem ser feitas, em princípio, a qualquer momento. No entanto, a modernização do sistema de medidas também engloba os dispositivos eletrônicos (hardwares) com o sistema operacional e aplicativos ou ferramentas de desenvolvimento, compatíveis entre si para que o sistema possa sofrer alterações futuras. Neste trabalho e com esse enfoque, se procurou acompanhar o processo de evolução tanto de hardware e/ou dispositivos eletrônicos pertinentes quanto de software e/ou aplicativos de apoio ao ambiente de desenvolvimento e de operação, desde quando se iniciou o desenvolvimento do sistema de medidas, até sua entrega em meados de 2006.

Como já mencionado anteriormente, as ferramentas CASE não foram utilizadas neste projeto. Então foram adotadas as ações do processo unificado para se garantir um projeto de software completo e correto, e o seu entendimento para projeções futuras. Estas ações descrevem o projeto através de representações e diagramações similares aos que as ferramentas do RUP prescrevem, e também se fez uso de outras representações gráficas julgadas apropriadas.

Para as modelagens gráficas do sistema de medidas e seus subsistemas, e do desenvolvimento de software, essencialmente, se fez uso dos recursos de linguagem de programação visual comercial, e de ferramentas ou aplicativos disponíveis comercialmente voltados para documentações e programações. Para a engenharia do software de aquisição de dados, se fez uso dos recursos da ferramenta *LabView*, com a qual se desenvolveu os subprodutos do software de aquisição de dados e da interface com usuário.

Para o software de redução de dados se fez uso de linguagem de programação textual, no caso a linguagem C (EMBREE, KIMBLE, 1991), com os recursos do aplicativo *C++ Builder 4*, e para interface com o usuário, foi planejada uma provisão para um uso futuro dos recursos de outro aplicativo da fabricante da *National Instruments*: trata-se da ferramenta *LabWindows*, em suas versões aplicáveis para o desenvolvimento dos subprodutos de software. Também se utilizou alguns recursos do aplicativo *C++ Builder 4* para a recuperação dos dados da antiga plataforma HP, para validar os novos subprodutos gerados para a plataforma PC.

À medida que os desenvolvedores aumentavam o conhecimento sobre o sistema de medidas e tecnologias aplicáveis, os ambientes de desenvolvimento e operação do TA-2 iam evoluindo, tornando-se progressivamente mais robustos e eficientes, visando melhorar o desempenho das funcionalidades do sistema de medidas.

CAPÍTULO 5 PROJETO DO NOVO SISTEMA DE MEDIDAS

Para o projeto do novo sistema de medidas se faz necessária a descrição do sistema e seus subsistemas envolvidos. Esta descrição aborda o sistema de medidas e seus subsistemas usando representações em forma diagramática. As arquiteturas gerais dos subsistemas e do sistema constam do plano de desenvolvimento de software. As especificações do sistema de software são necessárias para que a engenharia dos softwares de aquisição e redução de dados do sistema de medidas possa ser realizada com as características funcionais e operacionais pertinentes.

A combinação de equipamentos, dispositivos eletrônicos, sensores e softwares apropriados para plataforma PC, entre outros instrumentos e aplicativos de apoio, são os elementos que irão prover o sistema de software de aquisição e redução de dados. Esta preparação inclui computadores pessoais, dispositivos de aquisição de dados da *National Instruments* (NATIONAL INSTRUMENTS, 1998) com interfaces de comunicação, as quais, juntamente com o software, permitem adquirir todos os dados analógicos e digitais que serão processados pelo computador PC. Depois de armazenados em arquivos, tais dados são submetidos ao sistema de software de redução de dados para obtenção dos coeficientes com vistas à aplicação no projeto de engenharia.

Iniciou-se o trabalho com a ferramenta de apoio ao desenvolvimento *LabView* em sua versão 3.1, e fez-se necessário evoluir o ambiente de apoio ao desenvolvimento, os equipamentos e os dispositivos eletrônicos da *National Instruments* (NATIONAL INSTRUMENTS, 1998) , para melhoria do projeto do sistema de software. Este ambiente de engenharia era composto basicamente dos seguintes itens:

- Placas de Aquisição de Dados.
- Blocos de terminais.
- Placas de Condicionadores de sinais.
- Chassi com *slots* suficientes para abrigar as placas.
- Cabos de interfaces.

À medida que se constroem os subprodutos de softwares, os dispositivos eletrônicos e equipamentos de configuração básica podem ser alterados, ou utilizados aqueles julgados mais apropriados para as funcionalidades operacionais do sistema de medidas do TA-2. Os instrumentos virtuais criados e usados ou adaptados provenientes do uso da ferramenta *LabView* e que irão constituir o subproduto de software de aquisição de dados devem cumprir as funções mínimas de coletar, processar e armazenar os dados provenientes dos sensores e exercer os controles necessários aos experimentos.

Neste trabalho foram considerados os tipos de dados processados vindos dos sensores, dentre os quais estão, por exemplo, os de forças e de momentos do sistema da balança interna ao modelo ou externa no túnel, ou combinações de ambas as aquisições. Esses dados e outros necessários são processados através dos novos softwares de aquisição e de redução, que incorporam as capacidades e funções para as correções aplicadas ao modelo aerodinâmico ensaiado, incluindo tara, alinhamento e dados de transferência.

Outros tipos de dados que estes softwares também processam são as várias medidas de pressão necessárias para os ensaios de modelos aerodinâmicos. Esses dados de pressão são medidos e computados para obter os coeficientes pertinentes do modelo aerodinâmico, sejam dados vindos de pressões internas ao modelo ou dados de pressão do túnel advindos dos *pitots*.

Os dados em coeficientes são resolvidos e apresentados para os vários eixos apropriados, com respeito aos esforços aerodinâmicos aplicados, medidas de pressões entre outros, que podem levar os clientes a requererem, para algum programa ou campanha de ensaio, requisitos especiais. O cumprimento desses requisitos especiais, dos clientes de ensaios em túnel de vento, deve ser provido pelos softwares de aquisição e de redução de dados.

Algumas vezes o desenvolvedor pode necessitar incluir alterações nos softwares, para medir e garantir a qualidade dessas necessidades especiais. Para entender melhor o sistema de medidas do TA-2, apresenta-se a descrição dos esforços aerodinâmicos, que são os dados de forças e de momentos processados pelos softwares em questão. A figura 5.1 mostra esses esforços e o movimento dos ângulos (α e β) do sistema de medidas do TA-2. Os esforços aerodinâmicos que agem sobre um corpo são os seguintes:

- Força de arrasto (D): é o esforço que o escoamento impõe ao movimento do corpo. Atua na direção do escoamento.
- Força Lateral (Y): é a força responsável pelo desvio lateral do corpo de sua trajetória original. Atua no plano transversal ao corpo e é perpendicular à direção do escoamento.
- Força de Sustentação (L): é a força que atua no plano de simetria do corpo e é perpendicular à direção do escoamento.
- Momento de Arfagem (m): quantifica a tendência do corpo de girar no plano de simetria. O vetor momento de arfagem tem direção do vetor força lateral.
- Momento de Rolamento (r): quantifica a tendência do corpo de girar ao redor de seu eixo de simetria. O vetor momento de rolamento tem a direção do vetor arrasto.
- Momento de Guinada (n): quantifica a tendência do corpo de girar lateralmente. O vetor momento de guinada tem a direção do vetor força de sustentação.

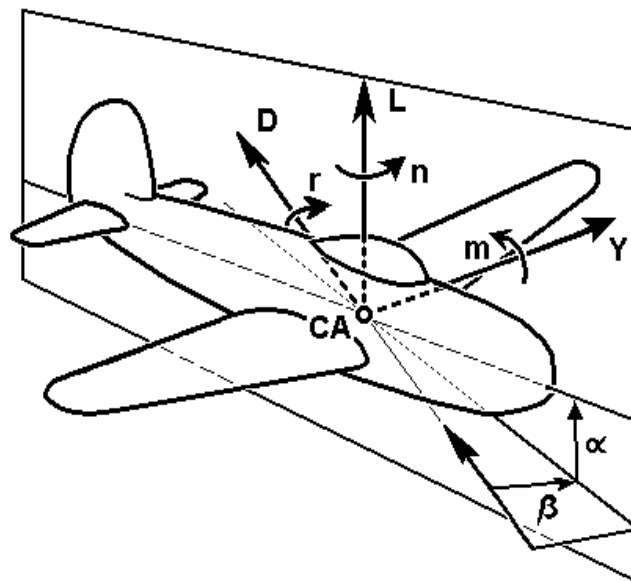


Figura 5.1 - Esforços aerodinâmicos sobre o corpo e ângulos de ataque (REIS, 2000).

O sistema de medidas do túnel de vento também necessita de comandos de controle para funcionar corretamente. Estes comandos são enviados e seus efeitos monitorados pelos operadores do túnel, tendo uma visão do posicionamento e atitude do modelo a ser ensaiado.

O ensaio experimental requer operação dos instrumentos e equipamentos, para, por exemplo, atuar o mastro para posicionamento do ângulo de ataque, iniciar o motor e ajustar a velocidade do vento da seção de ensaio. Os dados destas operações devem ser registrados como as medidas de velocidade do vento, ângulo de ataque e qualidade do fluxo, permitindo também a obtenção de dados das calibrações, previstas para obtenção da matriz de calibração, que serve de base para ajustes dos dados de aquisição de uma campanha de ensaio.

Assim, todos os subsistemas do túnel de vento que compõem o sistema de medidas, estão identificados neste trabalho, com objetivo de apresentar quais os elementos de dados e de controle são incluídos como funções a serem processadas, pelos softwares de aquisição e redução de dados. As características do TA-2 permitem testar os modelos aerodinâmicos em escala e é subsônico. Detalhes destas

características, dispositivos eletrônicos, softwares existentes e equipamentos requeridos para compor este trabalho, estão descritos no Apêndice B.

Fazem parte do sistema de medidas:

- Equipamentos para movimento e sensoriamento de ângulos do modelo.
- Células de carga para medida de forças aerodinâmicas (P1 a P6).
- Sensores de temperatura (T) e pressão (P).
- Sensor de umidade (H).
- Tubos de *Pitot* (q).
- 4 câmeras e 4 monitores.
- 1 gravador de vídeo SVHS.
- 1 placa digitalizadora.

Outro material empregado é a balança aerodinâmica externa, do tipo piramidal, fabricada pela empresa Taller & Cooper de 6 componentes, ou seja, 3 forças e 3 momentos independentes. Um desenho esquemático de algumas instrumentações, sensores, a balança e outros elementos usados nos ensaios do TA-2, é mostrado na figura 5.2.

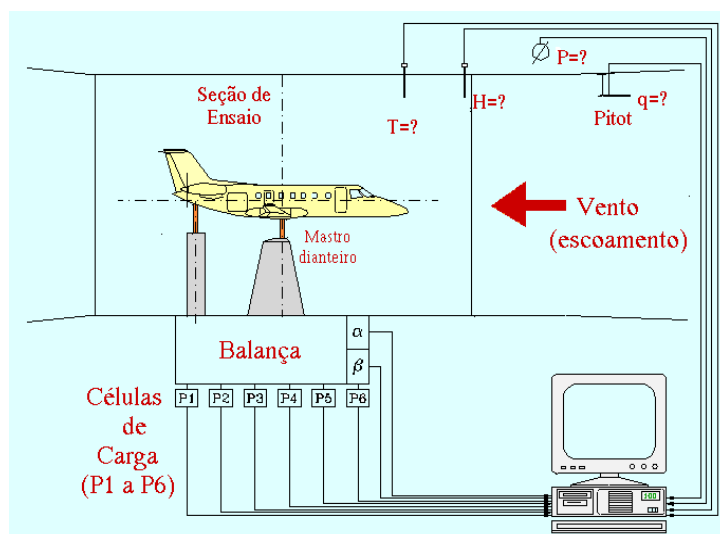


Figura 5.2 - Desenho esquemático dos dispositivos e instrumentações do TA-2 (IAE, 2005).

Existem diversos equipamentos ou dispositivos disponíveis e instalados no TA-2, cabendo então identificar os subsistemas e o uso desses materiais em cada um deles. As descrições dos subsistemas neste trabalho estão limitadas à arquitetura geral ou típica mais utilizada, uma vez que, existe uma grande variedade de materiais, que podem configurar um ambiente de ensaio diferente para cada tipo de modelo aerodinâmico a ser ensaiado.

5.1 IDENTIFICAÇÃO DOS SUBSISTEMAS DO TA-2

Os ensaios em túnel de vento são executados através da Seção de Ensaio, onde o modelo aerodinâmico é montado sobre a balança e o mastro. Nessa Seção estão instalados os sensores de pressão conhecidos como *pitot*; abaixo dela fica a balança externa Taller & Cooper, onde estão instaladas as células de carga. Sobre a balança estão os mastros e neles ficam instalados um controlador do mastro e o *encoder* digital.

O motor e a hélice ficam no final do túnel, com os sensores de movimento e de temperatura instalados. A figura 5.3 mostra a montagem do modelo aerodinâmico na Seção de Ensaio do TA-2.



Figura 5.3 - Montagem do modelo aerodinâmico na seção de ensaios do TA-2 (IAE, 2005).

Na parede da Seção de Ensaios encontra-se a instrumentação para tomada de pressão, e também os sensores de temperatura, umidade e barométrico. Os dados desses sensores são lidos, apresentados e armazenados computacionalmente, no início da calibração ou aquisição dos dados para os ensaios com os modelos. Além desses, também existem os equipamentos para filmagens dos ensaios, que estão instalados sob a Seção de Ensaio e mostram as imagens em tela de televisão instalada na Seção de Controle.

Nesta Seção também estão instalados os computadores PCs e os dispositivos eletrônicos da *National Instruments* (NATIONAL INSTRUMENTS, 1998) e de outros fabricantes, com os quais são apresentadas as aquisições de dados através das interligações desses diversos dispositivos com as instrumentações instaladas na Seção de Ensaios e locais adjacentes do TA-2. A Seção de Controle comporta praticamente todos os equipamentos e dispositivos, para aquisição e redução dos dados dos ensaios dos modelos aerodinâmicos do TA-2.

Adicionalmente, são instalados outros instrumentos de engenharia para ensaios no TA-2, como por exemplo, *Rakes*, *scanivalves*, transdutores de pressão e *strain gage*, quando a equipe de ensaio ou o cliente assim especificar ou necessitar. Neste caso, o sistema de medidas deve possibilitar que os dados sejam obtidos e processados, havendo flexibilidade para adequar essa atualização de instrumentos e funções, ou seja, havendo provisão para uso nos softwares.

Os dispositivos eletrônicos do sistema de medidas do túnel de vento devem possibilitar o condicionamento, a amostragem e quantificação das medidas oriundas de um conjunto de transdutores e/ou sensores. Para execução das medidas de grandeza física é preciso executar um processo de calibração dos transdutores e instrumentos de medidas, determinando os coeficientes da curva de calibração destes, visando minimizar os erros sistemáticos das medidas. A calibração dos instrumentos e transdutores para os ensaios aerodinâmicos no túnel de vento do CTA é feita segundo o seguinte processo:

- a) **Calibração direta dos transdutores ou sensores:** é feita no laboratório de força e/ou de pressão do túnel de vento do CTA. Envolve apenas o componente que se deseja calibrar, neste caso, os transdutores a serem utilizados na campanha de ensaios, como as células de carga, transdutores de pressão, entre outros e o software de aquisição de dados apropriado.
- b) **Calibração direta dos instrumentos:** é feita por organismos credenciados pelo INMETRO, os quais emitem a carta de calibração do instrumento. Abrangem os instrumentos envolvidos na calibração dos transdutores do laboratório de força e pressão, os instrumentos da cadeia de medidas e aqueles necessários ao suporte dos ensaios aerodinâmicos.
- c) **Calibração da cadeia completa de medidas:** é feita pelos especialistas em ensaios em túnel de vento, com todos os instrumentos e transdutores já calibrados (processos **a** e **b**), necessários para obtenção das grandezas físicas do experimento. Os coeficientes obtidos nesta calibração são comparados aos determinados pela calibração direta dos transdutores, considerando ou armazenando esses coeficientes quando houver resultados compatíveis.

Para os processos de calibrações **a** e **c** foram desenvolvidos os softwares de aquisição de dados neste trabalho. No processo **a** o software foi elaborado inicialmente para familiarização com a ferramenta de apoio ao desenvolvimento *LabView*, com objetivo de substituir o registro e o cálculo manual do operador. Foi implementado na plataforma PC, porém, mantendo a instrumentação com comunicação ou interface GPIB ou HP-IB e equipamentos ou instrumento de medidas HP.

No processo **c** emprega-se atualmente o novo software de aquisição de dados desenvolvido com auxílio do aplicativo *LabView*, e em cada subproduto as unidades de software denominadas *Virtual Instruments* “vi”, todos para a plataforma PC, com

dispositivos eletrônicos da *National Instruments* (NATIONAL INSTRUMENTS, 1998) e com instrumentação instalada na Seção de Ensaio e salas adjacentes do TA-2.

Neste trabalho procurou-se abordar a modelagem do sistema de medidas e seus subsistemas, com o enfoque voltado para a engenharia de software. Todos os subsistemas foram abordados, mas praticamente foram dadas ênfases para as funções ou operações interligadas aos softwares desenvolvidos. Alguns processos de engenharia foram também detalhados, devido às interligações com os dados processados pelos softwares. Todo esse conhecimento sobre o sistema de medidas e seus subsistemas do TA-2 são necessários, para a confecção dos softwares, visando garantir a qualidade dos ensaios e a confiabilidade dos dados.

A figura 5.4 mostra a modelagem do sistema de medidas do TA-2 no nível de visão macro do sistema, o qual é composto dos seguintes subsistemas:

- a) Subsistema de Motor e Hélice.
- b) Subsistema de Balança.
- c) Subsistema Mastro.
- d) Subsistema de Sensoriamento.
- e) Subsistema do Modelo Aerodinâmico.
- f) Subsistema de Visualização do Ensaio.
- g) Subsistema de Aquisição de Dados.
- h) Subsistema de Redução de Dados.

O conhecimento dos processos de engenharia dos subsistemas interligados ao software de aquisição de dados é importante para a construção do software de redução de dados. Os diversos arquivos de dados armazenados pelo software de aquisição de dados são processados pelo software de redução, dentre estes os dados da calibração. Com estes dados da balança é possível gerar a matriz de calibração de determinados modelos aerodinâmicos de ensaio, que são fundamentais na fase de pré-ensaio, pois estes determinam a continuidade do programa de ensaio em túnel.

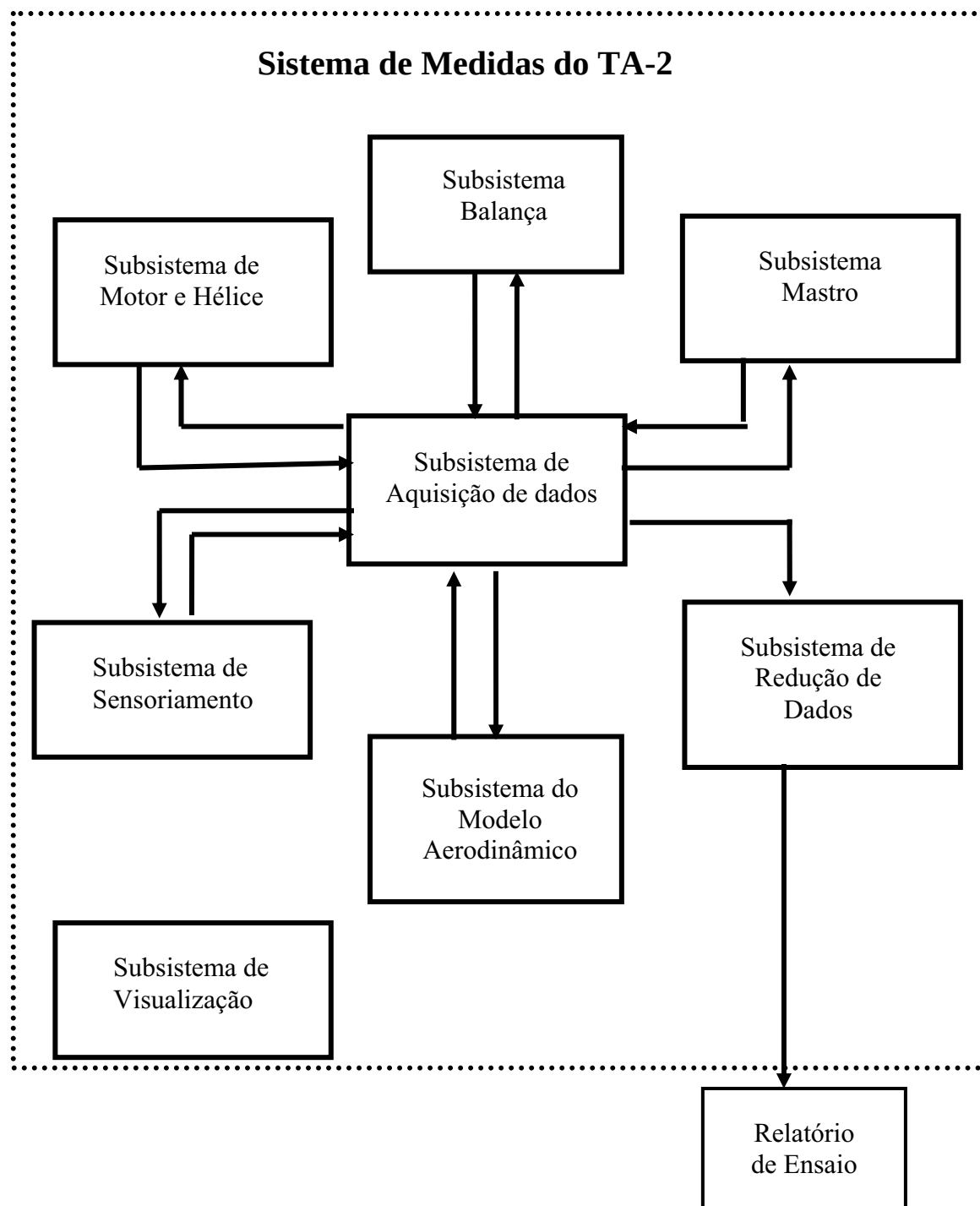


Figura 5.4 - Sistema de medidas e seus subsistemas.

O processo de obtenção da matriz de calibração foi o principal foco deste trabalho, sendo que as práticas implantadas da METAPLI tiveram, como objetivo principal, a documentação do projeto do sistema de software para obtenção da qualidade dos dados, dirimindo as possíveis fontes de erro humano.

5.1.1 Subsistema motor e hélice

O subsistema de Motor e Hélice é responsável por fornecer o vento necessário para execução dos ensaios nos modelos alojados na seção de ensaio, contendo os dispositivos e instrumentações para um controle de velocidade do túnel de vento. Ele foi projetado com dispositivos de segurança para o caso de não conformidades críticas durante a execução de ensaios, visualizados na tela da TV, que permite o desligamento rápido por intermédio de botões de emergência na sala de apoio da seção de ensaios e na sala de medidas da seção de controle. O processo de ligação do motor e hélice ou partida elétrica do túnel de vento também é provido por esse subsistema.

Este subsistema possui especificações especiais, pois deve apresentar resposta compatível às mudanças de rotação requeridas, ou seja, devem reagir de forma imediata aos comandos e fornecer as informações necessárias para monitoramento da velocidade do vento durante o ensaio. Trata-se do motor elétrico de 1600 HP para geração do vento e hélice de fibra de vidro, que substitui as de madeira freijó do sistema de medidas antigo. A figura 3.7 ilustra o subsistema de Motor e Hélice.



Figura 5.5 - Motor e hélice do TA-2 (IAE, 2005).

O motor é do tipo AC trifásico de indução com anéis. A tensão de trabalho é de 4160 V, com corrente máxima 400 A e potência máxima 1600 HP, atinge a rotação máxima de 400 rpm. A hélice possui características como: número de pás igual a 7, com diâmetro de 8,40 m, construídas em fibra de vidro, com mecanismo de variação e monitoração de ângulo do passo.

O controle do vento de forma automatizada é feito através de comandos relativos à variação da rotação do motor e monitorado durante os ensaios, comandos estes, nesta modernização, providos pelo novo software de aquisição. O subsistema de Motor e Hélice deve suportar variações de rotações e cargas, que estão sujeitos a diversas alterações por hora, em condições operacionais.

Neste novo software de aquisição de dados, estão previstos painéis virtuais para acionar e monitorar o motor e a hélice, de forma a possibilitar visualizar no monitor do computador PC, por exemplo, a temperatura do óleo, a rotação do motor entre outras condições apresentadas pelo motor.

Nas primeiras fases do desenvolvimento do sistema de medidas, optou-se por construir inicialmente partes do software de aquisição de dados de outros subsistemas, ou seja, continuou se utilizando neste subsistema o hardware e o software instalados no TA-2, e adquiridos de terceiros como produto COTS e a hélice em madeira freijó.

Os produtos COTS executavam as funções requeridas pelo subsistema em questão, o que era aceitável. No entanto, era considerado um produto caixa preta, ou seja, toda a manutenção ou reparos eram feitos pelo fabricante, que em caso de pane no TA-2, significava uma dependência total desta tecnologia.

Esses dispositivos de controle do subsistema do motor e hélice foram comprados como produtos *stand alone*, ou seja, módulos separados do subsistema de aquisição de dados. Houve alguns problemas operacionais na integração ao sistema de medidas,

chegando a paralisar alguns ensaios até que o hardware e o software fossem ajustados. Isso obrigou os especialistas ao acionamento do fabricante e a indisponibilidade temporária dos dispositivos durante a execução dos ensaios.

Por não ser uma tecnologia aberta, a solução gerencial implementada de imediato foi estudar a possibilidade de acesso ao código fonte junto ao fabricante, o que custaria muito caro. A apresentação de uma solução técnica que suprisse esta falha temporária do dispositivo de controle foi sugerida em reuniões com especialistas do TA-2. Trata-se de reativar em paralelo o painel de controle utilizado antigamente, até que o problema fosse resolvido via implantação dos requisitos pertinentes a estes controles pelo novo subsistema de aquisição de dados.

Com relação a esse subsistema, os novos dispositivos eletrônicos e instrumentações foram sendo integrados ao sistema de medidas, conforme iam sendo projetados e adquiridos os recursos necessários no nível de software e hardware. Devido a sua complexidade, este se tornou o último dos subsistemas a ter suas funções implantadas no software de aquisição de dados. As práticas da METAPLI consideraram o reuso de software, à medida que crescia o domínio do conhecimento do problema.

Os recursos funcionais implantados pelo sistema de software eram COTS, não podendo ser alterados ou atualizados, uma vez que o código não estava disponível. Então, na modernização deste subsistema foi projetado um novo software com inovação do projeto de engenharia, para o qual os requisitos foram especificados e refinados. À medida que se amplia o conhecimento da operacionalidade do sistema de medidas integrado, torna-se possível o reuso de funções e do conhecimento com respeito aos processos, tecnologias e demais partes de software dos subsistemas já implementados.

As funções como a partida e parada do motor, o monitoramento das temperaturas e pressão do óleo, a proteção contra excessos de velocidade, a telemetria, entre outros

desempenhos, foram incorporados de forma gradual. O controle e a indicação dos valores desses componentes estão integrados no software de aquisição de dados, na fase final de aprovação do projeto de modernização do novo sistema de medidas do TA-2.

5.1.2 Subsistema de balança

O subsistema de balança é responsável pelo suporte, balanceamento e alinhamento do modelo instalado na Seção de Ensaio, posicionando-o na angulação necessária. Este subsistema é constituído de projeto integrado de software, hardware, instrumentações e operações automáticas e manuais, que permitem que o modelo aerodinâmico em conjunto com os recursos operacionais do TA-2, seja calibrado, alinhado, movimentado e monitorado.

Muitos procedimentos existem para que o subsistema de balança funcione corretamente, a interdependência deste subsistema com sistema de medidas como um todo, é que foi tratada para obter um bom funcionamento do sistema de medidas. Para a criação dos softwares e integração ao sistema foram considerados neste trabalho os recursos funcionais e operacionais desse subsistema.

A balança externa na planta do túnel foi construída abaixo da Seção de Ensaios conforme mostrado na figura 5.6. É uma balança mecânica da Taller & Cooper para medir seis componentes, sendo piramidal, de centro virtual, resolvendo as forças atuantes no modelo de ensaio em três forças ortogonais e seus momentos associados. Esses seus componentes são medidos durante o ensaio e sob um sistema de vento orientado a eixos, tendo sua origem no centro de resolução da balança, que corresponde ao centro geométrico da Seção de Ensaios.



Figura 5.6 - Balança externa do túnel de vento (IAE, 2005).

As células de carga estão instaladas interligadas na balança, abaixo da Seção de Ensaio, das quais o software de aquisição de dados obtém os sinais de forças e momentos, transmitidos desses instrumentos para a seção de controle. Essas células de carga são calibradas no laboratório de força e seus coeficientes de calibração são inseridos no software de aquisição. Os dados da calibração da cadeia de medidas completa são comparados com esses coeficientes das células de carga, e devem ser pelo menos similares a estes.

5.1.3 Subsistema mastro

A balança sustenta o suporte também conhecido como mastro que é responsável pelas posições de atitude do modelo aerodinâmico ou, mais claramente, do movimento dos ângulos de ataque e guinada. Para cada modelo de ensaio o tipo de suporte pode variar. Atualmente o túnel possui alguns tipos de configuração de mastro possíveis, tais como: com dois mastros, com um mastro e com mastro imagem usado para calibração.

Por meios dos mastros todas as cargas são transmitidas para a balança externa. No entanto, é transmitida também a interferência destes mastros, ocasionando erros nas medidas, havendo a necessidade de suprimir estes efeitos, utilizando as correções de túnel, que são cálculos complexos executados pelo software de redução de dados, para aproximar os valores medidos em túnel aos valores reais.

O movimento do mastro ocorre pelo comando do operador, através do software de aquisição de dados. Com este software deve ser possível determinar o grau do ângulo α (subida ou descida), ou do ângulo β (esquerda ou direita) de forma operacional, através do mouse, e também de forma automática com uma seqüência de ângulos pré-determinada, a qual o operador seleciona.

O mastro pode ser desativado temporariamente, quando se fizer uso da cruz de calibração, que é um elemento mecânico montado sob a balança com pratos para carregamento de massas, cabos e roldanas, não sendo aplicada a função de atitude, pois o mastro neste caso não estará operacional. Apesar de muitas atividades implementadas nos ensaios em túnel serem, em sua maioria, de cunho manual como, a montagem da cruz de calibração e colocação de massas, a montagem do mastro e carenagem, considerou-se importante mencionar que o software de aquisição de dados deve prover o registro de cada massa durante a calibração.

Isto deve fazer parte do conhecimento dos desenvolvedores e dos parceiros, para a implementação correta dos demais subsistemas computacionais, além do posicionamento correto dos ângulos e seu registro e armazenamento de dados. Uma representação da cruz de calibração pode ser vista na figura 5.7.

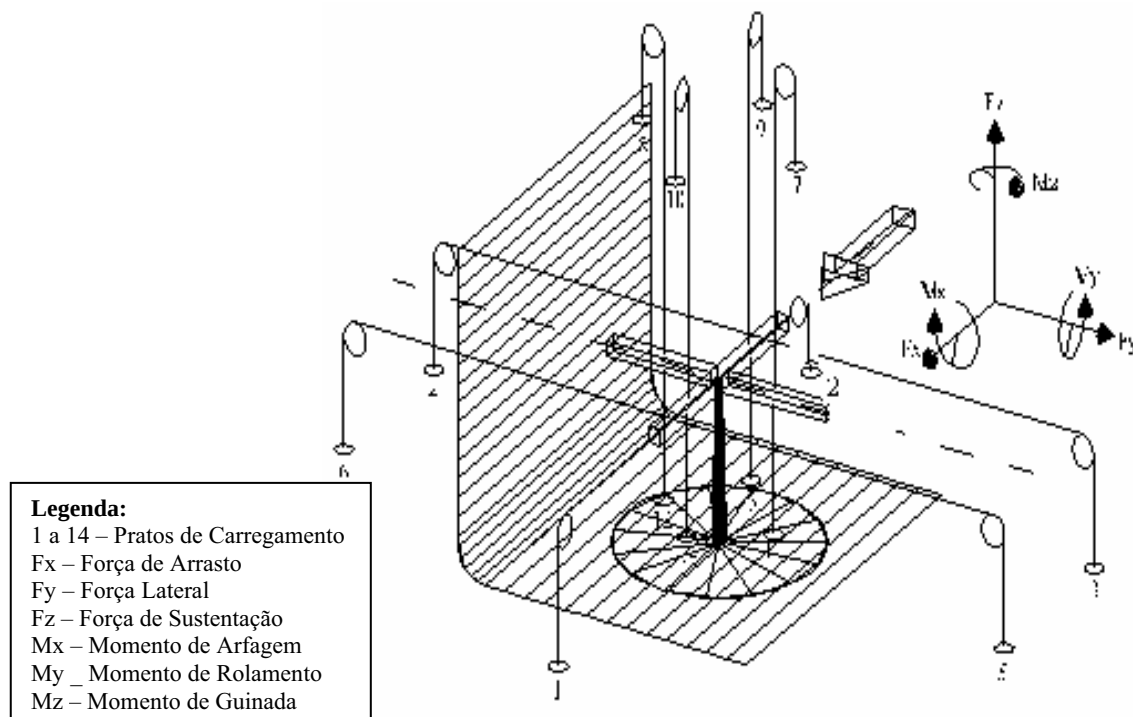


Figura 5.7 - Cruz de calibração do TA-2 (REIS, 2000).

O processo de calibração com a cruz faz parte da calibração da cadeia completa de medidas, da etapa de pré-ensaio. Este utiliza algumas roldanas fixadas em locais previstos na sala de apoio à seção de ensaios, para suportar os cabos e as massas colocadas sobre os pratos da cruz de calibração, identificados com os números de 1 a 14, conforme mostrado na figura 5.7. Com uma lista de carregamentos em mãos, gerada anteriormente pelo computador, o operador ajusta os movimentos manualmente e atentamente para a operação funcionar corretamente.

Esses carregamentos são medidos pelos sensores de carga instalados na balança externa, e o armazenamento e monitoramento das medidas são feitos através do software de aquisição de dados. Esses dados armazenados são processados posteriormente pelo subsistema de redução de dados, que gera a matriz de calibração, a qual servirá para obtenção dos resultados dos ensaios aerodinâmicos.

O carregamento de carga na balança é feito manualmente por elementos da equipe do TA-2 nesta cruz de calibração, e o registro da carga é feito através da entrada correta dos dados via computação. Para determinado número de carregamento

é então estabelecida uma carga ou descarga de massa em determinados pratos dessa cruz, correspondente às forças e/ou momentos pertinentes.

Neste cenário, um erro de carregamento ou de armazenamento de dados, pelo software de aquisição de dados, compromete a determinação das matrizes de calibração da balança. Da mesma maneira, pelo sistema de eixo um fenômeno físico medido correspondente a uma força pode influenciar ou ser influenciado por outros elementos de força e/ou momento, tendo, ao final do processo de calibração da balança, a computação comprometida de ambos os elementos envolvidos.

Devido à importância das calibrações em ensaios em túnel de vento, a implantação dos recursos funcionais e operacionais no software de aquisição de dados é considerada complexa. O sistema de software como um todo deve prover recursos que considerem o desempenho dos processos de engenharia de forma correta e completa.

Conhecer tanto os processos de engenharia quanto os subsistemas com processos de calibração e aquisição de dados, é essencial para a construção dos subprodutos de softwares deste trabalho e evoluções até a versão final. Um ponto importante considerado foi com respeito aos dados dessas calibrações, pois foram implantados critérios para comparações desses dados com aqueles obtidos pela plataforma HP, para a aprovação dos softwares de aquisição e redução de dados.

5.1.3.1 Arquitetura do subsistema mastro

O subsistema mastro faz uso de um processo de controle operador/atuador/sensor, e também é uma das partes do subproduto do software de aquisição de dados. O modelo de arquitetura geral é mostrado na figura 5.8.

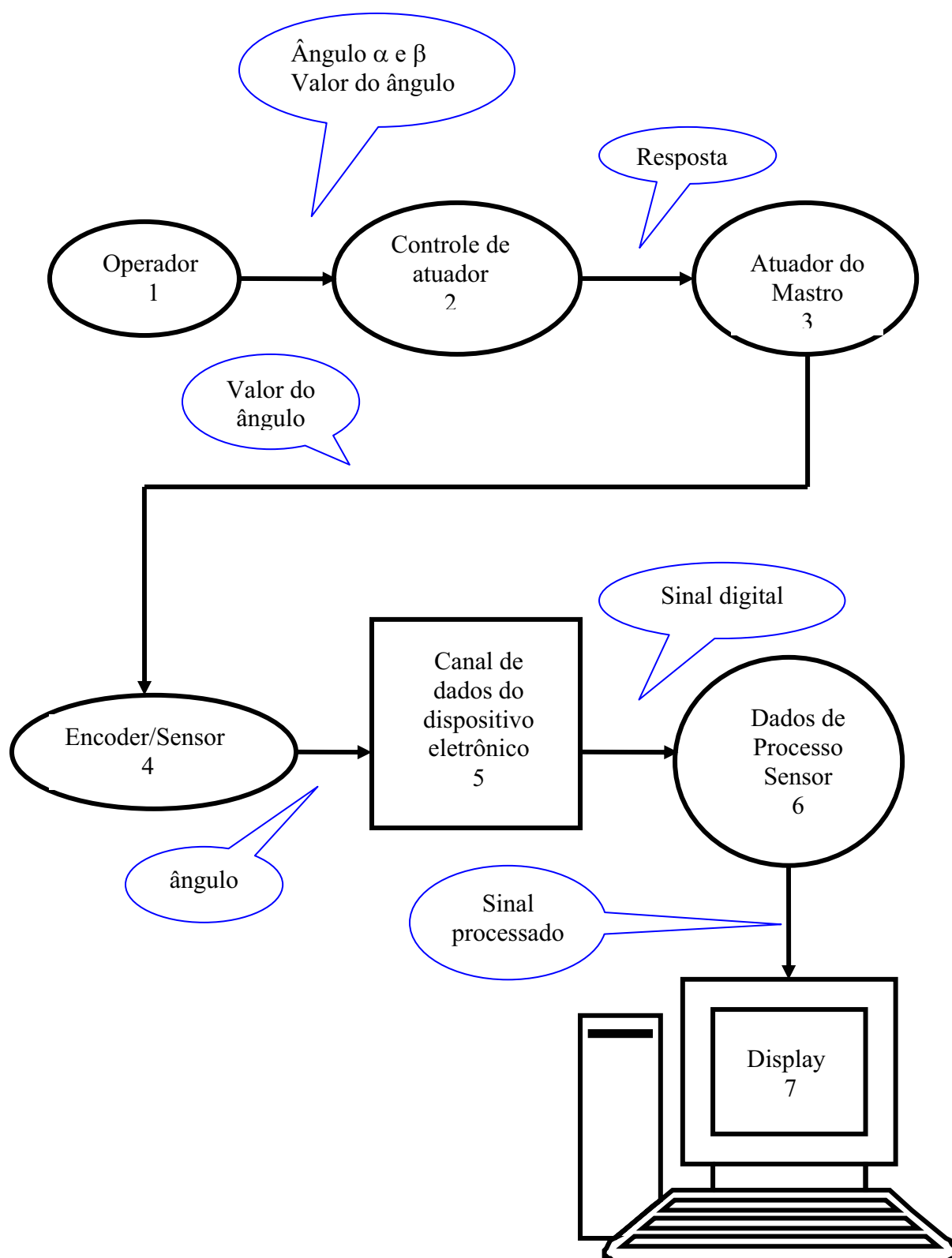


Figura 5.8 - Arquitetura geral do subsistema mastro

A arquitetura de controle operador/atuador/sensor do subsistema mastro, interligada ao software de aquisição de dados e ao dispositivo eletrônico pertinente, é organizada de modo que o controle seja transferido do operador ao atuador e o dado deste estímulo registrado pelo sensor e apresentado como resposta ao operador, para o registro em vídeo e armazenagem da posição do modelo e do valor do dado, para posteriormente serem processados corretamente pelo software de redução de dados.

5.1.4 Subsistema de sensoriamento

O subsistema de sensoriamento é composto por sensores e transdutores responsáveis por fornecer informações das forças, momentos, pressão, temperatura, atitude e outras medições necessárias sobre o ensaio do modelo aerodinâmico, para o subsistema de aquisição de dados. As medidas de forças, momentos são transmitidas ao computador através de células de carga e as posições de atitude via *encoder* que transmite sinais digitais diretamente para dispositivos de aquisição de dados e sistema de software de aquisição de dados, para processamento.

5.1.4.1 Arquitetura do subsistema de sensoriamento

Uma das partes do subproduto do software de aquisição de dados trata das interligações com o subsistema de sensoriamento, pelos quais os dados dos sensores passam por determinados canais do dispositivo eletrônico. Cada dispositivo tem um canal referente ao sensor convencionado, e estes foram definidos com ambientes diferentes em cada uma das etapas de construção do software de aquisição de dados.

Cada sensor tem um processo associado que converte o sinal de entrada analógico para um sinal digital, através do dispositivo eletrônico definido para cada construção do desenvolvimento. Os sinais dos sensores passam por seus canais convencionados do dispositivo eletrônico, que faz o processamento e os transmite para o software de aquisição de dados, que os apresenta no display para a saída na tela

específica, ao operador, tendo como modelo de arquitetura geral do subsistema sensor, mostrada na figura 5.9.

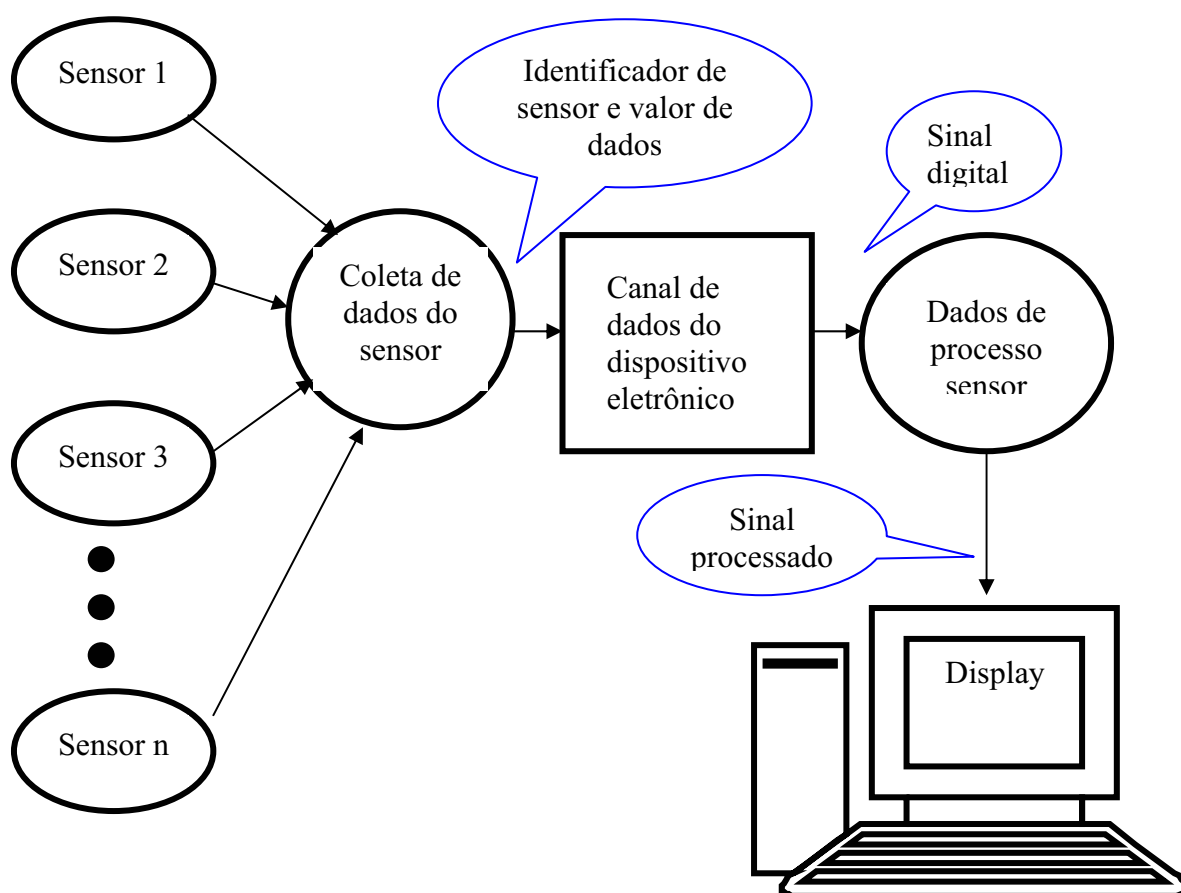


Figura 5.9 - Arquitetura geral do subsistema de sensoriamento

5.1.5 Subsistema de montagem do modelo aerodinâmico

Muitos sistemas de montagem diferentes são usados para suportar o modelo aerodinâmico para balança externa. A estrutura básica, as ferragens e ferramentas de fixação e a base do modelo aerodinâmico devem manter a máxima rigidez possível, e podem combinar as características de leveza e resistência. Para montagem do modelo aerodinâmico no TA-2, em alguns ensaios, são utilizadas as carenagens aerodinâmicas,

cujas funções são proteger os dispositivos eletrônicos, instrumentações e os fios de qualquer influência do fluxo de ar em sua volta.

Essas carenagens são montadas em volta do mastro e mantêm uma forma aerodinâmica, para não causar efeitos no modelo aerodinâmico, que poderiam influenciar nas características físicas a serem medidas pelo software de aquisição de dados. A carenagem é desmontável e no TA-2 existem em pares, para o caso de usar dois mastros para suportar o modelo. Pode parecer estranho mas a não proteção dos fios e cabos e das instrumentações e dispositivos exercida pela carenagem, influencia muito nos dados obtidos pelo software de aquisição: qualquer contato dos fios e cabos com o mastro, por exemplo, altera totalmente os sinais computados.

Uma informação importante sobre o modelo aerodinâmico de aviões, por exemplo, é que estes normalmente já vêm dos clientes, com um *KIT* de materiais de superfícies de controle (*flap*, *slat*, *aileron*, *spoilers*, *bordo de ataque*, *bordo de fuga*, etc), que nos intervalos programados entre uma aquisição de dados e outra, são substituídos por aqueles especificados pelo cliente, no programa de ensaio.

Os dados dessas configurações do modelo aerodinâmico devem ser armazenados pelo software de aquisição, corretamente, ou seja, devem ser armazenados aqueles tipos de materiais instalados na estrutura do modelo aerodinâmico como: *Ailerons*, *Flaps*, *Slats*, *Spoilers*, entre outros, que se situam no bordo de ataque e bordo de fuga da asa, e também os recursos usados com respeito à rugosidade da estrutura.

5.1.6 Subsistema de visualização

Este subsistema permite a filmagem e visualização das diversas técnicas utilizadas para verificação do escoamento no modelo aerodinâmico. Essas técnicas se constituem de aplicações de tufo de lã, óleo, tinta ou fumaça no modelo aerodinâmico

e visualização dos escoamentos, para análise e registro dos resultados dos ensaios. As figuras 5.10 e 5.11 mostram o uso de algumas destas técnicas.

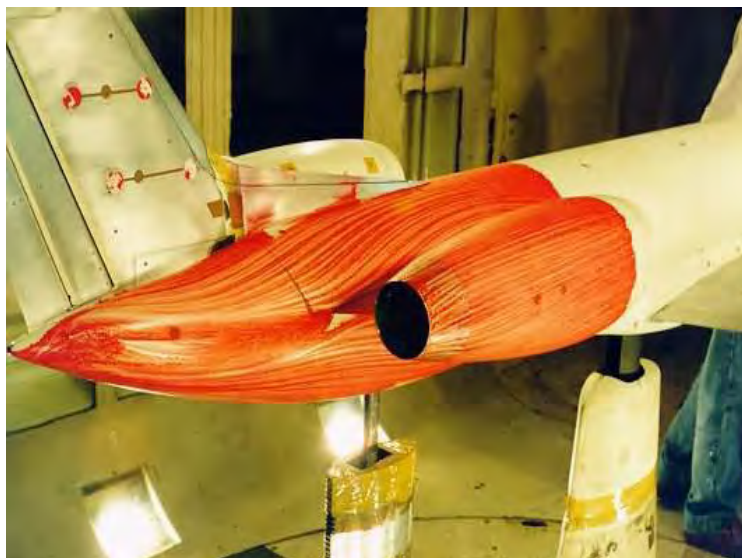


Figura 5.10 - Visualização com tinta (IAE, 2005).



Figura 5.11 - Visualização com fumaça (IAE, 2005).

O subsistema de visualização, o qual abrange os equipamentos de filmagem do modelo aerodinâmico na Seção de Ensaio, durante a execução deste, é composto de equipamentos e software COTS, onde os produtos das filmagens são descarregados em um computador PC, sem a pretensão de verificar computacionalmente qualquer

fenômeno físico, ou seja, o produto da filmagem servirá apenas para um registro. A análise visual do escoamento é feita no produto deste filme ou na inspeção do modelo após o ensaio. Em alguns ensaios os especialistas usam os recursos materiais adicionais disponíveis no TA-2, para uma melhor análise dos fenômenos físicos.

Alguns desses ensaios necessitam de softwares específicos que neste trabalho foram considerados como um sistema de software COTS. O software que reproduz os dados de visualização não foi desenvolvido no TA-2, tendo sido apenas especificado para ser compatível com a plataforma PC. Esse é considerado *stand alone*, portanto, não existe fluxo de dados ou de controle entre esse subsistema de visualização e os demais subsistemas do sistema de medidas. Por outro lado, reproduzem experimentos extremamente úteis, para demonstração dos resultados aos clientes, e por este motivo foi um subsistema modernizado e descrito neste trabalho.

5.1.7 Subsistema de aquisição de dados

O sistema de medidas do TA-2 tem como objetivo efetuar os ensaios dos modelos aerodinâmicos, em suas configurações requeridas e nas condições de velocidade do vento e ângulo de ataque. Auxilia na identificação dos problemas no projeto e na recomendação de possíveis soluções o quanto antes. Ele serve como ferramenta de auxílio aos projetos de aeronaves, navios, prédios, etc, antecipando informações importantes para a melhoria destes projetos, antes de submetê-los às situações reais de voo, navegação, construção, etc. O sistema de medidas deve permitir a inserção de novos dispositivos de geração e aquisição de sinais, caso necessário.

O subsistema de aquisição de dados tem o objetivo principal de obter as informações dos subsistemas envolvidos integrados no ensaio do sistema de medidas, assegurando o funcionamento adequado desses subsistemas e a confiabilidade dos dados e resultados obtidos. Esse subsistema de software deve fornecer as características e executar as funções necessárias dos dispositivos inseridos,

reaproveitando componentes que correspondam às grandezas físicas a serem medidas, às ativações e à modelagem do comportamento do modelo ensaiado.

Existem várias tecnologias para desenvolvimento de sistemas de aquisição de dados, e os softwares devem implementar os recursos funcionais que satisfaçam as funções requeridas para aquisição, análise e armazenagem de dados. Assim, deve-se detalhar no software a arquitetura do sistema de aquisição de dados, que deve considerar a arquitetura típica de um sistema de aquisição de dados, conforme a figura 5.12.

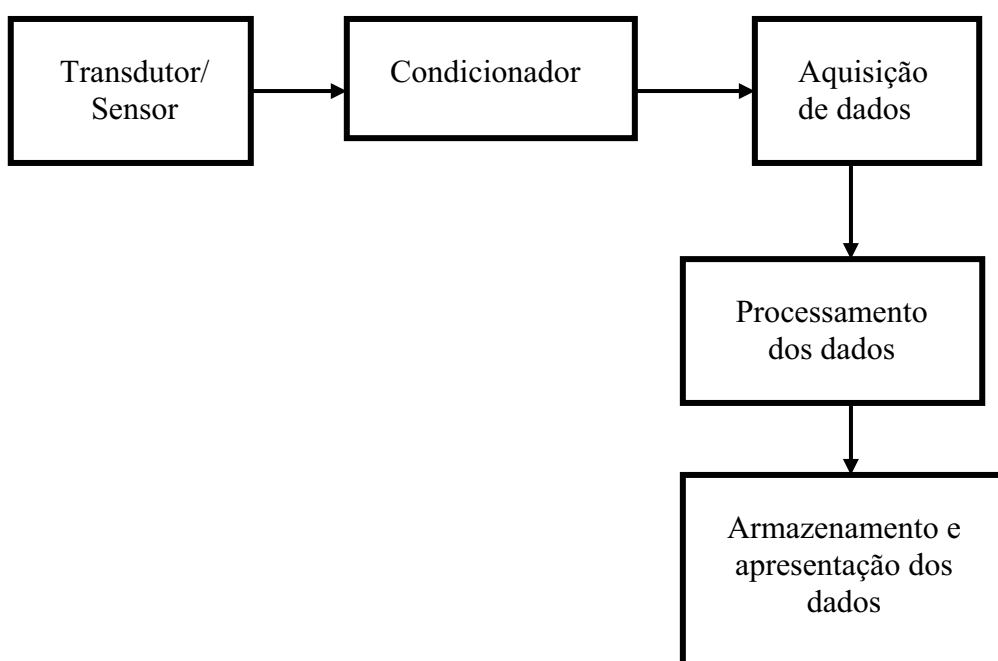


Figura 5.12 - Arquitetura típica de um sistema de aquisição de dados.

Para atender aos requisitos das diferentes aplicações, os ganhos dos sinais, por exemplo, correspondentes a cada sensor podem ser inseridos manualmente pelo operador, sendo que o software de aquisição deve prover o cálculo posteriormente, para comparação. Esses ensaios envolvem a ação direta do operador que estabelece as condições de aplicabilidade do teste.

A capacidade funcional do sistema de medidas deve ser suficiente para:

- Adquirir os sinais dos sensores instalados na Seção de ensaio e na balança.
- Atuar o mastro para mudança de ângulo de ataque em arfagem (*pitch*) e guinada (*yaw*).
- Calibrar a balança e armazenar os dados para obtenção da matriz de calibração.
- Acionar o motor e hélice.
- Obter e monitorar a rotação, as temperaturas de óleo e mancal do motor.

Com o uso das ferramentas de apoio adequadas ao desenvolvimento de software e de hardwares de multifunções, conectados e/ou instalados nos computadores pessoais, foram gerados e documentados os projetos dos subprodutos de software, para medir os fenômenos físicos como temperatura, pressão dinâmica, velocidade do vento, movimento das hélices e do motor, entre outros, e para os controles dos atuadores, através do reuso de um extenso número de biblioteca de funções de uso geral, disponíveis comercialmente.

5.1.8 Subsistema de redução de dados

O objetivo principal do subsistema de redução de dados é ler as informações dos diversos arquivos de dados gerados pelo subsistema de aquisição de dados, efetuar os cálculos e transformações segundo os recursos de engenharia e, possibilitar a conformidade com o programa de ensaio fornecido pelo cliente, através do ajuste de alguns elementos de entrada, relativos às especificações do objeto de ensaio, também fornecidas pelo cliente.

Para desenvolver o software de redução de dados optou-se inicialmente, por estudar o programa de redução de dados em linguagem BASIC denominado “RED1701OS”, de junho de 1999 e suas capacidades operacionais para o ambiente HP, visando entender melhor seu funcionamento. O programa em linguagem BASIC

executava uma diversidade de funções dentre elas a leitura de arquivos contendo dados processados e tratados por outros programas em BASIC na HP que foram abordados na reengenharia.

Sabe-se que o cérebro humano comete uma infinidade de erros quando se tem de manipular muitos detalhes, e os erros mais comuns na fase de implementação segundo Sommerville (2003) são:

- erros na programação;
- erros na manipulação de dados;
- ambiguidades;
- omissões.

Este autor Sommerville (2003) considera a programação uma das atividades mais trabalhosa e propensa a erros. O uso de diálogos em tela, acesso a banco de dados, geração de relatórios e especificações precisas são as técnicas que devem ser aplicadas, para que o código possa ser gerado com qualidade, e este foi o enfoque adotado para a confecção do software de redução de dados para plataforma PC.

Para o desenvolvimento do software de redução de dados foram utilizadas técnicas e padrões de engenharia de software como:

- a) Envolver o usuário final durante o desenvolvimento do software, de modo a ser informado, quanto às dificuldades de entendimento da especificação de requisitos e ser alertado quanto a problemas que poderiam causar erros na confecção deste software.
- b) Linguagem C (JAMSA, 1988) por se tratar de uma linguagem de programação, cuja estrutura básica permite o emprego de técnicas mais atualizadas de estruturação de dados e código. O aplicativo C++ Builder 4 Standard para Windows 95/98/NT foi utilizado para compilar e depurar o novo software.

- c) Checagem dos dados e das funções utilizadas, com o objetivo de verificar se o software atendia completamente as necessidades do usuário final. Checagem de procedimentos de ensaios e técnicas de engenharia para entender as informações das funções implementadas, analisando práticas recomendadas em bibliografias (BARLOW, ERA, POPE, 1999) com respeito a ensaios em túnel de vento, e cálculos similares aos utilizados no programa em linguagem BASIC da HP (NOGUEIRA, 1980).
- d) A análise criteriosa dos dados para melhor definir as estruturas destes dados, de modo a permitir o compartilhamento dos dados pelos diversos módulos ou processos e compor a reengenharia do software de redução de dados.
- e) O conceito de dividir para conquistar, de maneira que o software pudesse ser constituído de módulos independentes, pequenos, manipuláveis e testáveis individualmente.
- f) O processo de refinamento sucessivo, definindo-se os dados e as tarefas básicas, resolvendo os problemas de programação e funcionais. A cada etapa de refinamento foram tomadas decisões sobre evoluir ou refazer partes do código e testá-lo novamente, até que a alteração fosse considerada como uma alternativa de solução aceitável em termos de programação, funcionais e operacionais.

Para a checagem dos dados de aquisição, o procedimento adotado foi a análise da estrutura dos dados e programas em BASIC da HP, com dados de entrada dos ensaios anteriores que geraram os dados de saída e a comparação com os arquivos de dados transferidos para o PC, verificando e validando todos os elementos dos arquivos gerados, que se tornariam os dados de entrada para o software de redução de dados.

A solução técnica para analisar os requisitos e desenvolver o software de redução de dados foi aplicar a filosofia estruturada aos problemas. Dentre as técnicas

utilizadas podemos citar: decomposição funcional, abstração de dados e diagrama de fluxo de dados.

Através dessas técnicas foram gerados os documentos de arquitetura de projeto do novo sistema de software. Sendo assim, a reengenharia para o projeto e a análise estruturada foi descrita utilizando as práticas da METAPLI para modelagem do software de redução de dados baseado em diagramas de fluxo de dados.

Os requisitos do software de redução de dados são:

- Funcionar em ambiente operacional adequado para plataforma PC, visando as leituras dos dados, execução dos cálculos e obtenção de resultados conforme apresentados pelo ambiente HP.
- Implementar funções e cálculos condizentes com as condições estruturais do objeto ensaiado e com as condições ambientais do Túnel de Vento.
- Ler dados de entrada gerados pelo software de aquisição de dados ou por softwares que trataram os dados da aquisição.

Todas as práticas da METAPLI para desenvolvimento dos softwares de aquisição e redução de dados no novo sistema de medidas do TA-2, tiveram início com o estabelecimento das ações das metodologias ágeis e tradicionais aplicáveis ao desenvolvimento. Estão descritas neste trabalho, as técnicas para avaliação e escolha dos modelos de processo que mais se adequaram a esse projeto de modernização.

Para melhoria da qualidade do processo de calibração da célula de carga foi planejada a avaliação das técnicas empregadas para aceitação de viabilidade em termos de custo, e para confirmar se ao implantá-las, o objetivo de melhorar a produtividade e a confiança no sistema e nos dados obtidos seria alcançado. Para conduzir o trabalho de confecção dos softwares com as técnicas selecionadas, adotou-se um ciclo de vida de desenvolvimento de software apropriado e a aplicação das técnicas de reuso e reengenharia.

5.2 PROCESSO DE APLICAÇÃO DE REUSO.

No desenvolvimento do software de aquisição de dados do TA-2 com a abordagem iterativa, planejou-se o aproveitamento de parte dos subprodutos de software desenvolvidos internamente e daqueles adquiridos para serem usados. As várias iterações possibilitariam a seleção dos subprodutos julgados adequados para serem integrados na arquitetura do sistema de medidas do túnel. Assim, o ciclo de vida iterativo facilitaria a reutilização, por permitir a identificação de partes comuns ou semelhantes daqueles softwares, quando parcialmente projetados ou implementados.

Durante as iterações de elaboração, a identificação de partes reutilizáveis, disponíveis pelo fabricante do ambiente de programação gráfica visual, não se mostrou tão fácil, limitando-se apenas ao *driver* de comunicação GPIB. Nas iterações iniciais surgiram identificações de reutilizações inesperadas e potenciais, sem que o desenvolvedor de software as identificasse prontamente, e sem terem sido especificadas adequadamente para o reuso no software de aquisição de dados do laboratório. Somente nas iterações subseqüentes, depois de identificadas as partes com potencial de usabilidade e com a aplicação de técnicas de reuso, foi possível desenvolver um código aceitável. Esse código foi sendo posteriormente melhorado e amadurecido, até tornar-se adequado para o reuso no sistema de medidas integrado.

A análise de componentes de software existentes para possibilidade de reuso deve ser desempenhada através das seguintes tarefas:

- Identificação de componentes de reuso, com respeito aos requisitos funcionais.
- Avaliação da qualidade desses componentes, verificando e validando os requisitos para reutilizar no software, como aplicado em qualquer outra parte do desenvolvimento.
- Compatibilização dos métodos de engenharia com o modelo computacional.

Podem-se definir procedimentos, métodos e ferramentas para reuso e aplicá-las no processo de engenharia de software adotado, para completar o requisito de reusabilidade no desenvolvimento de software, que é um requisito não funcional. Os subprodutos COTS, os desenvolvidos para reuso, os COTS modificáveis, os softwares com os códigos fontes abertos, são entendidos como parte do produto de software final com abordagem de reuso. No fluxograma mostrado na figura 5.13 se propõem alguns passos para aplicar o reuso considerando a estabilidade dos requisitos.

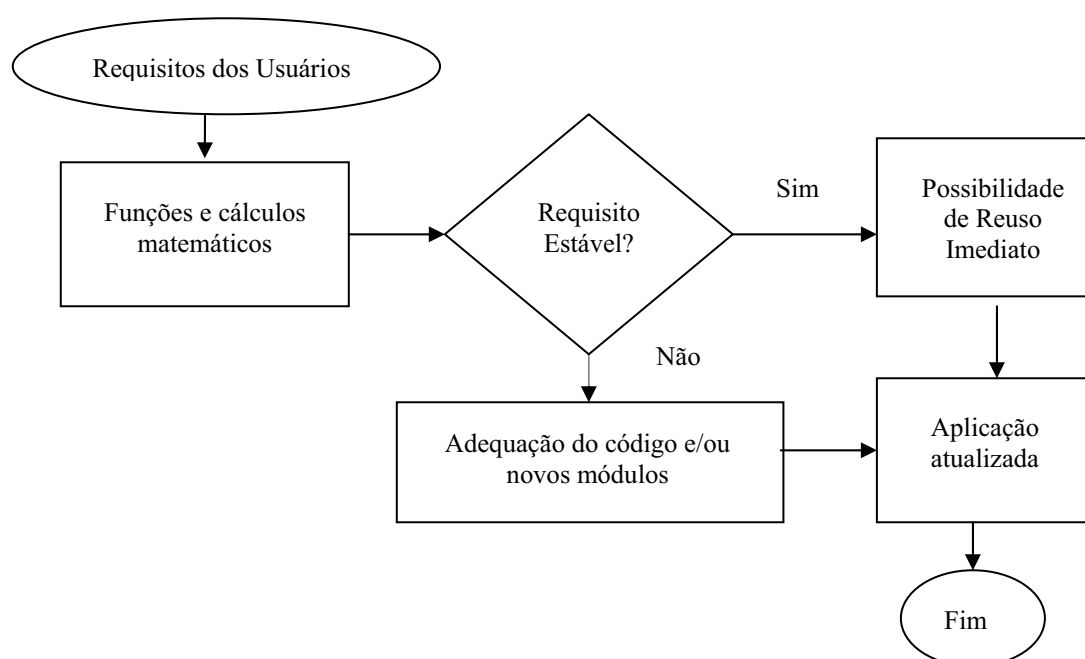


Figura 5.13 - Fluxograma de aplicação de reuso com requisitos estáveis.

Os requisitos do sistema de software podem variar durante o processo de desenvolvimento com uso do processo iterativo. Por isso, somente após várias iterações é que seria possível caracterizar o reuso de partes dos subprodutos. Os requisitos considerados estáveis podem ser implementados na prática de forma mais rápida, permitindo que mudanças tecnológicas possam ser aproveitadas.

A perspectiva técnica da reengenharia não apresenta uma variação dos requisitos para evolução do sistema de software, mas foi a solução mais apropriada para melhorar a estrutura do sistema e dos dados e para criar uma nova documentação

relacionada, prolongando o tempo de vida útil do software de redução de dados do TA-2, tornando o sistema de medidas de fácil compreensão, como um todo.

5.3 PROCESSO DE APLICAÇÃO DE REENGENHARIA DE SOFTWARE

No desenvolvimento do software de redução de dados adotou-se uma estratégia em que algumas atividades e artefatos gerados usam ações dos processos unificado e cascata que melhores se ajustam ao projeto específico. No entanto, as atividades do modelo de processo em cascata são seguidas com a representação do sistema de software usando uma parte da metodologia estruturada, quando apropriada. Foi planejado um desenvolvimento deste software em paralelo com o de aquisição de dados, e aqueles requisitos com respeito aos dados de saída armazenados pelo novo software de aquisição de dados foram definidos ou elaborados em conjunto, ou seja, entre os desenvolvedores de ambos os softwares.

Razões como a atualização da plataforma e a escassez de pessoal habilitado contribuíram para se optar pela reengenharia do software. A implantação de práticas da METAPLI aplica a reengenharia, em similaridade com os passos preconizados por Sommerville (2003), para obtenção do novo software de redução de dados, visando obter uma versão estruturada e modularizada em linguagem de programação C, dos programas existentes em linguagem BASIC, conforme os passos a seguir:

- Tradução de código fonte.
- Engenharia reversa.
- Melhoria da estrutura do programa.
- Modularização do programa.
- Reengenharia dos dados.

Os passos de reengenharia nesta implantação via METAPLI incluem além dos passos preconizados por Sommerville (2003), as práticas de gerência, voltadas para o

entendimento dos métodos matemáticos e técnicas usadas na execução de ensaios em túnel e suas documentações. Para o entendimento destes métodos e técnicas, seriam documentadas a análise de domínio, a generalização de requisitos e especificações e a determinação dos requisitos, no início do projeto de reengenharia do software de redução de dados. Além dessas atividades estão também incluídas as documentações das reestruturações dos dados e do projeto, em consonância com os requisitos dos subprodutos de software de aquisição de dados, para possibilitar a facilidade na tradução do código fonte e os passos seguintes do processo de reengenharia.

Para executar a tradução do código fonte para a linguagem de programação C, procurou-se identificar os aspectos que induziriam o usuário a erros. Estes itens identificados são problemas para a codificação do novo software, e estes aspectos devem ser eliminados com o uso de técnicas de programação estruturada, provendo um novo código *stand alone* com qualidade. Esses códigos quando na integração software/software, software/hardware e ao sistema de medidas, são avaliados quanto ao cumprimento dos requisitos funcionais e operacionais, implantados pelos especialistas do TA-2 naqueles softwares existentes da época, dentre estes, as funcionalidades das equações e cálculos matemáticos, as funções similares da linguagem e os algoritmos.

Foi planejada a execução dessa tradução de forma manual para possibilitar compreender detalhadamente as funções e operações do software e, quando necessário, exercer as modificações na arquitetura do sistema de software ou de dados. À medida que ia se evoluindo na compreensão do código fonte existente e se progredia na sua tradução, os ajustes e as melhorias do sistema de medidas integrado foram sendo implantados. Dentre estes ajustes está uma definição da sequência dos canais para os dados que seriam reduzidos, que traria melhoria substancial na estrutura do código, diminuindo o tempo de execução do novo software quando integrado.

Para recuperar o projeto e a especificação do software de redução de dados a partir do código fonte, utilizou-se a engenharia reversa. O software foi analisado para

descobrir sua estrutura e adicionadas informações que constam no diagrama de fluxo de dados. Todas as informações obtidas serviram para a geração de documentos, que permitiram aos usuários recriar as especificações do software de redução de dados, antes não totalmente compreendidas.

Também foi possível comparar os dados e resultados obtidos com o novo software da plataforma PC com os obtidos usando a plataforma HP, em similaridade com resultados de ensaios aplicados no TA-2. Além disso, é possível comparar os dados e resultados com ensaios executados em túneis de vento do exterior, quando se validam estes dados e resultados de ensaios executados com o modelo aerodinâmico NACA 0012 (BARLOW, RAE, POPE, 1999).

Com as diversas alterações procedidas nos softwares de redução de dados existentes ao longo de sua utilização antes do processo de modernização, a estrutura lógica de dados e de controle tornou-se complexa, provavelmente devido à adoção de solução rápida que resultou em código incompreensível. Foi prevista a melhoria na estrutura do programa também manualmente, visando prover a geração de um programa melhor estruturado, principalmente sem aquelas declarações “go to” contidas nos programas não estruturados existentes.

A modularização foi planejada de forma manual com a identificação das relações entre os componentes existentes nos programas em linguagem BASIC. A reorganização do programa se daria da seguinte forma: as partes que se relacionavam seriam coletadas e consideradas um único módulo. Foi então identificado que alguns dos softwares de redução de dados não se utilizavam de funções para controlar dispositivos de hardware: nestes, a maioria dos módulos seria agrupada segundo as funções que realizavam tarefas semelhantes ou relacionadas. Contudo, na análise preliminar de tais softwares existentes já se havia deparado com problemas da linguagem BASIC, que apresentava recursos limitados de estruturação dos dados. Nestes códigos havia implementações de diversos vetores compartilhados, tornando

mais complexa a compreensão dos dados, pois um mesmo vetor era utilizado para diferentes tipos de estrutura de dados.

Com a reengenharia dos dados e códigos, a qualidade desses dados deveria ser melhorada, tornando-os mais compreensíveis. As diversas modificações nos dados que geravam novas versões do código poderiam introduzir mais problemas, como por exemplo, erros de valores, valores duplicados, extensão diferente dos campos, denominação dos dados obscuros, ausência de dicionário de dados, entre outros. A reengenharia dos dados deveria ser executada de modo que suas definições, valores literais, formatos e valores de dados fossem alterados de forma criteriosa.

Com a reengenharia do software de redução de dados do TA-2 deveria ser possível obter a especificação dos requisitos e sua documentação pertinente. Assim sendo, depois de aplicada a reengenharia, a descrição destes requisitos deveria estar completa, e também deveria se proceder à revalidação da arquitetura, à implementação e à execução dos testes do software desenvolvido em linguagem C. A qualidade deste software deveria ter melhorias significativas, para que a equipe de ensaios do túnel de vento compreendesse melhor os requisitos, os dados e os códigos obtidos, devido a sua documentação formalizada.

5.4 REQUISITOS DO SISTEMA DE SOFTWARE DO TA-2

Os requisitos de sistema de software ou especificação de requisitos de software devem ser descritos em um documento, que é uma declaração oficial do que é exigido dos desenvolvedores do sistema de software (SOMMERVILLE, 2003). Existem diversas normas que podem ser adotadas que definem os padrões para os documentos de desenvolvimento de software, que neste trabalho foram apenas citadas como recomendação de uso, das quais se adotou a DOD-STD-2167A (1988). Esta norma estabelece que o documento de requisitos se inicie com uma descrição, sendo que no presente trabalho foi inicialmente apresentada no documento como a declaração

abstrata e de alto nível de uma função ou restrição do sistema de software. Estes requisitos normalmente são classificados em:

- Requisitos funcionais.
- Requisitos não funcionais.
- Requisitos de domínio.

Para a segunda iteração da evolução foram estabelecidos os requisitos que se espera dos novos softwares do sistema de medidas. As funcionalidades ou os serviços que se espera que o sistema de medidas forneça foram descritas, anteriormente neste capítulo, como requisitos funcionais. Os fatores de qualidade, como a confiabilidade do sistema de medidas e dos dados, são considerados requisitos não funcionais. Os cálculos, as atuações e as especificidades matemáticas e operacionais, aplicadas em ensaios no túnel de vento são os requisitos de domínio.

Os requisitos de domínio foram levantados nos diversos programas em BASIC existentes para plataforma HP, bem como nos demais documentos e relatórios de ensaios, e foram então implementados e testados nos novos softwares. Devido à complexidade e à quantidade, não são descritos detalhadamente neste trabalho, mas são demonstradas as funcionalidades e operacionalidades implantadas nas representações dos resultados dos softwares.

5.4.1 Requisitos gerais do sistema de medidas

Os requisitos gerais do sistema de medidas foram definidos como sendo os fundamentais para a viabilidade do projeto de software, que são:

- Funcionar em plataforma PC.
- Operar em um ambiente Microsoft Windows[®].

- Utilizar as ferramentas propostas ou recomendadas ou convencionadas pelos envolvidos no projeto de modernização do túnel de vento.
- Utilizar hardware e dispositivos COTS para aquisição de dados, preferencialmente aqueles recomendados da fabricante *National Instruments* (NATIONAL INSTRUMENTS, 1998) ou especificar outros necessários.
- Ter a capacidade de abranger elementos de medidas e de controle com disponibilidade de dados para todos os subsistemas envolvidos.
- Executar todas as medidas dos sensores especificados para os ensaios.
- Abranger ou executar os controles especificados para os ensaios.
- Utilizar um vocabulário comum do TA-2 para as informações do sistema de medidas.
- Ter capacidade de validação de dados inseridos pelo operador.
- Ter capacidade de validação de dados obtidos dos sensores.
- Ter capacidade de atribuir e selecionar as permissões de acesso para os operadores.
- Possibilitar condições de uso de aplicativos para análise dos dados e validação das funcionalidades dos subsistemas integrados ao sistema de medidas.
- Possibilitar a calibração dos instrumentos de medidas para o ambiente de ensaio.

5.4.2 Requisitos de interface homem-máquina

A interface com usuário ou interface homem-máquina tem uma natureza dinâmica que necessita de requisitos específicos que as descrições textuais e os diagramas não são suficientes. O processo de projeto de interface com usuário neste trabalho envolveu o projeto de telas gráficas com a ferramenta LabView, em que componentes como menus, botões, campos e ícones estão posicionados em uma interface. Shneiderman (1998) apresenta uma série de sistemas de projeto de telas

gráficas ou projeto de interface em que o usuário tem participação ativa na avaliação destas interfaces.

As interfaces deste trabalho são avaliadas a cada projeto de subproduto de software e todos os membros ativos do grupo de projeto participam desta avaliação. Os requisitos de interface homem-máquina foram definidos baseados em funções do sistema de medidas, as quais precisam ser acionadas e monitoradas pelo operador dos ensaios em túnel de vento.

- Consistir de janelas alfanuméricas e gráficas (ambiente padrão Windows[®]).
- Possibilitar ao operador o uso do teclado alfanumérico e do mouse.
- Prover o uso das funções mais freqüentes com acesso direto para o operador.
- Possibilitar que cada ação do operador tenha um efeito visível na tela para que o operador possa confirmar sua ação de aceitação ou rejeição no software.
- Prover as unidades e formatos configuráveis pelo operador.

Assim, a engenharia do software de aquisição de dados com a ferramenta *LabView*, deve permitir desenvolver uma série de instrumentos virtuais de medição, os chamados módulos ou “vi”. Também através de biblioteca de instrumentos deve prover a reutilização em outros “vi”, mais complexos. Na criação dos “vi” do novo software de aquisição de dados, as capacidades funcionais requeridas para aquisição e monitoramento dos dados, gravando os resultados em arquivos, para posterior análise e redução dos dados são requeridas.

Foram definidas algumas evoluções do sistema de medidas do TA-2 na segunda iteração da fase de elaboração, e descritos alguns componentes que fariam parte da construção de subprodutos de software. Consistem de dispositivos DAQ, que abrangem também os *Signal Conditioning eXtensions for Instrumentation* - SCXI, utilizados para aquisição de dados e algumas análises, conforme mostra a figura 5.14.



Figura 5.14 - Componentes do sistema DAQ e SCXI para aquisição de dados (NATIONAL INSTRUMENTS, 1998).

A necessidade desta evolução foi constatada na primeira iteração da fase de elaboração com o aumento do domínio do conhecimento da nova tecnologia, provida pelo ambiente de programação gráfica visual. Constatou-se que os instrumentos da cadeia de medidas poderiam ser substituídos pelos instrumentos virtuais, sem prejuízo para as funcionalidades e operacionalidade. Ficou definido que este trabalho de substituição seria desempenhado pelos parceiros em conjunto com os desenvolvedores de software do TA-2.

Qualquer uso de outros dispositivos ou recursos na evolução dos subprodutos de software, ou de outras ferramentas como *Excel* da *Microsoft*, *ARENA* da *ROCKWELL*, *C++ Builder 4* da *National Instruments*, *LabWindows/CVI* da *National Instruments* (NATIONAL INSTRUMENTS, 1998) ou outros aplicativos similares, devem ser documentados e suas construções abordadas, com os benefícios apontados e providos para a qualidade do sistema de software ou dos dados.

5.4.3 Requisitos de interface de comunicação computacional

A aplicação de uma interface deve ser feita de modo a prover o uso cooperativo de informações, entre as instrumentações e dispositivos eletrônicos empregados para o sistema de medidas. Estas serão compartilhadas entre os diversos subsistemas ou

ambientes de engenharia de sistema com a tendência de transmissão de dados, através de linhas comerciais, que estão cada vez mais sendo aplicadas.

A escolha da interface necessária para comunicação entre computadores, instrumentações e dispositivos eletrônicos dependerá dos tipos de comunicações disponíveis nos aplicativos ou ferramentas de apoio ao desenvolvimento de software que tenham funções computacionais, ou de soluções computacionais em conjunto com placas eletrônicas, cablagem e ferramentas, que convertam os sinais e provejam a comunicação entre os dispositivos e instrumentações especificados para uso na plataforma PC.

Essas escolhas devem apresentar as seguintes vantagens com uso de interface ou barramento que são:

- Facilidade de interfaceamento.
- Alta velocidade de comunicação.
- Acesso múltiplo e formato para comando/resposta dos dados.
- Boa capacidade do controlador de barramento para controlar, de forma programada, o fluxo de dados de um local para outro.
- programação sincronizada de transferência de dados que assegure que a informação chegue no local e no momento em que ela está sendo requerida, de forma absolutamente controlada.

Os tipos de barramentos de dados e protocolos seriais e paralelos mais utilizados estão disponíveis pela fabricante *National Instruments* (NATIONAL INSTRUMENTS, 1998) através dos chamados *drivers*, que são softwares contendo funções ou os “vi’s” de protocolo, necessários para a comunicação com determinados instrumentos de medida ou de controle, que analisados previamente podem oferecer soluções viáveis com computadores PC, totalmente compatíveis com ambiente operacional padrão *Windows*, que são as seguintes:

- Paralela, GPIB (*General Purpose Interface Bus*) ou similar ao HP-IB da plataforma HP, com protocolo de comunicação de software IEEE 488.2.
- Serial, ou protocolo comum para comunicação e controle de dispositivos, podendo escolher entre RS-232 ou RS-422 ou RS-485.
- Paralela, com protocolo de comunicação IEEE-1284.

Visando a aplicação com algum reuso no desenvolvimento do software de aquisição de dados do TA-2, foram adquiridas as bibliotecas com “drivers” julgadas adequadas, disponíveis pelo fabricante *National Instruments* (NATIONAL INSTRUMENTS, 1998), para aquisição de dados com medidas e controle, usando interface serial e paralela, inclusive a GPIB para instrumentos convencionais da HP.

Na segunda iteração da fase de elaboração planejou-se adotar uma tecnologia que oferecesse melhorias para sistema de medidas, referentes ao acionamento e monitoramento do motor e hélice. Então, foram avaliados os recursos da *PCI eXtensions for Instrumentation - PXI*, também oferecidos em dispositivos eletrônicos da *National Instruments*. Estes recursos deveriam substituir as funções providas pelo equipamento COTS e código “caixa preta”, do sistema de medidas antigo. Para atender estas funcionalidades, os recursos deveriam ser voltados para sistemas de tempo real com alta capacidade de processamento e leitura de dados de sensores, razão pela qual se efetivou o uso dos recursos PXI.

Os requisitos de interface homem-máquina para os subprodutos de software com funções de controle deveriam comportar na tela do painel frontal do *LabView*, formas visuais de todos os controles, gráficos e indicadores, informando no *display* do software de aquisição de dados a simulação de um painel físico para controle e monitoramento dos dispositivos e instrumentações interligados ao computador PC. A maioria destes controles é formada por botões, *leds*, *knobs* e indicadores que permitem a interação com o operador através do mouse ou do teclado do computador PC.

5.5 AMBIENTES DE ENGENHARIA PARA MEDIDAS E CONTROLE DO TA-2

Ao longo do desenvolvimento do sistema de software do TA-2, novas soluções de recursos de instrumentos de medidas foram sendo implementadas, com a preocupação de responder as necessidades de manter operacional o TA-2, para os ensaios programados.

Em muitas situações, por exemplo, no ensaio de partes aerodinâmicas da aeronave EMB-170, era necessário obter as medidas com as *scanivalves* e alguns outros instrumentos. Nestas circunstâncias, os ambientes de engenharia, de software e gerenciais foram preparados, para obter as medidas e exercer os controles dos subsistemas que resolveriam as questões da situação requerida, com a seleção de dispositivos e desenvolvimento de subprodutos de software compatíveis com os experimentos programados. Esses ambientes com relação às ferramentas de apoio ao desenvolvimento do ambiente de engenharia de software constam das tabelas 5.1 e 5.2.

Para a seleção das ferramentas que ajustassem ao desenvolvimento do software de aquisição de dados, buscou-se escolher aquelas de fácil utilização e mais consistentes para serem integradas. A maioria das ferramentas de apoio ao desenvolvimento utilizadas neste trabalho foi indicada sob consenso dos especialistas do TA-2 e seus parceiros. Apenas a ferramenta *LabView* da *National Instruments* para desenvolvimento do software de aquisição de dados foi designada pela equipe de especialistas do TA-2, devido às atividades previstas pelo projeto de modernização.

Tabela 5.1 - Ambientes operacionais

Plataforma Computacional	Sistema Operacional
Plataforma HP: <i>workstation</i> HP 9000 série 300 de 16 bits	UNIX-4.4 BSD -1992
Plataforma PC: <i>Pentium</i> 133 Mhz	Windows 95/98/NT/2000/XP

Tabela 5.2 - Ferramentas do ambiente de engenharia de software

Atividade Engenharia de Software	Ferramentas
Requisitos	Microsoft ® Word Microsoft ® Power Point Adobe Reader
Projeto	Microsoft ® Word Microsoft ® Excel
Codificação para Plataforma HP	HP BASIC 5.1
Codificação para Plataforma PC	LabView versões 3.1, 4.1, 7.0 e 7.1 C++ Builder 4
Codificação para Análise de dados	Microsoft ® Excel ARENA
Codificação para Interface com usuário	NI 488.2 NI-DAQ Wizard
Teste	Microsoft ® Word Microsoft ® Excel
Gerenciamento de Configuração	Microsoft ® Word
Documentação	Microsoft ® Word

Foi levantada a forma como a equipe conduzia o trabalho dos elementos envolvidos na calibração, aquisição e redução dos dados, para ser possível compreender a contribuição de cada fator, visando a diminuição dos problemas na implantação de um novo processo de obtenção do produto final de software. A finalidade deste levantamento é demonstrar e discutir os problemas com a equipe, para que no decorrer do desenvolvimento do novo sistema, esses pudessem ser dirimidos.

As ferramentas novas apontadas para o desenvolvimento de software de aquisição de dados, para a época, eram consideradas relativamente recentes. Para seu uso correto, havia a necessidade de consultar com frequência uma grande quantidade de manuais de usuário e de operação, devido à diversidade de dispositivos eletrônicos

a ser implantados no sistema de medidas e a pouca habilidade tanto com a ferramenta quanto com a integração hardware/software. Nas demais ferramentas de suporte já havia certo domínio e essa transição não consumiu muito tempo quando foram utilizadas.

Muitas dessas ferramentas foram decididas antes da segunda iteração da fase de elaboração, para suprir as necessidades com relação a plataforma HP, visando garantir o armazenamento dos dados na plataforma PC. Um exemplo é a ferramenta C++ Builder 4, utilizada para elaboração de um subproduto de software que, mesmo antes de obter os primeiros subprodutos do softwares de aquisição de dados do PC, deveria ser criado e validado para transferir dados dos arquivos dos ensaios realizados no TA-2, com a plataforma HP para arquivos de dados que pudessem ser armazenados na plataforma PC, facilitando assim a análise destes pelos especialistas e os testes a serem empregados no desenvolvimento do novo software de redução de dados.

Neste trabalho por questões de inviabilidade não se fez uso de ferramenta CASE para documentação do projeto de software, a maioria da documentação foi gerada com uso da ferramenta *Microsoft Word*. Foram elaborados alguns guias e procedimentos necessários para representar o desenvolvimento do software, também fazendo uso desta ferramenta. Esses planos e manuais foram confeccionados usando inclusive os recursos gráficos quando apropriados, que depois de finalizados, por questões de segurança da informação, foram convertidos para o formato seguro pela ferramenta *Acrobat Reader* da *Adobe* e disponibilizados em rede *Ethernet*, para a consulta dos envolvidos.

O software de aquisição de dados é responsável pelo controle do sistema de medidas, permitindo ao operador inserir parâmetros, comandar e monitorar os dados, durante a aquisição. Esses dados são armazenados em forma de arquivos que podem ser utilizados e consultados posteriormente. A forma mais comum de armazenagem são os *Spreadsheets* ou Planilhas de Cálculos, fato que possibilitou o uso da ferramenta *Microsoft Excel* para armazenar e analisar esses dados na plataforma PC.

As funções básicas dos *spreadsheets* foram também utilizadas para guardar e apresentar as informações das aquisições transferidas da Plataforma HP para a plataforma PC. Para essa transferência de dados, foi elaborado um software usando a ferramenta *C++ Builder 4* visando suprir as necessidades dos especialistas do TA-2 tendo em vista a degradação do equipamento HP-9000/300, no qual estavam armazenados todos os dados adquiridos.

Para os ambientes de desenvolvimento foram providos e usados vários dispositivos, alguns destes definidos para deixá-los à disposição, visando equipar o TA-2 para medir, monitorar e controlar os parâmetros aerodinâmicos necessários, para cada ensaio programado. As provisões dos equipamentos do sistema final de aquisição de dados compreendem um sistema da *National Instruments* composto, por exemplo, de dois módulos de controle modelo NI PXI 1045 e *Rack* de instrumentação SCXI, módulo de controle com controlador interno ao gabinete modelo PXI-8186RT (microcomputador com processador Intel Pentium 4 de 2.2 GHz). Estas provisões destinam-se ao controle do sistema e registro dos seus parâmetros e a aquisição dos dados de ensaio.

Este sistema de controle pode adquirir até 64 parâmetros do sistema com conversão de 12 bits e taxa de aquisição de 1,25 mega *samples* / segundo e 8 canais de sinal RMS de 6 ½ dígitos para monitoração da vibração do sistema propulsor. Pode ainda atuar no sistema por 32 canais de estímulo de corrente e tensão. Através de software de *driver* que são as bibliotecas do *LabView* para reuso, e das placas de aquisição de sinais analógicos e digitais, como por exemplo, dispositivos *DAQs* e placas SCXI, os sinais são multiplexados, amplificados e convertidos, para serem armazenados pelo software de aquisição de dados.

Esses dispositivos *DAQs* e placas SCXI da *National Instruments*, juntamente com os softwares de *driver* pertinentes, os dispositivos GPIB e de controle como as placas PXI, foram sendo implementados e testados a cada iteração e incremento, gerando os chamados *build* ou subprodutos da fase de construção, conforme preconiza

o processo unificado. Cada *build* engloba aqueles elementos necessários, representados através das práticas implantadas da METAPLI, para obtenção dos subprodutos do software de aquisição de dados, para serem integrados ao sistema de medidas, os quais foram validados tanto no nível de hardware como de software.

Na segunda iteração da elaboração os dispositivos eletrônicos foram definidos e constam do Apêndice B deste trabalho. Foram previstas também as provisões de dispositivos para a segunda e terceira iteração da fase de construção dos subprodutos de software de aquisição de dados. Assim como provisões para quarta e última iteração da fase de construção do subproduto final do sistema de medidas, que também constam do Apêndice em questão.

5.6 PROCESSO DE ANÁLISE DE DESEMPENHO DOS DADOS

Através de recursos computacionais complementares, o trabalho desenvolvido provê uma análise de desempenho, a qual é feita a partir de comparações de dados. Embasado em dados de ensaios anteriores com a plataforma HP, avaliam-se as ações do operador com os novos subprodutos de software e os resultados dos dados de ensaio. Para medir o desempenho neste trabalho, se definiu o processo de calibração de forças e momentos, sendo este processo com maior volume de dados.

Os dados iniciais são os de calibração das células de carga, obtidos com um dos subprodutos de software deste trabalho, dispositivos e instrumentos planejados, para o ambiente de engenharia do laboratório de força. Os dados são os chamados coeficientes de calibração, que são armazenados para serem lidos por outro subproduto de software de aquisição de dados da cadeia de medidas e também pelo software de redução de dados, caso os dados da calibração no ambiente de ensaio do TA-2 sejam similares.

A calibração no ambiente de ensaio do TA-2 é executada com a cruz de calibração e os diversos carregamentos em Alfa ou Beta, na balança externa sob a qual está instalada a cruz de calibração, sendo usadas as mesmas células de carga calibradas no laboratório de força. Os resultados de ambas as calibrações são comparados visualmente pelo operador e através de cálculos processados pelo subproduto de aquisição de dados. Os resultados desses coeficientes devem ser similares, podendo o operador optar pelos coeficientes do laboratório, também atribuídos como coeficientes do fabricante.

Outra análise de desempenho dos dados é feita neste trabalho, através de análises estatísticas executadas após a obtenção dos dados com o subproduto de aquisição, análise esta que se descreveu como um novo processo gerencial para a qualidade denominado pré-redução de dados. Para esta fez-se uso do aplicativo ARENA da ROCKWELL e baseou-se num artigo sobre produtividade em túnel de vento (GASPERS, MULHLSTEIN, 1998), no qual os autores descrevem que os dados de força e de momento de um túnel de vento devem apresentar uma distribuição normal.

Executou-se a análise estatística para verificar o nível de fidelidade dos dados de força e momento em relação a distribuição normal, usando os recursos do aplicativo comercial. Os modelos analíticos aplicados aos dados disponíveis dos ensaios do TA-2 permitiram a disponibilidade e obtenção de informação de pré-redução, cobrindo uma série de deficiências na avaliação desses dados, antes que fossem submetidos ao subsistema de redução de dados.

5.7 REALINHAMENTO DE DADOS E PROCESSOS

A característica da METAPLI é fornecer uma maneira de proceder, de forma que à medida que os projetos dos componentes de hardware e dos subprodutos de software foram sendo definidos, concebidos e avaliados, as ações de migração de um ambiente

funcional para outro prosseguiam. Em outras palavras, somente depois de atingidos os aspectos de desempenho, confiabilidade, facilidade de programação e provisão para futuras evoluções do sistema de software, dava-se continuidade ao projeto do sistema de medidas através das iterações e incrementos.

O realinhamento dos dados e processos de engenharia somente se tornaria possível após algumas iterações, incrementos e reestruturações para obter os subprodutos de software. No entanto, as práticas implantadas via METAPLI também visam a melhoria dos processos gerenciais, para dirimir os riscos e garantir a continuidade do projeto de software, em seu conteúdo proposto. Essas práticas correspondem ao registro de problemas e propostas de modificações, tanto para os processos técnicos como para os processos gerenciais, que são conduzidas durante o ciclo de vida do projeto de software.

Melhor dizendo, à medida que os subprodutos dos softwares foram sendo desenvolvidos e os requisitos foram sendo validados, foram sendo ajustados os dados e processos, e para processos gerenciais acrescidos os requisitos suplementares. A especificação de requisitos técnicos e gerenciais do sistema de medidas e sua documentação seguiram parte da metodologia tradicional, que deve iniciar com uma visão clara do sistema e subsistemas a ser concebido. Também seguiu parte da metodologia ágil que deve definir e analisar os requisitos gerenciais, ou seja, os processos, as práticas, os métodos e as técnicas.

O documento de especificação de requisitos de software foi sendo complementado na medida em que se detalhavam as partes dos subsistemas definidos no documento de visão do sistema de medidas. Esta forma de trabalho corresponde à realização de atividades do desenvolvimento de software nas épocas em que elas são efetivamente necessárias, conforme a visibilidade que se tem sobre o processo de desenvolvimento dos subprodutos de software.

Este processo de refinamento traz, como aspecto positivo, a rapidez com que a versão operacional do subproduto de software pode efetivamente ficar pronta. Também em termos gerenciais concentra os esforços no detalhamento dos requisitos de qualidade e de engenharia, tornando o sistema de medidas melhor implementado, testado e acompanhado, completando as capacidades operacionais com o decorrer do desenvolvimento, acrescentando aspectos de qualidade e de engenharia requeridos, que não foram previstos originalmente.

5.7.1 Gerência de requisitos variáveis

Os requisitos de software podem variar devido a mudanças que podem ocorrer no projeto de software de aquisição de dados. Devido à complexidade demonstrada por seus subsistemas, pode ocorrer evolução das ferramentas e dos equipamentos eletrônicos, que acarretem mudanças no software de aquisição de dados para incorporar os requisitos pertinentes, até que se chegue no produto de software final.

O desenvolvimento do software de aquisição de dados é feito conjuntamente com parceiros, e algumas vezes, em locais físicos separados com execução de trabalhos progressivos, podendo ocorrer versões múltiplas de subprodutos de software, havendo a necessidade de gerenciar as mudanças de requisitos e ambiente de engenharia, para manter sob controle as versões e configurações, e assim garantir a qualidade dos subprodutos e do produto final.

Alguns diagramas propostos na implantação da METAPLI foram adequados para gerenciar os requisitos variáveis. Os documentos e planos do projeto do novo sistema de medidas provêm a gestão dos subprodutos de software, dos componentes de hardware e a engenharia, como requisitos de cada fase do ciclo de vida do desenvolvimento de software. Os detalhes técnicos e gerenciais do desenvolvimento de software podem ser vistos, nos capítulos seguintes deste trabalho.

CAPÍTULO 6 IMPLANTAÇÃO E RESULTADOS DA METAPLI

Para obter resultados aceitáveis e qualidade no projeto de software aplicado em ensaios em túnel de vento se implantou a METAPLI. Esta implantação é desempenhada em uma série de processos contínuos, que abrange o mapeamento das práticas utilizadas para a documentação gerencial e técnica, práticas estas aplicadas na modernização do sistema de medidas. Como o sistema antigo possuía alguns processos documentados e outros não documentados do sistema de medidas e seus softwares, desde o início desta implantação procuraram-se maneiras de documentar as atividades do ciclo de vida de desenvolvimento.

A implantação via METAPLI das atividades das fases do processo de desenvolvimento, com as ações selecionadas e práticas pertinentes estão detalhadas neste capítulo. Esta implantação abrange as aplicações das atividades de informação, planejamento, desenvolvimento, entregas e aceitações freqüentes, testes e validações de resultados, demonstrações e qualificações dos clientes e também parte da gestão do conhecimento. Fazem parte também deste trabalho, as lições aprendidas e o realinhamento dos dados e processos, sendo que todas as atividades foram modeladas para aplicações práticas no projeto do sistema de software do TA-2, podendo ser estendidas para outros projetos de software.

As modelagens descritas neste trabalho são em sua maioria aplicáveis a projetos de sistema de software de organizações governamental ou privada. Elas ajudam a alinhar o conhecimento dos envolvidos quanto à maneira de entender e administrar as diferentes funcionalidades do projeto de sistema de software. Também auxilia na detecção dos requisitos funcionais e não funcionais que servem de subsídios para o projeto de arquitetura, implementação, teste e implantação do software a ser produzido. No caso deste trabalho, as modelagens auxiliam na documentação do projeto dos softwares de aquisição e redução de dados, e conseqüentemente na documentação do sistema de medidas do TA-2, completamente renovado.

A maneira como foram desenvolvidos esses softwares, o ciclo de vida, as ferramentas, as modelagens e as técnicas, constam das práticas da METAPLI. Praticamente a grande maioria do processo de desenvolvimento conta com representações e modelagem das atividades de cada fase, as quais são apresentadas neste capítulo. Também são apresentados os resultados dos dados processados pelos novos softwares de aquisição e de redução de dados, comparados aos obtidos com a antiga plataforma HP.

6.1 DOCUMENTAÇÃO GERENCIAL VIA METAPLI

A documentação gerencial via METAPLI foi gerada, adotando-se os recursos da ferramenta Microsoft Office. Esta escolha foi feita pelos especialistas em ensaios em túnel devido à facilidade de aquisição e por questões econômicas, para gerar os documentos julgados adequados da norma DOD-STD-2167A (1988), e a grande maioria dos documentos gerenciais voltados aos ensaios aerodinâmicos. As atividades da metodologia e dos processos foram implantadas sem utilizar qualquer ferramenta CASE disponível comercialmente que pudesse sustentar a geração de documentação de desenvolvimento dos softwares de aquisição e de redução de dados.

As modelagens gerenciais e técnicas dos processos ou atividades são representadas em documentos como planos, listas, procedimentos, instruções de trabalho, diretrizes entre outros, os quais contêm diagramas, fluxos de trabalho e fluxogramas explicativos. Essa modelagem abrange a engenharia de softwares experimentais, subprodutos de softwares para o sistema de medidas e de softwares finais da modernização do TA-2, engenharia esta representada de uma forma geral na Figura 6.1, correspondente aos modelos e artefatos do sistema de software.

O artefato inicial é o plano de avaliação que foi gerado para averiguar a situação e os riscos técnicos e gerenciais inerentes ao projeto de software, para a implantação de metodologia e de processos de engenharia de software formais. Com a necessidade de modernização do TA-2, esbarrou-se na ausência de padronização e de gerência das atividades de engenharia de software, com a abrangência de planejamento e de processos de projeto do sistema de software. Então em função disto elaborou-se o plano de implantação de metodologia e processos, contendo a modelagem gerencial.

No escopo deste plano estão incluídos a aquisição, o desenvolvimento e modificações significantes de software, para uso ou suporte aos ensaios aerodinâmicos. O objetivo é preparar e implantar práticas comuns que incorporem atividades integradas para a documentação técnica e gerencial, consolidadas pelos envolvidos com o uso no projeto de desenvolvimento dos novos softwares de aquisição e de redução de dados. Este se tornou um dos artefatos importante para a engenharia dos softwares, aplicado no desenvolvimento do novo sistema de medidas do TA-2.

Outro artefato gerado que contribui para o entendimento do sistema de software é o documento de visão, que aborda o sistema de medidas do TA-2 e seus subsistemas, no qual é descrita uma visão deste sistema para o planejamento, gerenciamento e implantação do projeto dos softwares de aquisição e redução de dados, com enfoque nas características essenciais do sistema e nos níveis aceitáveis de qualidade. Esta visão focaliza as práticas e padrões de engenharia aplicados nos ensaios aerodinâmicos, visando informar e manter relações com os parceiros sobre estes enfoques de engenharia.

Após gerar os documentos de visão e de implantação de metodologia e processos, obtêm-se informações importantes sobre o sistema de medidas, e os conceitos de engenharia de software aplicáveis ao projeto do sistema de software. Então, esses conceitos são tratados formalmente no plano de desenvolvimento de software, do qual se originam os requisitos iniciais descritos nos documentos de

especificação de requisitos de segmento/sistema e de interface, de projeto de segmento/sistema e de descrição de projeto e de interface, entre outros documentos, segundo a norma DOD-STD-2167A (1988).

Como resultado da implantação da metodologia e de processos, foi elaborado um processo interno para a especificação de software. Este é descrito de forma resumida como uma parte da instrução de trabalho usada internamente no laboratório de túnel de vento. O conteúdo informativo desta instrução de trabalho irá compor o caso de desenvolvimento. Essa instrução de trabalho contém o seguinte:

1. Ao identificar a necessidade de desenvolvimento de software, qualquer indivíduo do TA-2 pode gerar uma especificação de software, devendo comunicar a sua proposta de desenvolvimento à chefia da organização para estudo de viabilidade.
2. A especificação deve conter no mínimo a identificação do item de software a ser desenvolvido, a descrição da visão geral do sistema, subsistemas, a descrição de interfaces e documentos pertinentes. Também deve conter a forma de organizar os documentos pertinentes, os recursos humanos e materiais com suas capacidades, os dados, a adequação e segurança, os métodos de qualificação e a determinação dos elementos complementares para o glossário.
3. Recomenda-se o uso das modelagens gerenciais e técnicas para descrever os detalhes conhecidos da especificação de software.

A especificação de software é muito utilizada para estabelecer seus requisitos iniciais, visando esclarecer melhor as informações e possibilitar seu desenvolvimento. Muitas destas especificações contêm as representações consideradas mais explicativas dos processos de ensaios do TA-2. As informações contidas numa especificação de

software serão complementadas no decorrer do desenvolvimento, e gerados os artefatos como especificação de requisitos de software e suplementar, entre outros.

As informações apresentadas nos documentos internos e gerados usando o preconizado na norma, consistem de contexto, escopo, capacidades específicas e características dos diferentes tipos de software e cenários operacionais, que visam garantir a compreensão dos envolvidos com respeito ao novo sistema de software, podendo ser utilizadas em projetos futuros.

Os modelos definidos na figura 6.1 serão detalhados neste capítulo, bem como alguns artefatos interligados, pertinentes ao ciclo de vida de desenvolvimento. O modelo gerencial e a aplicação das atividades de informação e de planejamento da METAPLI, em sua forma detalhada, são definidos através de representações gráficas como o diagrama da tartaruga, os fluxogramas, fluxos de trabalho e outros recursos, que se proponham a resguardar as atividades de desenvolvimento e suas informações relevantes.

6.2 RESULTADO DA MODELAGEM GERENCIAL

Uma das maneiras adotadas para a modelagem gerencial é a utilização do diagrama da tartaruga. Este esclarece o processo de desenvolvimento dos softwares de forma diagramática, apropriado para uso da equipe de qualidade de software. Neste diagrama são detalhados os processos do sistema de medidas renovado, no qual estão descritos os elementos usados para comprovar a qualidade deste projeto de sistema de software como um todo. Para cada processo técnico do desenvolvimento de software foi modelado um diagrama da tartaruga, com as representações das atividades gerenciais aplicáveis, mostradas nas figuras 6.2, 6.3 e 6.4.

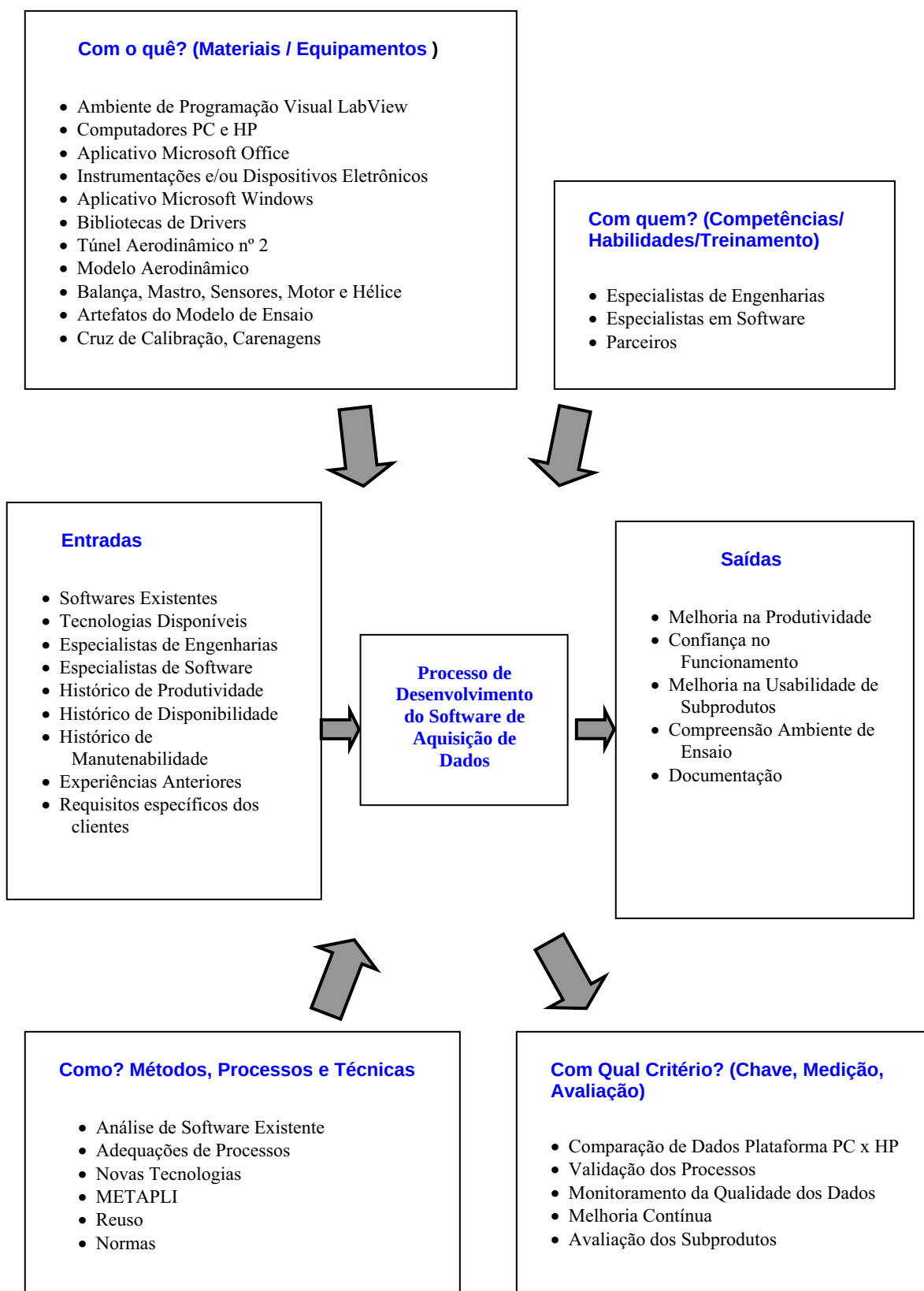


Figura 6.2 - Diagrama da tartaruga do processo de desenvolvimento do software de aquisição de dados

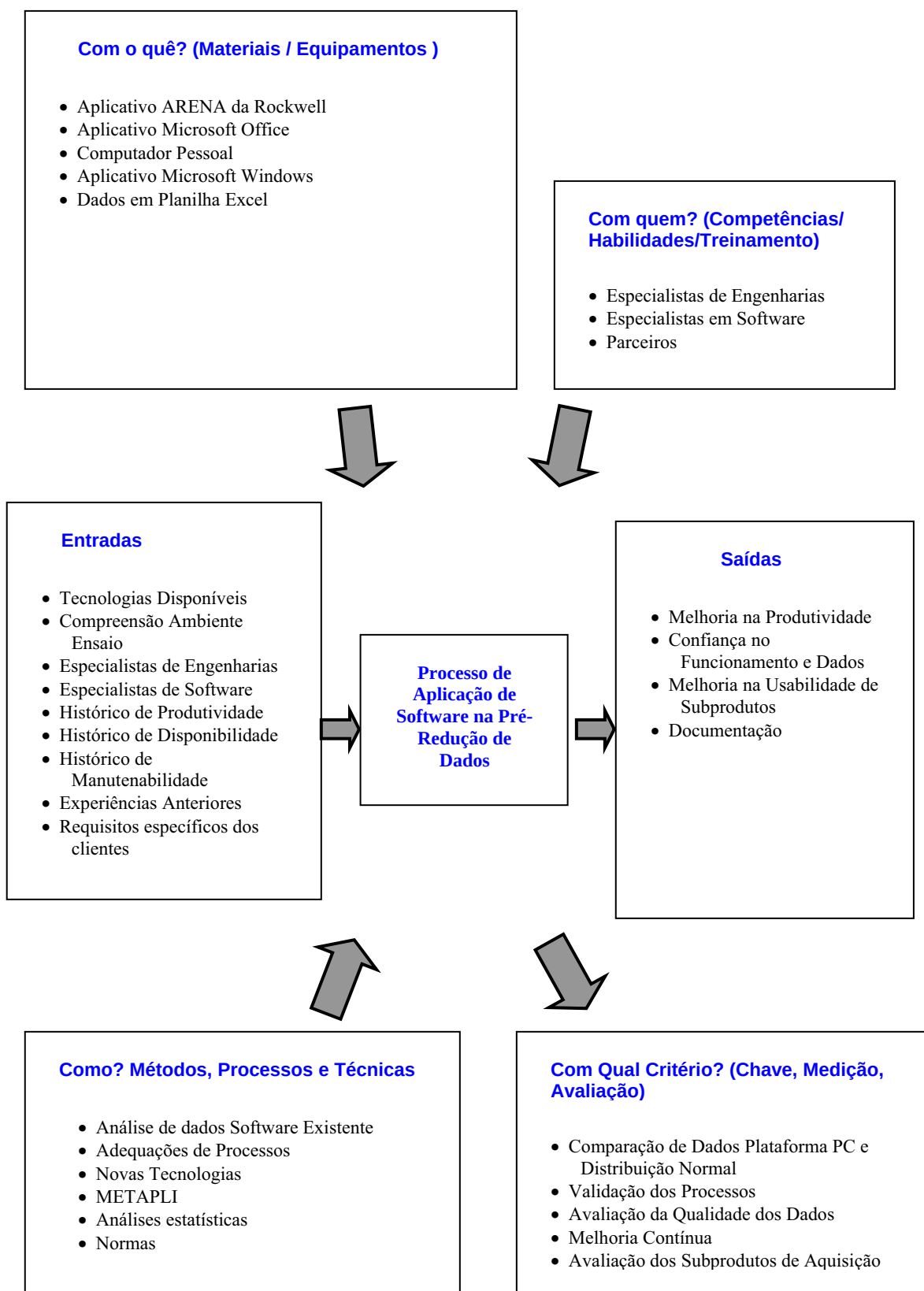


Figura 6.3 - Diagrama da tartaruga do processo de aplicação de software na pré-redução de dados

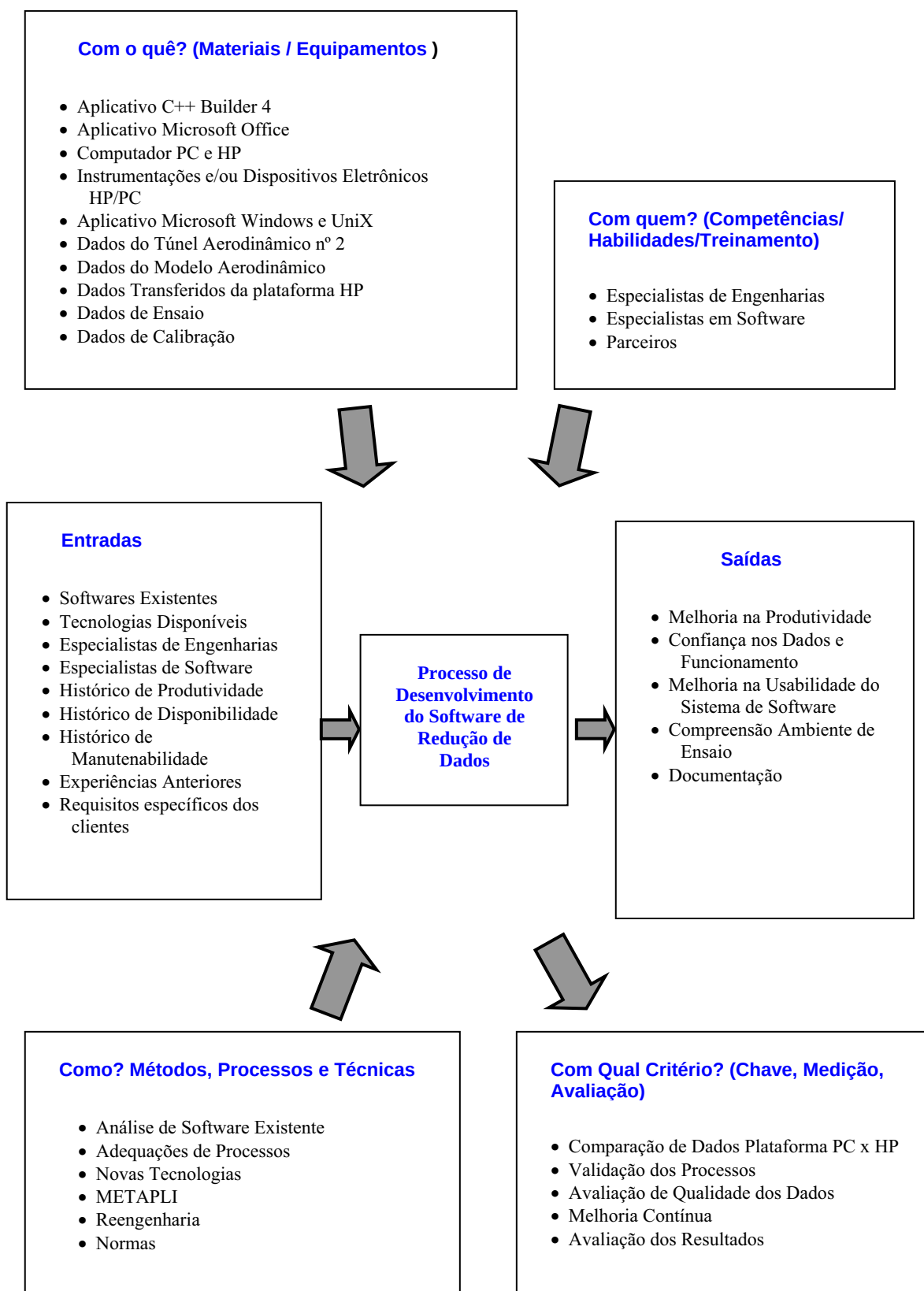


Figura 6.4 - Diagrama da tartaruga do processo de desenvolvimento do software de redução de dados

Os diagramas da tartaruga mostram os elementos do processo de desenvolvimento dos softwares de aquisição de dados e de redução de dados, e também do processo de aplicação de software na pré-redução de dados. Este processo da figura 6.3 consiste na aplicação de análise de dados usando um aplicativo COTS, e portanto, não corresponde a um software desenvolvido, mas, devido à sua importância na avaliação dos dados do TA-2, foi considerado um processo a ser gerenciado.

Como mostrado nos diagramas da tartaruga, as informações e os recursos necessários para o desenvolvimento de cada software constam das entradas, materiais/equipamentos, métodos, processos e técnicas, e competências/habilidades e treinamento. Esta modelagem gerencial demonstra uma intensa necessidade dos especialistas alocados nos ensaios em túnel e dos dados do sistema existente, sem os quais, o conhecimento e o entendimento do sistema de medidas seria ainda mais dificultado.

Cada recurso do projeto do sistema de software está alocado, baseado numa abstração das necessidades dos processos de desenvolvimento de software. Os recursos alocados nos diagramas da tartaruga foram avaliados no desenvolvimento desses softwares, quanto ao quesito qualidade, ao qual cada processo de desenvolvimento deve corresponder com o atendimento aos critérios de avaliação e medição, e também às saídas dos diagramas referentes a esses processos.

Estes diagramas demonstram o envolvimento de parceiros com conhecimento de outros desenvolvimentos de software, visando aprimorar a qualidade dos softwares, o conhecimento e o entendimento do sistema de medidas do TA-2, e renová-lo em conjunto com os especialistas, para que as documentações produzidas possam servir de base para outras evoluções ou produções de sistema de software. A modelagem gerencial adotada com representações das atividades, papéis e artefatos auxiliam esta equipe na documentação do sistema de software. Assim, é papel do engenheiro de software prover o planejamento e o gerenciamento do processo de engenharia dos

softwares de aquisição e de redução de dados. As atividades e os artefatos sob responsabilidade deste papel são mostrados na figura 6.5.

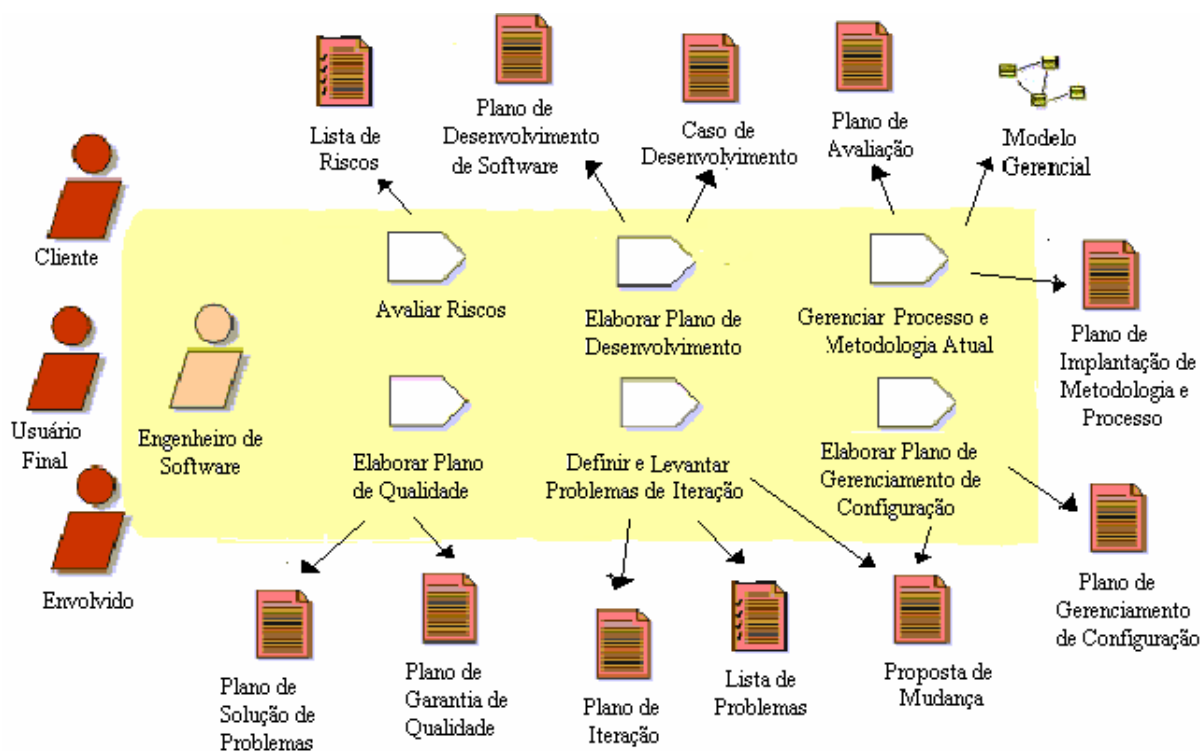


Figura 6.5 - Modelagem gerencial do papel de engenheiro de software

O trabalhador identificado por engenheiro de software é responsável por diversas atividades referentes ao projeto de desenvolvimento de um novo sistema de software, dentre elas o planejamento, a engenharia e as modelagens construtivas e evolutivas do sistema de medidas do TA-2. Esta identidade é responsável pelo artefato “caso de desenvolvimento”, no qual são descritos os processos de desenvolvimento, capturados e adaptados para o projeto do sistema de software, que são usados como fonte de informações essenciais para as atividades de requisitos, análise, projeto e teste do sistema de software, o qual apresenta o comportamento deste projeto de sistema específico.

O engenheiro de software propõe a abrangência dos processos de desenvolvimento para o sistema de medidas, e assegura que cada parte do software

será concluída e refinada pela equipe de desenvolvimento. As atividades de reuso e de reengenharia aplicadas no desenvolvimento dos softwares de aquisição e de redução de dados, são atividades detalhadas pelo papel de engenheiro de software, bem como os planejamentos de revisões e/ou iterações propostas no ciclo de vida de reuso e de reengenharia, demonstradas na figura 6.6.

No ciclo de vida de desenvolvimento com ações do processo unificado foram atribuídos pontos de controle conhecidos como iterações, que também podem ser os pontos de controle de revisões usados pela norma de documentação. Neste trabalho, são mantidas as fases ou atividades dos ciclos de vida, pertinentes a cada processo de desenvolvimento escolhido, seja unificado ou tradicional. No entanto, dado o enfoque em reuso e em reengenharia, foram adequadas e acrescentadas algumas atividades, contemplando as apresentações das revisões, mostradas na figura 6.6.

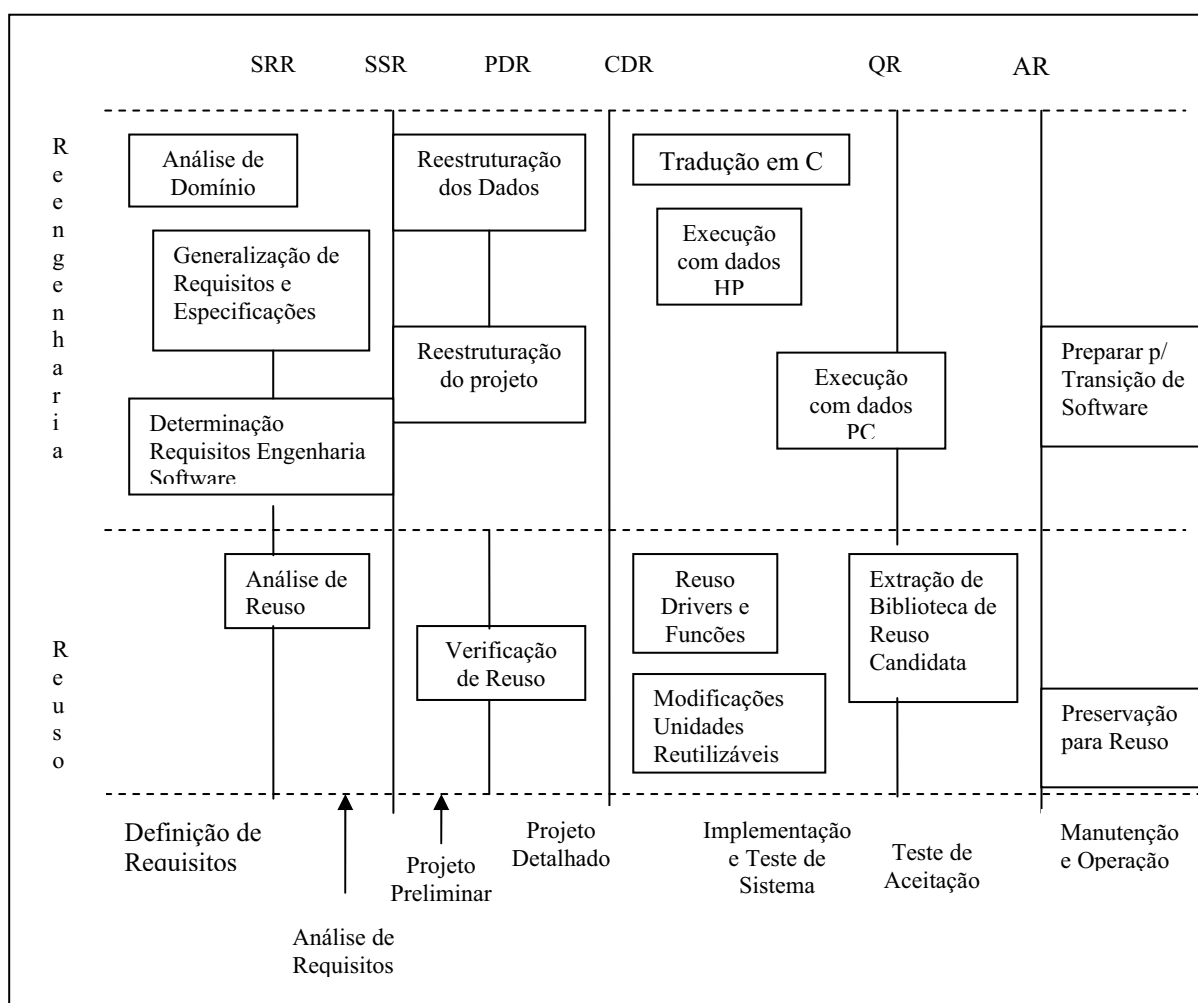


Figura 6.6 - Ciclos de vida com enfoque em reuso, reengenharia, ações e revisões.

As revisões técnicas do ciclo de vida com enfoque em reuso e em reengenharia, são orientadas pelas ações relativas a cada fase ou atividade, e visam atender a estruturação da documentação necessária ao projeto de desenvolvimento do sistema de software. Estas revisões são identificadas como indicado a seguir:

- *SRR (System Requirements Review)*: os documentos para Revisão de Requisitos de Sistema são os planos e as especificações do sistema e subsistemas de medidas do TA-2 e de requisitos operacionais básicos do TA-2. Também a análise de artefatos existentes ou funções disponíveis para o reuso, análise do domínio para a especificação de requisitos de sistema voltados para a reengenharia e o reuso.
- *SSR (Software Specification Review)*: na Revisão de Especificação de Software os documentos são os planos mais detalhados e as especificações de requisitos de software e de interface, voltados para a descrição operacional do sistema de medidas. As funções e artefatos disponíveis são analisados mais detalhadamente em relação aos requisitos operacionais do sistema e subsistema de software, voltados para a reengenharia e o reuso.
- *PDR (Preliminary Design Review)*: a documentação para a Revisão de Projeto Preliminar são os planos de teste de hardware e software, de garantia da qualidade e de gerenciamento de configuração, e o documento de projeto preliminar de software. Também são feitas a reestruturação de dados e do projeto na reengenharia, e a verificação de funções ou artefatos para o reuso.
- *CDR (Critical Design Review)*: para a Revisão Crítica de Projeto os documentos são as descrições do projeto detalhado de software e de interface, os planos de teste de software, de garantia da qualidade e de

gerenciamento de configuração completos, e os documentos das descrições dos testes. A reestruturação dos dados e do projeto estão mais detalhadas em documentos, à medida em que se entende melhor os subsistemas e o sistema, e em que os artefatos e as funções estão sendo mais exercitadas.

- QR (*Qualification Review*): para a Revisão de Qualificação os documentos são o código fonte e o código executável, o documento das descrições de versões, as descrições dos testes e resultados, a relação das configurações, o resumo das realizações de software, os manuais de usuário e do operador. O sistema de software é traduzido para a linguagem de programação escolhida para reengenharia, e os dados e resultados da plataforma HP e PC são comparados para a validação do software. São usados os *drivers* e funções das bibliotecas de funções do ambiente de programação visual, e procedidas algumas alterações para a reutilização de dados, artefatos, conhecimento, funções, entre outros.
- AR (*Acceptance Review*): para a Revisão de Aceitação os documentos são os planos de transição, instalação de software e o manual de operação do computador, que fornece informações necessárias para operar o computador e seus periféricos. Nesta etapa o sistema de medidas está integrado e documentado, e os softwares de aquisição e de redução de dados estão ajustados e preparados para a transição, as funções ou artefatos sendo preservados e disponibilizados para o reuso em projetos futuros.

Em um projeto de desenvolvimento de sistema de software as iterações ou revisões iniciais (SRR e SSR) são as mais importantes. Nelas são esperados os documentos e planos que determinarão a continuidade ou não de um projeto. No desenvolvimento do novo sistema de medidas, através de avaliações dos riscos da lista

de riscos, se determinou a continuidade deste novo projeto de modernização do TA-2, até a revisão ou iteração final.

No plano de avaliações se determinou o recurso financeiro e os prazos para o projeto de modernização do TA-2. Mesmo se tratando de um projeto de desenvolvimento para uma organização governamental de pesquisa, os riscos foram avaliados freqüentemente. As avaliações são feitas nas iterações do ciclo de vida, quando usando ações do processo unificado. Para gerenciar melhor as iterações, as avaliações das evoluções do sistema de software são processadas, em conformidade com as revisões estabelecidas no ciclo de vida.

Para isto, é descrita no plano de iteração outra modelagem gerencial adotada para o desenvolvimento do projeto do novo sistema de medidas, que corresponde ao fluxo de trabalho. Para este, dependendo do projeto em elaboração, foram propostos vários caminhos que podem ser percorridos, em cada etapa em que se está no ciclo de vida do desenvolvimento, conforme preconiza o processo unificado (RATIONAL CORPORATION, 2000).

Durante a fase inicial do ciclo de vida o ritmo de mudanças no desenvolvimento do sistema de medidas foi lento, ou seja, nas iterações iniciais onde se obtém a visão geral e as descrições do sistema novo ou modificado, a obtenção dos artefatos de planejamento e de gerência foi demorada. A identificação de áreas de risco para uso de tecnologias de apoio ao desenvolvimento (hardware e software), recursos financeiros e de pessoal, e para os processos ocorreu aos poucos. Delinearam-se então as atividades de redução destes riscos, aplicadas entre as iterações do desenvolvimento do projeto.

Assim, aos poucos estes riscos eram dirimidos e as descrições e visões foram se tornando abrangentes e detalhadas, sendo planejadas a gerência das atividades de redução destes riscos em diversos planos e para cada iteração, os quais são interligados ao plano de desenvolvimento. Esta modelagem pode ser vista na figura 6.7, aplicada ao projeto de modernização do sistema de software, através de um fluxo de trabalho, conforme preconizado no processo unificado (RATIONAL CORPORATION, 2000).

Neste fluxo de trabalho são descritas as declarações ou condições que especificam determinados fatos do processo, e determinam as atividades de avaliação da continuidade do projeto de modernização do sistema de software. Além disso, são exploradas as atividades gerenciais ou de engenharia que auxiliam a compreender melhor o projeto em desenvolvimento.

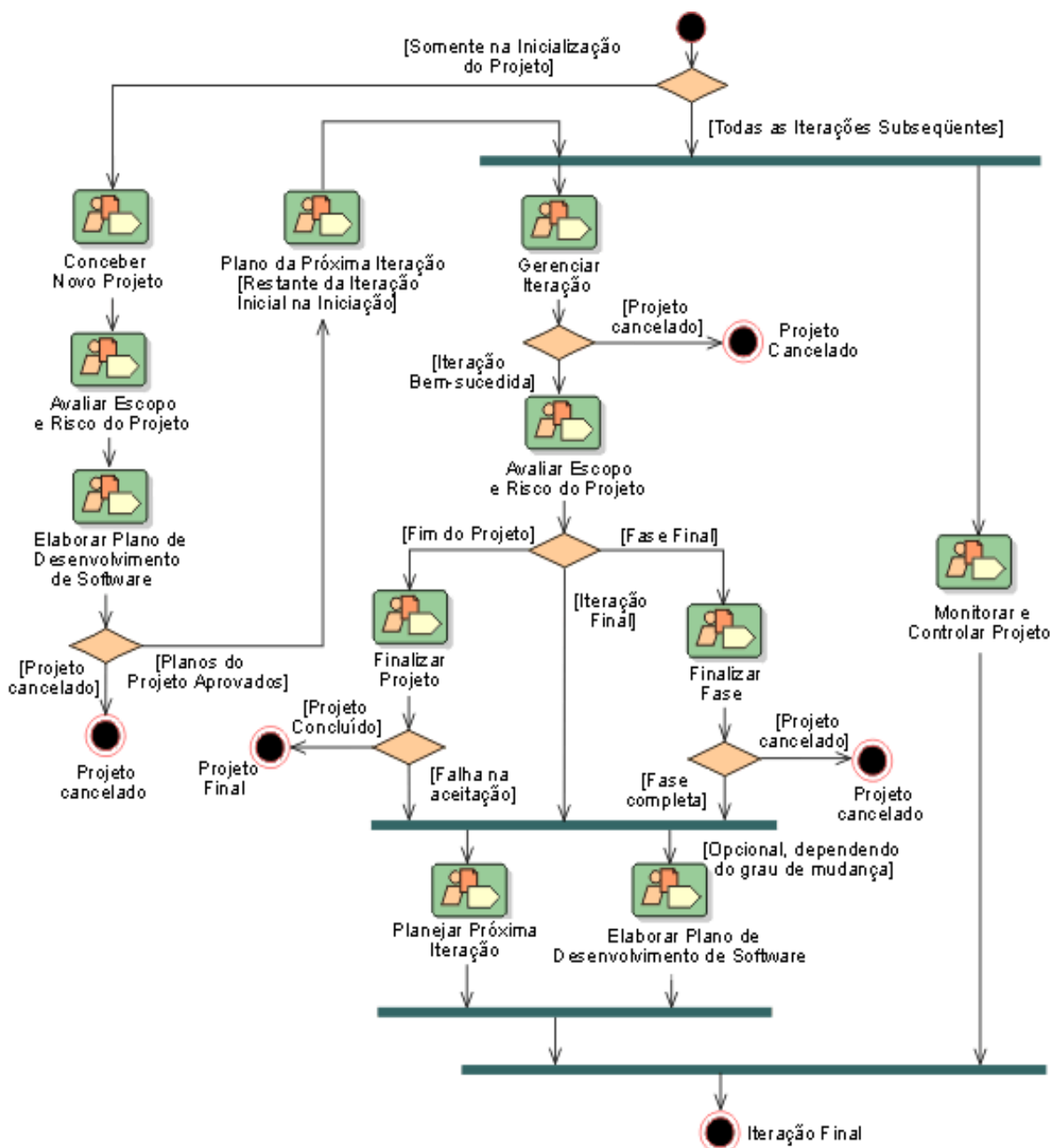


Figura 6.7 - Fluxo de trabalho do projeto de modernização do sistema de software (RATIONAL CORPORATION, 2000).

Muitas dessas atividades, referenciadas e modeladas graficamente já foram apresentadas neste trabalho, o que abrange o planejamento e a definição do novo sistema de medidas, o planejamento da evolução do projeto e avaliação dos riscos envolvidos, para cada iteração do processo. O que difere estas atividades do processo unificado das preconizadas na METAPLI são as ações gerenciais e técnicas, para obtenção dos subprodutos de software e documentação do projeto específico.

No fluxo de trabalho do projeto (RATIONAL CORPORATION, 2000) são mostradas as práticas efetuadas e representadas as possíveis situações, as quais têm lugar no projeto de desenvolvimento dos novos softwares. A característica fundamental da notação em forma gráfica é descrever os processos gerenciais e técnicos, que são identificados e usados durante um projeto de desenvolvimento de software. Essa modelagem pode tratar de representações de gerência ou de engenharia, aplicadas a um sistema de software.

Através da implantação da METAPLI é demonstrada a provisão de acesso a uma base de informações, representada e composta por fluxos de trabalho, fluxogramas e diagramas usados nas atividades de desenvolvimento do novo projeto de software. Essas informações são compartilhadas por todos os componentes do grupo de desenvolvimento do sistema de software do TA-2, garantindo que a visão do desenvolvimento de software em questão seja a mesma para todos os envolvidos.

As atividades de um processo para o ensaio aerodinâmico podem ser representadas em um fluxograma. Detalhes do fluxograma usado no TA-2 podem ser vistos no Apêndice B deste trabalho. As atividades se iniciam com o recebimento do modelo aerodinâmico, na qual o cliente normalmente envia um documento contendo as coordenadas deste modelo.

A partir deste recebimento e é definido em conjunto o programa de ensaio, o qual abrange as configurações que serão ensaiadas, bem como o material destas

configurações, que devem ser considerados pelo operador e outros especialistas durante o desenvolvimento e testes dos subprodutos de software de aquisição e de redução de dados. A previsão de carga e a preparação de instrumentação são especificadas pelos especialistas em ensaio em túnel. Também devem ser registradas as configurações de hardwares e dispositivos eletrônicos envolvidos, para a consistência e confiança nos dados e resultados obtidos.

Em todo o processo de calibração da balança são adquiridos os dados e armazenados através do software de aquisição de dados, para serem processados pelo software de redução de dados para obtenção das matrizes, usadas posteriormente no cálculo dos coeficientes aerodinâmicos. Por isso, os dados obtidos através do software de aquisição de dados devem ser avaliados e analisados, durante toda a campanha de ensaio. Neste trabalho, estes foram comparados com os dados obtidos com a plataforma HP, para que a transição para o novo sistema de medidas obtivesse sucesso.

Também foram elaborados neste trabalho alguns softwares na fase de construção, os chamados subprodutos ou artefatos técnicos, usados para obtenção das medidas de referência da calibração, validação e experimentação dos dados, e para validar outras instrumentações aplicadas nos ensaios em túnel para alguns modelos aerodinâmicos. Os softwares construídos referentes às demais instrumentações são opções de escolha, contidas em subprodutos desenvolvidos para os ensaios durante a modernização, que estão integrados ao software final do novo sistema de medidas do TA-2, com reuso de funções, artefatos e do conhecimento não mais fragmentado.

Para todas as atividades de desenvolvimento dos softwares e preparação do ambiente de ensaios em túnel, foram estabelecidos papéis com responsabilidades de executar as atividades e gerar os artefatos pertinentes. Sem estes, o projeto do novo sistema de medidas poderia consumir tempo disponível despropositado da equipe de projeto, e transmitir poucas informações relevantes sobre este empreendimento de modernização.

Através da METAPLI designou-se diversas atividades para o papel de engenheiro de software, responsável por muitos artefatos gerenciais gerados no decorrer do projeto de desenvolvimento. Cabe ao engenheiro de software liderar e coordenar as atividades gerenciais, para a obtenção dos artefatos técnicos ou subprodutos. Este trabalhador é encarregado de planejar e propor o conteúdo gerencial esperado para obtenção dos subprodutos e a ordem das iterações sucessivas.

Outra prática gerencial é o fluxo de trabalho que representa o processo de soluções de problemas, aplicado na mudança de partes do software ou documentação, cuja finalidade é levantar problemas, propor soluções para eles, controlar os desvios e as modificações a serem efetuadas, e avaliar o impacto nos subsistemas e nos softwares de aquisição e redução de dados, durante o desenvolvimento deste sistema de software.

O fluxo de trabalho de soluções de problemas que melhor descreve as atividades é mostrado na figura 6.8, o qual é aplicado no controle de configuração do sistema de software antes da fase de transição. Este fluxo de trabalho surgiu devido às dificuldades encontradas para lidar com requisitos variáveis, pois, nem todos os requisitos eram implementados corretamente e para cada problema de levantado, alguns ainda não teriam solução imediata e outros teriam diversas alternativas de solução.

Neste fluxo se estabelece, de forma gerencial, que as soluções propostas precisam ser analisadas com cuidado antes de serem efetivadas, principalmente em relação aos aspectos de engenharia dos subsistemas. Alterações em um subsistema podem afetar adversamente o resultado esperado e o desempenho de outros subsistemas, do sistema de medidas do TA-2. Assim, surgiu a necessidade de se adotar este fluxo de soluções de problemas para o sistema de software em desenvolvimento.

O processo de gerenciamento de soluções de problemas de software é acionado quando o subproduto de software ou a documentação associada estiver com problemas

que afetem a iteração. Quando este problema é relevante deve-se incluir as atividades de investigação e análise de modo mais criterioso, devido ao nível de importância com que os subprodutos e/ou sua documentação são afetados.

Os problemas levantados são registrados na lista de problemas, e aqueles que não apresentarem muita relevância podem ser considerados não prioritários, podendo ser solucionados quando liberados pelo revisor de engenharia. Assim se prossegue até ser executada a atividade de propor soluções, ou seja, até que a análise destes problemas menos relevantes possa ser feita, e a solução dada conjuntamente com os problemas relevantes.

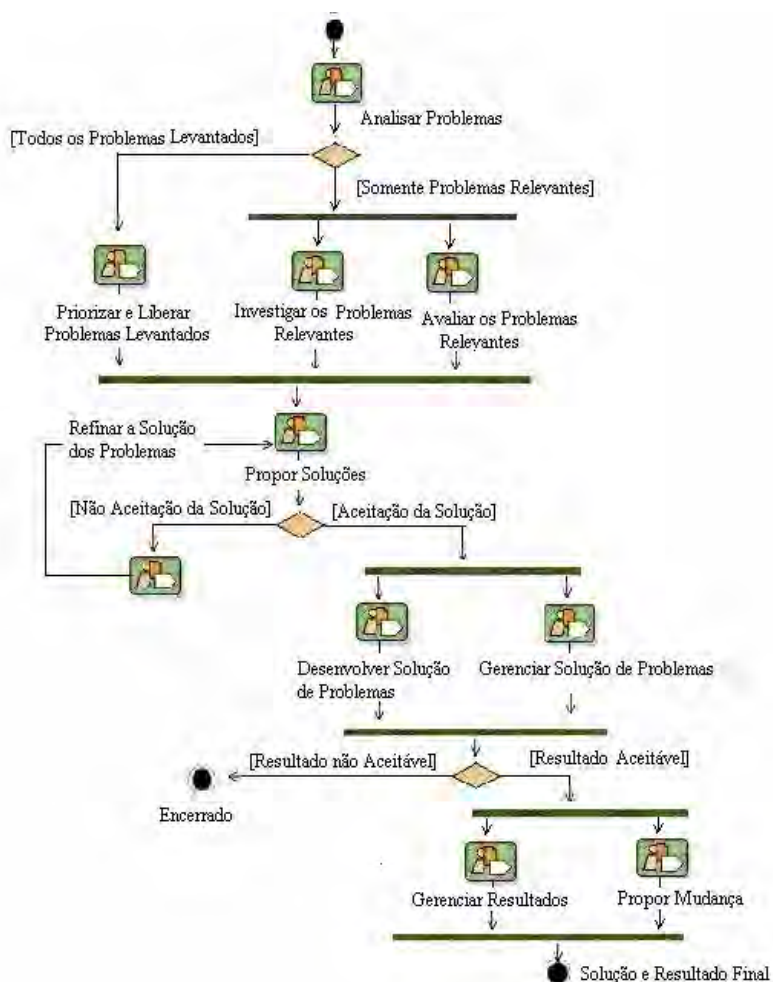


Figura 6.8 - Fluxo de trabalho de soluções de problemas de sistema de software

Para que o sistema de software não seja corrompido por mudanças não controladas, ou por evoluções nos subsistemas não compreendidas foi determinado o uso do fluxo de trabalho de soluções de problemas, e todos os envolvidos devem seguir as atividades prescritas neste processo. Essas atividades são desempenhadas durante todo o ciclo de vida, do desenvolvimento dos softwares de aquisição e redução dos dados do TA-2.

Coordenar as propostas de mudanças, armazenar as configurações em banco de dados, e gerar o documento de descrição de versão são atividades do revisor de engenharia. Este trabalhador é designado para efetuar as avaliações e revisões de documentos ou artefatos, métodos e processos, e dos subprodutos e produto final do sistema de software. As atividades e os artefatos designados para o revisor de engenharia são mostrados na figura 6.9.

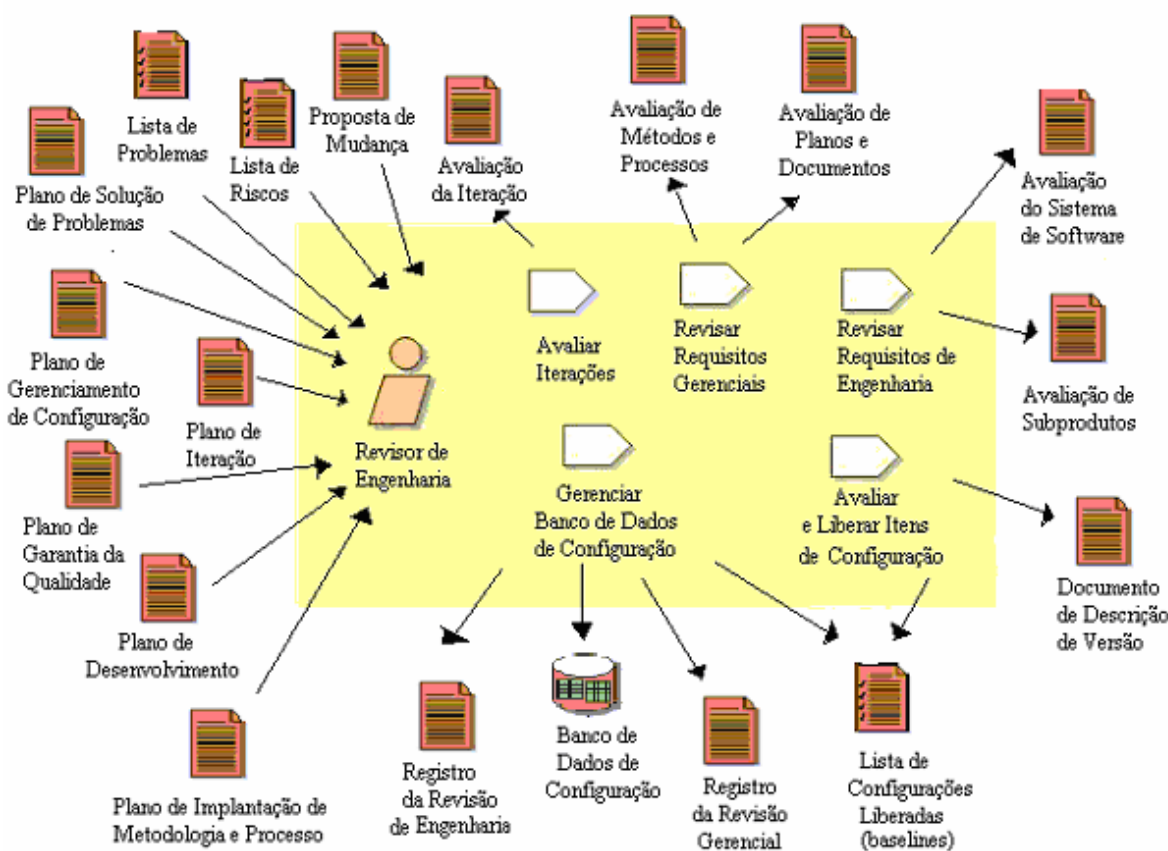


Figura 6.9 - Revisor de engenharia de sistema de software.

O revisor de engenharia é responsável por avaliar os artefatos ou documentos de desenvolvimento do projeto do novo sistema de software, nos principais pontos de revisão ou iteração do ciclo de vida do desenvolvimento. Os eventos de revisão são importantes e podem determinar o cancelamento do sistema de software, por um planejamento inadequado ou se o andamento do projeto se mostrar improdutivo.

As revisões ou avaliações dos processos técnicos e de ensaios dos subprodutos do sistema de software e dos documentos, também são de responsabilidade do revisor de engenharia, bem como a gerência do banco de dados de configuração feito em Microsoft Access, e a emissão do documento de descrição de versão, conforme preconiza a norma DOD-STD-2167A (1988).

Para o gerenciamento de configuração do sistema de software foi adotado um fluxo de trabalho, que trata de outra forma gerencial para representar as atividades, as quais estão ligadas aos papéis estabelecidos para o projeto do novo sistema de software. O fluxo de trabalho da figura 6.10 foi adotado para o gerenciamento de configuração e controle de mudanças sendo o mesmo da metodologia RUP (RATIONAL CORPORATION, 2002). Este fluxo abrange as atividades pertinentes à gerência da documentação do sistema de software do TA-2, dos subprodutos de software de cada iteração, e do produto final, através das ações implantadas via METAPLI.

É de responsabilidade do revisor de engenharia avaliar os métodos e processos aplicados durante o ciclo de desenvolvimento, para obtenção da qualidade do sistema de software. Este trabalho inclui revisar os elementos fundamentais dos subsistemas e os documentos suplementares, usados para manter o túnel de vento operacional. Estes incluem os artefatos desenvolvidos, planejados, e aplicados nos diversos subprodutos de software, principalmente aqueles entregues nos marcos importantes, considerados *baselines* ou *releases* previstos no plano de gerenciamento de configuração e demais planos.

Os subprodutos e artefatos do final de cada iteração ou final de cada fase do processo em cascata, depois de submetidos às avaliações e testes com obtenção de resultados satisfatórios, são “congelados” e entregues para uma revisão. A partir da aceitação dos códigos das unidades ou componentes de software, e dos documentos destes subprodutos, só podem ser alterados mediante processo de soluções de problemas e gerenciamento de configuração, via METAPLI.

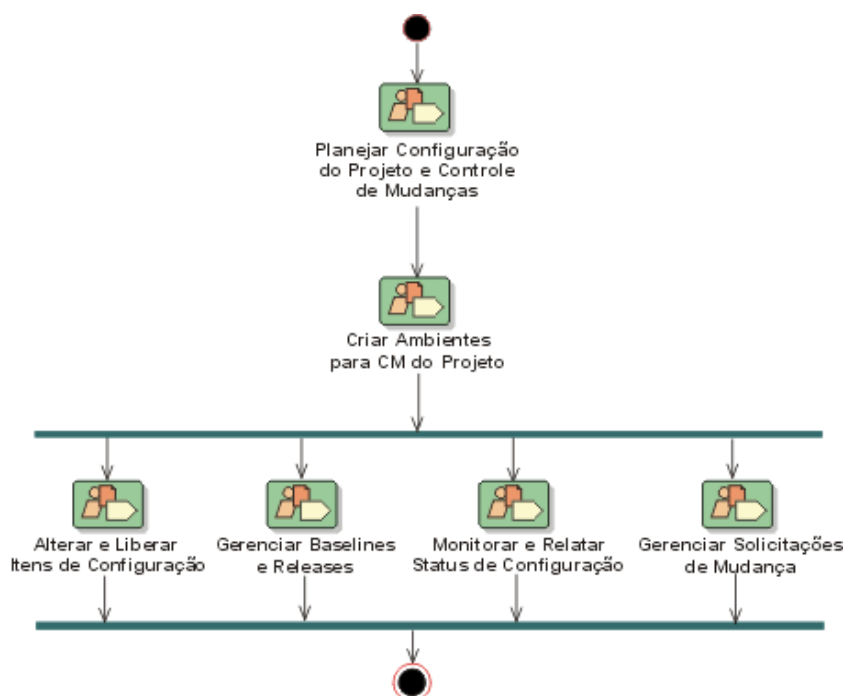


Figura 6.10 - Fluxo de trabalho do gerenciamento de configuração e controle de mudanças do RUP.

Um plano de gerenciamento de configuração é o documento gerado na primeira atividade do fluxo de trabalho da figura 6.10. Neste plano estão as descrições de padrões e procedimentos para gerenciar os subprodutos e documentos dos softwares de aquisição e de redução de dados. Também são definidas as responsabilidades pelos procedimentos de gerenciamento de configuração e pela submissão dos documentos e subprodutos ao revisor de engenharia. Foi elaborado um esquema formal, para identificar cada documento ou subproduto, conforme acordo entre especialistas em ensaios em túnel e parceiros de desenvolvimento.

No fluxo de trabalho há eventos para gerenciamento da configuração dos componentes ou partes dos subprodutos, entregues em cada final de iteração ou fase, dependendo do processo especificado. Com a definição do responsável pela obtenção dos subprodutos e artefatos funcionais, estes são entregues como partes dos itens de configuração, e cabe ao revisor de engenharia tratar dos documentos ou mídias, avaliando e monitorando as partes dos softwares submetidas ao controle, bem como gerenciar as solicitações de mudanças.

Nem todos os documentos técnicos ou gerenciais são controlados no projeto do novo sistema de medidas do TA-2. Entretanto, aqueles mencionados pela norma DOD-STD-2167A (1988) e outros considerados na metodologia deste trabalho, estão sob controle de configuração, entre os quais estão incluídos os planos de gerenciamento de configuração, garantia da qualidade, desenvolvimento de software, iteração, entre outros.

É mantido um banco de dados de configuração que é um repositório para uso dos envolvidos, os quais o utilizam para as armazenagens de documentos, código de desenvolvimento e testes de componentes, e software integrado e teste final. Os itens de desenvolvimento de software são controlados e mantidos armazenados em banco de dados, somente pelo revisor de engenharia, que também se responsabiliza pelo controle de cópias de documentos do projeto, e da distribuição desses subprodutos para os envolvidos.

Como os softwares são desenvolvidos em pares, quando estes trabalham separadamente no mesmo artefato devem fazer alterações seqüenciais, e os possíveis problemas devido às mudanças incorporadas devem ser resolvidos antes de sua entrega, podendo diminuir o ritmo da produção do software, mas garantindo a integridade do artefato e dos dados, conforme previsto no plano de garantia da qualidade de software.

As entregas dos itens de configuração são feitas com a concordância de ambas as partes, assim que forem definidas e testadas as funcionalidades básicas, ficando sob a revisão por parte do revisor de engenharia. Caso ocorram mudanças, na versão especificada após a entrega, deve-se seguir o processo de solução de problemas. O revisor de engenharia deve manter o relato dos defeitos e das demais partes afetadas, em um histórico dessas versões no documento de descrição de versão, conforme requisito da norma DOD-STD-2167A (1988).

Neste trabalho foram previstas quatro iterações na fase de construção para o software de aquisição de dados, das quais se obtém os subprodutos do desenvolvimento e suas documentações. À medida que os subprodutos de software foram sendo criados, surgiram muitas versões diferentes de software, que incorporaram propostas de mudanças, correções de defeitos e adaptações para os diferentes dispositivos eletrônicos, hardwares, e versões mais recentes dos sistemas operacionais e do ambiente de programação utilizado.

Como para o desenvolvimento do software de aquisição de dados se adotou o processo incremental e iterativo, utilizou-se uma abordagem que se baseia em diversas construções até completar todo o sistema de software. Assim, o planejamento da configuração do projeto e o controle de mudanças abrangem mais intensamente cada uma das versões, das construções previstas no desenvolvimento durante todo o ciclo de vida, com o armazenamento e a gerência das configurações em banco de dados.

Também abrange a gerência e o controle das partes de subprodutos do software de aquisição de dados, tais como os *drivers* do ambiente de programação virtual, e as funções que compõem as extensões “vi”. Os documentos ou manuais correspondentes são registrados ou listados, e também aqueles documentos referentes aos *drivers* e funções em que as descrições são reproduzidas, desde que previstos e determinados em prazos de entrega no planejamento.

A construção de partes dos subprodutos de software acompanhada de suas documentações também contém a descrição dos problemas que surgiram na criação e interações dessas partes e nos testes executados das versões. Tais testes devem resultar de versões alteradas ou novas versões, que apresentem construções bem sucedidas, ou seja, a descoberta e resoluções de defeitos de software devem ser descobertas durante os testes dessas unidades, e documentados os problemas e os reparos ou soluções conforme fluxo de trabalho definido na figura 6.8. O gerenciamento de versões dos subprodutos é controlado através do registro em banco de dados de um histórico das alterações, acompanhado das versões das documentações pertinentes.

Praticamente todas as representações gráficas contidas neste trabalho, com respeito à modelagem gerencial do desenvolvimento do sistema de software, deram suporte aos envolvidos no desenvolvimento dos novos softwares, e são adotados até hoje pela organização.

Um dos aspectos considerados importantes na modelagem gerencial ou técnica é a organização das pessoas com as várias competências, identificadas e definidas através dos papéis, com as atividades e ações para produzir a documentação, os subprodutos e o produto final desejados. Assim, os esforços de gestão foram feitos priorizando a provisão da melhoria na qualidade dos softwares e dos dados gerados, visando quando possível o aumento da produtividade do TA-2.

Para este trabalho, alguns processos foram estabelecidos e modelados, os quais têm a finalidade de garantir qualidade usando ambiente de programação apropriado, testes formais para comprovação de requisitos, verificações de projeto e aspectos de implementação com base nos softwares existentes.

A implantação da METAPLI provê a geração de planos, procedimentos, métodos e processos para as atividades de desenvolvimento de software, gerenciamento de configuração de software, revisões de documentos, validação dos subprodutos de

software, abrangendo as incorporações graduais das funcionalidades dos subsistemas do sistema de medidas.

No decorrer do projeto do novo sistema de software, o acompanhamento e o controle do que era realizado pela engenharia durante o desenvolvimento, foi comparado com o planejado. Quando ocorressem desvios ou mudanças fossem necessárias para evolução do sistema de software, as ações cabíveis eram tomadas, sempre mantendo atualizado o histórico do projeto de software.

As atividades ou processos técnicos aplicáveis ao desenvolvimento dos softwares de aquisição e redução de dados e aos demais subprodutos importantes, que possibilitassem manter operacional o sistema de medidas do túnel de vento, foram documentados através de práticas da METAPLI. A metodologia provê modelagens através de ações do processo unificado e tradicional, a partir das quais se obteve a documentação técnica do projeto do novo sistema de medidas.

6.3 DOCUMENTAÇÃO TÉCNICA VIA METAPLI

Para o desenvolvimento dos softwares de aquisição e redução de dados do TA-2, as atividades de engenharia são similares a um processo industrial. A cada etapa do ciclo de vida, os envolvidos no desenvolvimento do sistema de software geraram os subprodutos, que puderam ser operados e entendidos por todos os elementos da equipe.

A METAPLI é implantada abrangendo as práticas dos processos de desenvolvimento selecionados, provendo também a modelagem das atividades do sistema de software: requisitos, análise e projeto, implementação, teste e implantação. Suas práticas são aplicadas conforme o plano de implantação de metodologia e processo, para a obtenção dos subprodutos de software e documentação pertinente a

cada iteração ou revisão previstas no ciclo de vida, considerados os marcos para as atividades de entregas e aceitações frequentes.

Durante o processo de desenvolvimento do software de aquisição de dados, foi dada ênfase no reuso de funções, provenientes da biblioteca de funções do ambiente de programação visual, que facilitou uma parte do planejamento e da aplicação de engenharia. Para o desenvolvimento do software de redução de dados foi dada ênfase na reengenharia dos programas em linguagem BASIC existentes, que auxiliou na produção da documentação do sistema de medidas do TA-2, facilitando o entendimento deste sistema e dos subsistemas interligados.

Para se entender melhor a modelagem do sistema de software, as práticas empregadas no desenvolvimento foram descritas, em sua maioria, para cada software em separado, ou seja, foram geradas modelagens das atividades do desenvolvimento de software de aquisição de dados em separado daquelas do software de redução de dados. Há poucos casos em que estas modelagens das atividades técnicas do desenvolvimento podem ser as mesmas, e quando isto ocorre, são descritas de forma que haja interação entre as atividades as quais representam.

6.4 RESULTADO DA MODELAGEM DO SISTEMA DE SOFTWARE

A modelagem do sistema de software baseia-se numa diversidade de recursos práticos e numa arquitetura aberta, desenvolvida em parceria com conceituados desenvolvedores do mercado industrial de software. Esta modelagem visa agregar funcionalidade e garantir ao sistema de medidas, um nível elevado de qualidade e de confiança no funcionamento.

Para este trabalho os envolvidos priorizaram a obtenção de dados e de resultados dos ensaios em similaridade aos obtidos com o sistema antigo. Então, foram elaborados testes em que esses dados e resultados pudessem ser comparados de

imediatamente, sem afetar a dinâmica de trabalho do ensaio em túnel. Para isto, houve a necessidade de gerar outros subprodutos ou usar aplicativos específicos nas iterações da fase de construção, que contribuíssem para a validação dos dados e resultados de ensaios aerodinâmicos.

O uso das práticas da METAPLI para o desenvolvimento do sistema de software tem a finalidade auxiliar na modelagem desse sistema. Isso ocorre gradualmente, através das descrições e representações dessa modelagem, que vai tornando o sistema menos abstrato na medida em que vai se detalhando os requisitos. Os modelos de requisitos e de arquitetura são aplicáveis a praticamente todas as fases de desenvolvimento do sistema de software do TA-2.

Em todas as fases do ciclo de vida de desenvolvimento foram introduzidas práticas gerenciais e técnicas usadas na indústria de software, visando facilitar a validação do código, a obtenção de dados e resultados confiáveis, a permanência operacional do túnel de vento e a implantação das configurações de hardware e de software pertinentes. A modelagem do sistema de software tem o objetivo de padronizar os métodos de trabalho, provendo o reuso de conhecimento para obter a consistência do projeto dos softwares de aquisição e de redução de dados.

As práticas da METAPLI, implantadas e descritas em parte no capítulo 3 deste trabalho, para as fases de iniciação e elaboração do sistema de software, abrangem algumas atividades descritas em detalhes na segunda iteração da fase de elaboração, para esclarecer e definir como proceder nas quatro construções previstas. Desde o início desta segunda iteração da fase de elaboração do software de aquisição de dados, implantaram-se as seguintes práticas:

- Desenvolvimento Iterativo e Incremental
- Gerência de Requisitos e Riscos
- Arquitetura de componentes
- Verificação contínua da qualidade

- Modelamento do sistema de software
- Gerenciamento de mudanças

Como mencionado anteriormente, o enfoque iterativo e incremental do desenvolvimento de software de aquisição do TA-2 segue as atividades de forma parecida com o chamado “processo” do enfoque de desenvolvimento tradicional, ou seja, abrange as atividades de requisitos, análise e projeto, implementação e teste, também conhecida como estrutura estática no processo unificado, sendo estas atividades listadas nas fases do processo tradicional.

A gerência de requisitos e riscos neste trabalho para o desenvolvimento do sistema de software segue as atividades ou fases citadas de ambos os processos. No entanto, com uso do processo unificado no desenvolvimento do software de aquisição de dados, é possível a prática de modificação de requisitos, artefatos e de decisões gerenciais e técnicas em função dos riscos, até mesmo durante a obtenção de resultados dos subprodutos da fase de construção. Isto significa que aplicando as atividades do processo unificado adequadamente, há possibilidade de fazer modificações e melhorias aceitáveis no software de aquisição de dados ou documentos pertinentes a qualquer tempo, sem a necessidade de fazê-las ou analisá-las somente na fase ou atividade de requisitos.

As modelagens para os softwares de aquisição e redução de dados são diferentes, devido à definição de seus processos seguirem ações para representá-los de formas diferentes. No entanto, as documentações referenciadas na norma DOD-STD-2167A (1988) são as mesmas. Isto significa que, independente das ações do processo adotado, devem ser gerados os documentos preconizados na norma, referentes às atividades de requisitos, análise e projeto, implementação e teste do sistema de software.

Para implantar a METAPLI com as ações do processo unificado no desenvolvimento do software de aquisição de dados, cada disciplina é abordada com

as modelagens que melhor se adaptem à versão usada do ambiente de programação gráfica visual comercial. Para a modelagem do software de redução de dados adotou-se a abordagem estruturada, com algumas variações para adequação dos métodos e processos de engenharia do sistema de medidas, voltados para os ensaios aerodinâmicos.

Cabe lembrar que, as atividades referentes ao domínio do problema foram apresentadas no capítulo 3, no qual é descrita a arquitetura do projeto do sistema de medidas do TA-2. Esse sistema é composto pelos subsistemas de aquisição e de redução de dados com novos softwares, e dos demais subsistemas interligados, com implementação de medições, controles e processamentos, adequados à modernização do sistema de medidas.

Para esta modernização foi criada uma modelagem especial para o ambiente de desenvolvimento, considerada neste trabalho como uma representação técnica do sistema de medidas, devido às suas especificidades. A modelagem técnica permite que todo o suporte ao ambiente de desenvolvimento seja feito por profissionais alocados no túnel de vento. Desta forma houve facilidades para o processo de aquisição, instalação e configuração das ferramentas, aplicativos, hardwares de apoio, e do ambiente de programação visual comercial, necessários para esta modernização em específico.

6.4.1 Modelagem do ambiente de desenvolvimento

A modelagem do ambiente de desenvolvimento do projeto de modernização do TA-2 tem a finalidade técnica de fornecer e manter disponíveis os recursos de hardware e software de apoio, os ambientes de programação e as ferramentas ou aplicativos de software estabelecidos para suportar o desenvolvimento do sistema de medidas.

Esta foi implantada via METAPLI com ações um pouco diferenciadas do processo unificado, devido à complexidade do sistema de medidas. Esta modelagem do ambiente de desenvolvimento engloba os aplicativos operacionais e de apoio ao desenvolvimento e a documentação do sistema de software.

Também, os dispositivos eletrônicos ou hardwares específicos da *National Instruments* não são tratados pela modelagem do ambiente de desenvolvimento. Estes componentes de hardware fazem parte da arquitetura do sistema de medidas do TA-2, e foram tratados nas modelagens de requisitos, análise e projeto, e implementação e teste, nos quais suas interligações foram detalhadas.

Assim, a modelagem de ambiente trata especificamente dos computadores, das ferramentas e do ambiente de programação usadas para desenvolver os softwares de aquisição e redução de dados, e demais ferramentas de suporte para gerar softwares ou para serem aplicadas com a finalidade de avaliar e analisar os dados, transferir dados entre as plataformas, ordenar dados experimentais, entre outras.

Os recursos relativos aos computadores de apoio, ambientes e ferramentas foram selecionados para suportar o desenvolvimento dos softwares principais do sistema de medidas. Os demais recursos de hardware, aplicativos ou ferramentas foram requisitados na medida das necessidades para o auxílio no desenvolvimento ou para aplicação específica nos softwares de ensaios em túnel, na validação de dados experimentais e validação de redução de dados. Para este papel foi designado o especialista em ferramentas e ambiente, conforme mostra a modelagem de ambiente de desenvolvimento da figura 6.11.

Em sua modelagem de ambiente de desenvolvimento, destaca-se o papel deste especialista como responsável por instalar, configurar e verificar a funcionalidade dos computadores, ferramentas ou aplicativos de apoio utilizados no projeto de software, tais como, microcomputador *Pentium*, *Microsoft Windows*, *Microsoft Office*, *C++ Builder 4*, *LabView*, entre outros.

Também se designa para este papel a responsabilidade por manter operacional o ambiente de desenvolvimento da plataforma onde os softwares serão instalados, ou seja, a manutenção dos hardwares e softwares de apoio, *backup*, redes, entre outros, do túnel de vento como um todo e, principalmente, daqueles disponibilizados para desenvolvimento dos softwares e processamento dos dados de ensaio aerodinâmico.

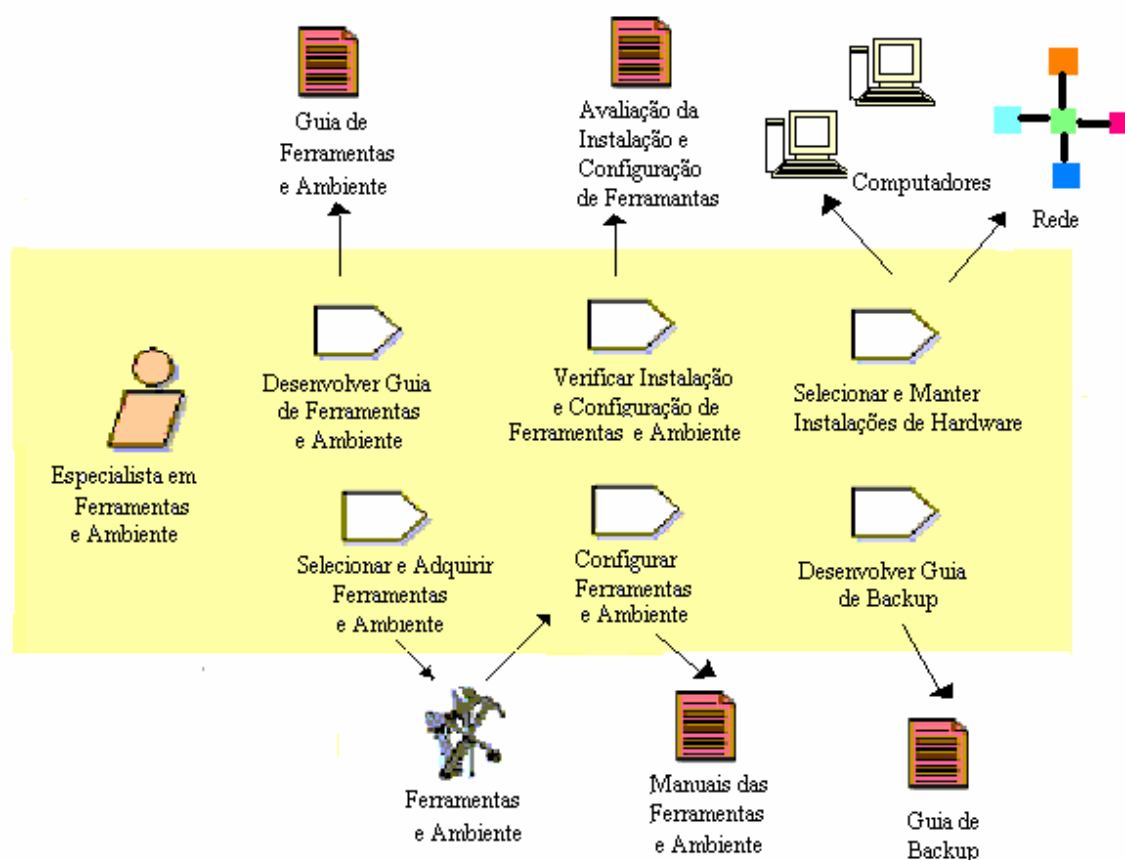


Figura 6.11 - Modelagem de ambiente de desenvolvimento

Com o papel e as responsabilidades definidas, cabe ao especialista dar suporte e cuidar da utilização adequada dos hardwares, ambientes e ferramentas, para possibilitar a elaboração dos artefatos, subprodutos ou documentos do desenvolvimento dos softwares de aquisição e de redução de dados, e também dos demais softwares de suporte ao desenvolvimento ou análise e interface de dados.

Conforme já mencionado em capítulos anteriores, o ambiente de desenvolvimento do sistema de software do TA-2 representa um conjunto de atividades relacionadas à coordenação e uso de hardwares e softwares de apoio, que podem melhorar a produtividade dos documentos e do desenvolvimento do novo sistema de medidas. Neste ambiente não se inclui ferramenta CASE, pois o aplicativo *LabView* em sua versão 3.1, disponível para uso no desenvolvimento do novo sistema de software do TA-2, em 1998, não suportava diretamente uma programação orientada a objetos.

Também conforme mencionado anteriormente, em uma análise criteriosa detectou-se que a ferramenta para UML (*Unified Modelling Language*), que é parte de uma ferramenta CASE, sendo a principal ferramenta de programação orientada a objeto quando usadas ações do processo unificado, não possuía geração automática de código em *LabView*. Isto implicou na inviabilidade do uso de ferramentas CASE e de modelamento orientado a objeto, com declaração explícita de unidades e atores, casos de uso e estados, pois estes não poderiam ser integrados e tomados por completo para a geração do sistema de software. Este fato foi avaliado como um risco tecnológico que tornou inviável o uso de ferramenta CASE. Além disso, restrições financeiras também contribuíram para inviabilizar a aquisição da ferramenta RUP.

Assim, neste trabalho os modelos são abordados sob a perspectiva de processo tradicional, em que as funções pretendidas para o sistema de software estão ligadas ao ambiente de ensaio em túnel, ou seja, são orientadas a funções com uma mesclagem de conceitos do processo unificado e do tradicional, adotados em comum acordo com os envolvidos no desenvolvimento.

As modelagens dos softwares a que se faz referência neste trabalho foram feitas em atividades conjuntas para cada construção prevista. São agrupadas as modelagens

das atividades de requisitos com as de análise e projeto, que foram conduzidas a partir de requisitos e considerando os dispositivos eletrônicos disponíveis no TA-2.

Também foram agrupadas as atividades de projeto detalhado, implementação (codificação) e teste, que foram conduzidas com as arquiteturas de hardwares e softwares definidas, para as quais alguns casos de teste são apresentados, e demonstrados os dados e resultados alcançados com os novos softwares, que são comparados aos obtidos com os softwares da plataforma HP.

6.4.2 Modelagem de requisitos, análise e projeto

Uma modelagem de requisitos é usada neste trabalho quando um conjunto de requisitos é gerado em determinada fase do ciclo de vida do sistema de software. A modelagem de arquitetura ou análise e projeto é usada quando estes requisitos, de hardware ou de software, são distribuídos entre unidades físicas ou lógicas do sistema de medidas do TA-2.

A especificação de requisitos independe da tecnologia, ou seja, um conjunto de requisitos pode ter várias soluções possíveis, que são analisadas pelos responsáveis designados para os diversos papéis, que envidam esforços para a escolha de uma solução para o problema, a qual depende do que for decidido para a transformação.

Um modelo de arquitetura depende da tecnologia, devido às decisões para transformação influenciarem nas iterações entre os modelos de requisitos e de arquitetura de software ou de hardware. Portanto, a transformação de um modelo de requisitos num modelo de arquitetura é considerada um processo iterativo, que é usado para resolver todas as interfaces e distribuir os requisitos entre as alocações de arquitetura, esta última definida neste trabalho na modelagem de análise e projeto.

Praticamente todas as atividades do ciclo de vida do sistema de software se iniciam simultaneamente e variam de esforço e escopo na progressão do desenvolvimento. As bases iniciais adotadas devido a esta simultaneidade, segundo representado via METAPLI, são os modelos e artefatos da modelagem geral do sistema de medidas. Em conformidade com esta modelagem na atividade de requisitos é preparado o documento de especificação de requisitos de software, conforme as solicitações dos principais envolvidos e a visão do sistema de medidas do TA-2 e de seus subsistemas.

O diagrama do sistema de medidas e subsistemas apresentado no capítulo 3 representa o diagrama de contexto do sistema de software, com os elementos externos com os quais os softwares de aquisição e redução de dados interagem. Para estes softwares foram descritos os requisitos de funcionais e de domínio da engenharia, que abrangem as atividades desde o início de seu desenvolvimento para fornecerem integridade dos subsistemas físicos do TA-2.

Os requisitos funcionais e de domínio são cobertos com as modelagens do sistema de software e constam do documento de especificação de requisitos de software. Os requisitos não funcionais são tratados com a modelagem gerencial. Os requisitos de interface que são definidos no documento de requisitos de interface são alocados de acordo com os subprodutos e entidades ou dispositivos eletrônicos (hardwares), com os quais o software de aquisição de dados troca dados e controles, com possibilidade de reuso na construção de outro subproduto.

Os requisitos constantes nestes documentos são derivados da análise do sistema de medidas do TA-2, o qual é subdividido em subsistemas de acordo com as funcionalidades. A funcionalidade dos softwares de aquisição e de redução de dados inclui os aspectos necessários para atender aos requisitos dos especialistas em ensaios no túnel e seus clientes. Através das modelagens buscou-se garantir a qualidade do sistema de medidas, e obter subprodutos de software e documentos aceitáveis para

manter em operação o TA-2. Para cada atividade são definidos os artefatos e os papéis detalhados na figura 6.12 como de responsabilidade do analista de sistema.

Para o papel de analista de sistema foi designada uma responsabilidade importante: trata-se da captura de um vocabulário comum que contempla a terminologia básica, que consta do glossário deste trabalho no Apêndice C. O analista de sistema tem a responsabilidade de atualizar este glossário, e gerar um documento no qual estão definidas as terminologias comuns usadas no projeto de modernização do TA-2, de forma consistente para uso de todos envolvidos.

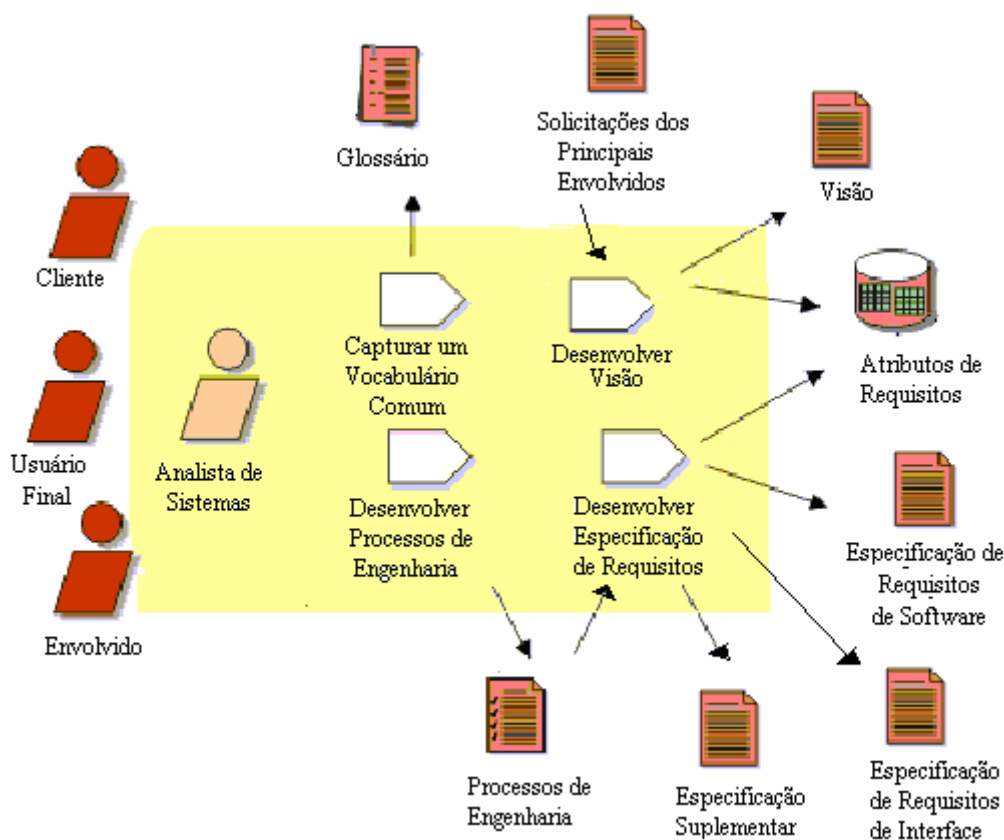


Figura 6.12 - Papel do analista de sistema na modelagem de requisitos

É papel do analista de sistema gerar os documentos de visão, especificação de requisitos, e especificação suplementar, nos quais são descritos o sistema de medidas do TA-2, seus subsistemas, e os requisitos funcionais e de domínio. Para o atendimento aos requisitos do subsistema de aquisição e de redução de dados são associados um ou mais subprodutos funcionais, que incluem hardwares e softwares

essenciais para o ensaio aerodinâmico e permanência operacional do TA-2, durante o projeto do novo sistema de medidas. As especificações de requisitos e suplementar são complementadas com detalhes técnicos e operacionais de ensaio para cada subproduto construído através do artefato de processo de engenharia.

O analista de sistema é responsável por especificar os processos de engenharia para os ensaios a serem executados pelos subprodutos de software, apropriados para a implementação de cada subproduto. Trata-se do desenvolvimento da atividade referente aos processos de ensaios para a zeragem da balança, ZQ, ZAB, TARA, entre outros, para cada subproduto a ser desenvolvido.

É também papel do analista de sistema o desenvolvimento das atividades referentes aos processos operacionais dos ensaios, para possibilitar o projeto e implementação destes nos subprodutos referentes ao novo sistema de medidas do TA-2. Trata-se dos processos ou métodos de calibração da balança e daqueles pertinentes à calibração de outras instrumentações, métodos e cálculos matemáticos aplicáveis para obtenção das matrizes, processos para estabelecer a amostragem de dados necessária para cada ensaio aerodinâmico, processos para aprovar ou rejeitar conjuntos de dados de aquisição de cada amostra, entre outras atividades pertinentes.

Com a definição dos processos de ensaios e operacionais aplicáveis, os subprodutos de software podem ser construídos de forma relativamente independentes, visando um processo gradativo de implementação e integração, contendo componentes ou unidades de software, *drivers*, funções, subfunções entre outros elementos, codificados ou adaptados de ferramentas ou códigos existentes.

Para fins deste trabalho são especificados os requisitos e definidos os subprodutos funcionais, que serão desenvolvidos nas iterações da fase de construção. Estes seguirão os processos de análise, projeto, implementação e teste, com as aplicações de práticas da METAPLI. Os subprodutos de software foram agrupados considerando as funcionalidades que deveriam executar, para atender a demanda de

serviços de ensaios em túnel, solicitados pelos clientes naquela época. São definidos os papéis designados para obtenção dos artefatos e execução das atividades, ligadas às funcionalidades necessárias para a construção dos subprodutos.

Estes subprodutos são abordados neste trabalho pelos tipos de serviços, conforme apresentado a seguir:

SWA – Software de Apoio ao Desenvolvimento: são softwares COTS que são instalados em computadores pessoais para auxílio ao desenvolvimento, visando seu uso para documentação ou modelagem de dados. Esses são subdivididos em software operacional - SOP, como por exemplo, Windows e Unix, e software aplicativo – SAP, tais como Microsoft Word, Microsoft Office entre outros, sob responsabilidade do especialista em ferramenta e ambiente de desenvolvimento.

SWE – Software de Ensaios em Túnel: são os subprodutos de software que visam dotar o sistema de medidas e/ou seus subsistemas de funcionalidades para operação em ensaios aerodinâmicos específicos. Estes subprodutos têm o objetivo de auxiliar na verificação e validação de parte das funcionalidades do sistema de medidas, e manter operacional o TA-2. Parte destes serão integrados ao software final de aquisição de dados considerando o enfoque em reuso, e suas saídas irão auxiliar na reestruturação de dados do software de redução. Sua documentação e construção estão sob responsabilidade de diversos papéis, tais como analista de sistema, especialista em interface, arquiteto de instrumentação e arquiteto de software. Esse tipo de software foi subdividido em:

1. *Software de Aquisição de Dados do Laboratório de Força – SLF:*
desenvolvido para o processo de calibração de células de carga no Laboratório de Força e o armazenamento de dados, para obtenção dos coeficientes de calibração, que servirão de base para comparação quando da criação do software de aquisição de dados final. Esse software será

associado à parte do software de aquisição de dados do TA-2, com reuso daquelas funções e dispositivos eletrônicos e computacionais pertinentes.

2. Software de Aquisição de Dados de Scanivalve – SSV: desenvolvido para aquisição de dados da balança e de “scanivalves”, com comandos do operador para atuação dos ângulos do subsistema mastro, com dispositivos eletrônicos adequados, com reuso de partes pertinentes do subproduto e dados do SLF.
3. Software de Aquisição de Dados e Calibração da Balança – SBA: desenvolvido para aquisição de dados da calibração da balança e dos ensaios dos modelos, com comandos do operador para atuação dos ângulos do subsistema mastro, com dispositivos eletrônicos adequados, com reuso de partes pertinentes do subproduto e dados do protótipo SSV.
4. Software de Aquisição de Dados e Controle do Túnel de Vento – SCB: - software de aquisição de dados da balança e de controle do subsistema motor e hélice, com comandos e monitorações dos sinais dos demais subsistemas, acrescentando mais funcionalidades e englobando e aprimorando aquelas demonstradas pelo subproduto SBA.

SWV - Software de Validação de Dados Experimentais: são softwares que visam modelar características dos dados ou de parte dos dados, com o propósito de validar o comportamento dos dados, avaliar soluções alternativas para sua aquisição, identificar problemas nestes dados e reduzir os riscos do uso destes dados pelo software de redução. Os papéis ligados a este tipo são o especialista em interface, arquiteto de software, arquiteto de instrumentação e analista de sistema. Esses softwares foram subdivididos em:

1. Software com Aplicativo ARENA para Cálculo de Distribuição de Dados – SAV: uso do software ARENA, aplicado para cálculo de distribuição dos

dados de forças e de momentos. Essa aplicação de software é usada para a análise dos dados, que determina se deve ser executado o processamento desses dados pelo software de redução dos dados, ou se esses dados devem ser amostrados novamente.

2. Software com Aplicativo EXCEL para Ordenação de Dados da Scanivalve – SVE: software elaborado através dos recursos disponíveis de edição e gravação de macro da ferramenta Microsoft Excel, para ordenação de dados de pressão adquiridos através do SSV. Esse software foi usado para auxílio à redução de dados de um modelo durante o desenvolvimento dos softwares de aquisição e de redução de dados, para viabilizar a obtenção dos resultados do ensaio de um modelo aerodinâmico, solicitado por cliente do TA-2.

SWI – Software de Interface: são softwares COTS adquiridos da *National Instruments*. Tratam-se de softwares de *drivers* com bibliotecas de interface GPIB e DAQ usados durante o desenvolvimento dos softwares de ensaio. Esses softwares são considerados como de reuso e foi designado o papel de especialista em interface para coordenar essas atividades. Esses softwares foram subdivididos em:

1. Software Aplicativo de Interface GPIB – SIG: consiste na aquisição comercial de *driver* do ambiente de programação visual, o qual faz a comunicação com outros dispositivos GPIB através de comandos advindos de um microcomputador. Possui um protocolo padrão IEEE 488-2, que corresponde a rotinas de alto nível que combinam um número de sequência de controle, para operações dos instrumentos. Esta interface de comunicação é constituída de um conjunto de comandos de instrumentos específicos, advindo do *driver* específico ou software aplicativo COTS, comandos estes que controlam o dispositivo eletrônico.

2. Software Aplicativo de Interface DAQ – SIW: consiste em outro software comercial ou software de *driver* da ferramenta *LabView*, o qual faz a comunicação do microcomputador com dispositivos eletrônicos ou placas da *National Instruments*, conhecidos como hardwares ou dispositivos DAQ, com a função de interface de comandos e controles para aquisição de dados.

SWR – Software de Validação da Redução de Dados: são softwares desenvolvidos para o subsistema de redução de dados *stand alone*. Abrange o desenvolvimento de um software que transfere os dados da plataforma HP para o PC, visando a validação dos dados e resultados da redução de dados. Inclui também o desenvolvimento do software de redução de dados deste trabalho, para obtenção dos resultados de calibração e dos ensaios com o modelo aerodinâmico. Para estes tipos foram designados os mesmos papéis definidos para os SWE, tais como analista de sistema, especialista em interface, arquiteto de instrumentação e arquiteto de software. Estes são subdivididos em:

1. Software de Redução de Dados do Túnel de Vento – SRD: software que calcula os coeficientes aerodinâmicos, com base em reajustes dos dados do processo de calibração que geram diversas matrizes, as quais são processadas para obtenção dos resultados do ensaio aerodinâmico.
2. Software de Transferência de Dados da HP para o PC – STD: software desenvolvido em linguagem C, para processar a transferência dos dados da plataforma HP para PC, em cumprimento ao requisito especificado de manter operacional o TA-2 durante a modernização. Também foi usado para validação do SRD, através da comparação de dados e resultados com execução dos respectivos softwares nas plataformas pertinentes.

Todo o sistema de software precisa fornecer algum meio de apresentar as informações ao usuário, podendo ser de forma textual ou gráfica. Para encontrar a melhor apresentação das informações, o conhecimento e a experiência dos

especialistas em ensaios em túnel foram imprescindíveis. Assim foi possível determinar o papel do especialista de interface de modo que se responsabilizasse, além de outras atividades, por encontrar a maneira adequada para apresentar as informações do sistema de medidas.

O especialista em interface provê algum meio de apresentação das informações aos usuários ou operadores do TA-2, por intermédio de telas, gráficos e tabelas a serem construídos para os softwares de aquisição e de redução de dados. Este é responsável pela abordagem visual de todos os dados pertinentes aos ensaios aerodinâmicos, e por prover a melhor das possibilidades de apresentação dessas informações. O papel do especialista de interface pode ser visto na Figura 6.13.

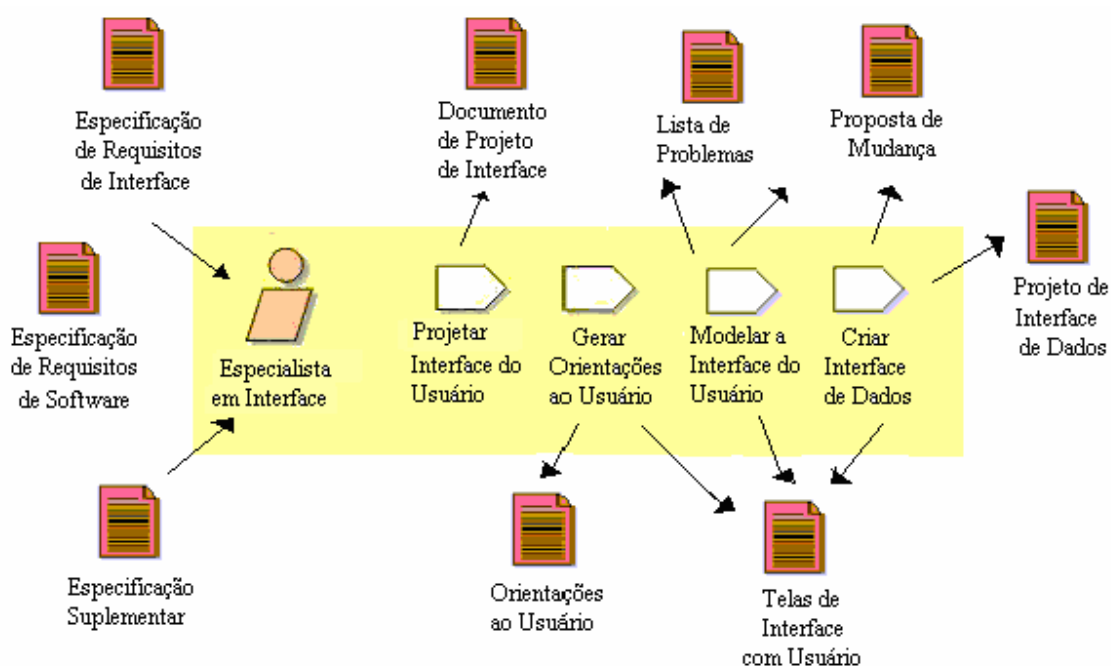


Figura 6.13 - Modelagem do papel e das atividades do especialista em interface.

O especialista em interface possui um conjunto de atividades associadas de acordo com seu perfil, e dentre elas está a coordenação e a construção da interface de dados do sistema de software do TA-2. Para este especialista é designada a criação da interface de dados, para a qual são definidas as principais abstrações do sistema de medidas, dando uma idéia de como os dados devem ser compostos e armazenados, para que possam ser processados pelo software de redução de dados.

O especialista de interface tem por função relacionar os dados com o projeto das estruturas de dados, devendo ter ciência das possibilidades de visualização dos dados importantes dos ensaios aerodinâmicos, procurando o aperfeiçoamento da interface em termos das informações a serem apresentadas, podendo ser de forma atrativa, mas que não induzam a erros de interpretação ou de entendimento do operador.

O fornecimento de orientação ao usuário advinda do projeto de interface, desenvolvida para o projeto do novo sistema de medidas, abrange a documentação fornecida com os subprodutos de software e as mensagens ou informações produzidas para o sistema de software, em forma textual ou gráfica. Todas as informações úteis e esclarecedoras para os usuários ou operadores do TA-2, passaram por um processo de qualidade, e os problemas identificados foram corrigidos, e praticadas as melhorias usando o processo de solução de problemas.

A princípio, a avaliação qualitativa das interfaces com usuário é feita com o operador ou usuário, através da análise do cumprimento dos requisitos, os quais devem corresponder às informações necessárias aos ensaios em túnel, contendo as apresentações nas telas dos dados, gráficos e resultados especificados para cada subproduto de software construído. Para a aceitação dos subprodutos de software, são previstos testes de qualidade mais criteriosos, com avaliação do aprendizado de seus usuários, para comprovar a facilidade de entendimento da interface (SHNEIDERMAN, 1998).

Outro papel definido é o arquiteto de instrumentação que se encarrega de especificar a interface de comunicação, selecionar um determinado número de dispositivos eletrônicos, especificar a arquitetura de hardware ou ambiente de ensaio integrado, para suportar os ensaios aerodinâmicos solicitados pelos clientes. Essas arquiteturas se tornarão os recursos mais importantes para obtenção dos dados e resultados, que são partes integrantes do sistema de medidas. As atividades e artefatos são mostrados na figura 6.14.

Para cada instrumentação especificada para o ensaio ou modelo aerodinâmico, pode ocorrer a calibração, tanto em laboratório quanto no próprio túnel. O papel do arquiteto de instrumentação é coordenar, inspecionar e testar o ambiente de ensaio integrado com todas as instrumentações, hardwares e dispositivos eletrônicos, interligados e comunicando de forma correta antes do pré-ensaio, e gerar a lista de controle de instrumentação.

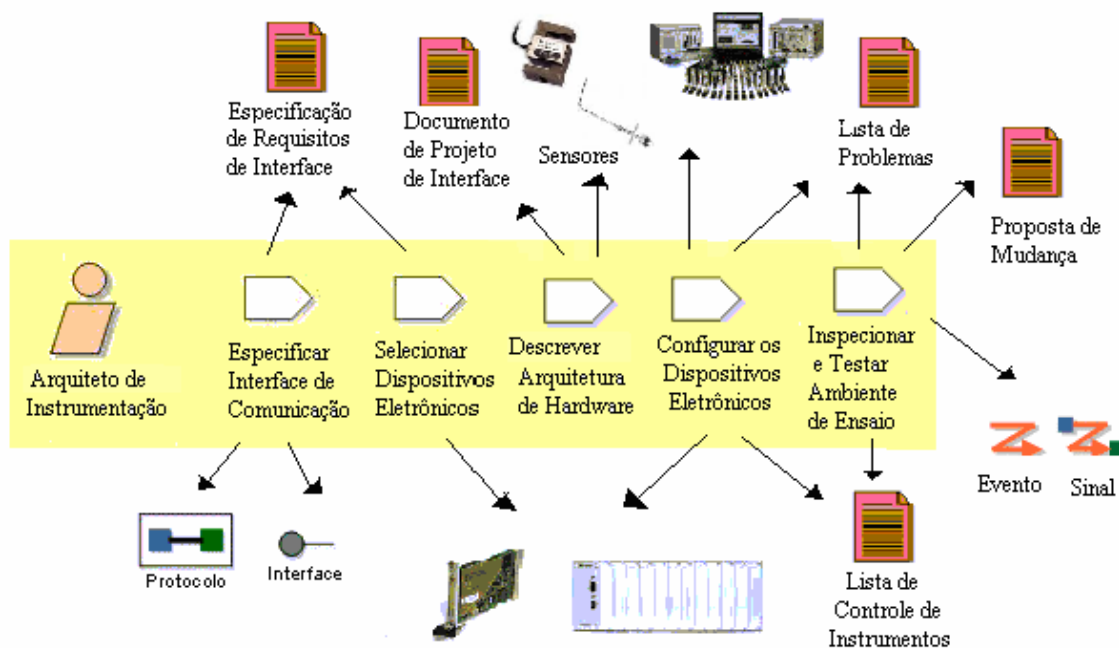


Figura 6.14 - Atividades e artefatos do arquiteto de instrumentação

O arquiteto de instrumentação é também responsável pelas especificações das interfaces de comunicação dos subprodutos a serem construídos, em cada iteração da fase de construção do ciclo de vida de desenvolvimento. Neste trabalho, trata-se das interfaces GPIB e DAQ aplicáveis através de *drivers* e seus protocolos, no desenvolvimento do software de aquisição de dados, interligados através de placas de aquisição e de outros dispositivos eletrônicos ao meio computacional, para aquisição de dados e controle das instrumentações.

O arquiteto de instrumentação é responsável por configurar os dispositivos eletrônicos para uso nos ensaios aerodinâmicos, conforme especificado pelo

fabricante, que pode ser através de *jumps* contidos na própria placa ou dispositivos especificados, ou por intermédio de funções específicas da ferramenta *LabView*.

A medida que são definidos os requisitos se inicia o projeto preliminar dos subprodutos de software, com uma arquitetura de hardware que vai sendo definida pelo arquiteto de instrumentação e adaptada para cada subproduto, até resultar em um projeto que possa ser implementado com os dispositivos eletrônicos ou componentes de hardware, que a princípio são projetados como componentes “*stand alone*”.

O resultado deste processo de projeto de hardware com a definição e adaptação de componentes, depende dos processos computacionais, que permitam que os dados sejam coletados com rapidez a partir dos sensores alocados no TA-2, e também possibilitar que o processamento da resposta do atuador associado ao subsistema seja realizado, para um funcionamento integrado entre o software e o hardware.

A arquitetura de hardware definida e adaptada para os subprodutos SSV e SBA é mostrada na figura 6.15, sendo que pode ser observada a adaptação com o reuso de uma mesma arquitetura, para o processamento da aquisição de dados dos sensores scanivalves e células de carga, e também para acionamento dos atuadores dos subprodutos SSV e SBA nas figuras 6.15 e 6.16 (SCXI 1160 e SCXI 1180). O reuso de alguns componentes de hardware também pode ser observado em outra arquitetura definida e adaptada, para o processamento dos dados usando o subproduto SCB na figura 6.17, considerado o sistema de software final do novo sistema de medidas do TA-2.

Com relação à adaptação da arquitetura de hardware projetada para o subproduto SBA, foi feita a substituição da placa de aquisição de dados “AT-MIO-64F5” pela placa “AT-MIO-16E-10” e a interligação das placas SCXI com o antigo filtro da plataforma HP. Além disso, as configurações dos canais das placas condicionadoras de sinais foram alteradas via proposta de mudança, conforme definido pelo arquiteto de instrumentação, e as instrumentações para o ensaio com este subproduto de software

foram relacionadas na lista de controle de instrumentação, para obtenção correta dos sinais dos sensores de células de carga e do resultado do ensaio aerodinâmico.

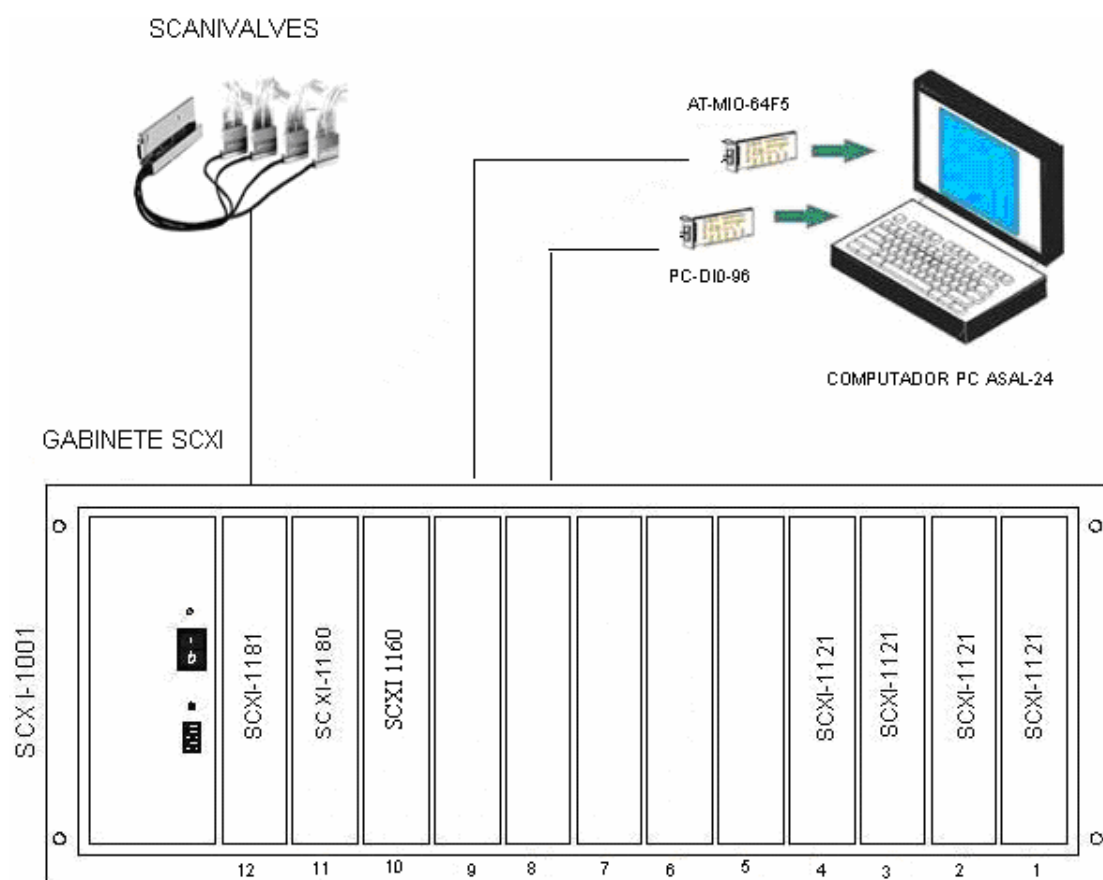


Figura 6.15 - Arquitetura de hardware do subproduto SSV e de parte do SBA.

As arquiteturas de hardware foram definidas e adaptadas para uso conjunto com subproduto SCB, através do levantamento dos sinais de estímulo e leitura associados ao controle e supervisão do subsistema motor e hélice, e de parte da balança, subestação e seção de ensaio, e também daqueles sinais de estímulo e leitura relacionados ao subsistema de aquisição de dados, sensoramento, balança e mastro.

Para o novo sistema de medidas são definidas duas arquiteturas de hardware compostas dos dispositivos eletrônicos adequados, os quais trabalham integrados ao software SCB, para o controle e aquisição de dados do TA-2. Estas duas arquiteturas podem ser vistas nas figuras 6.16 e 6.17.

Figura 6.16 - Arquitetura de hardware para sistema de aquisição de dados do TA-2

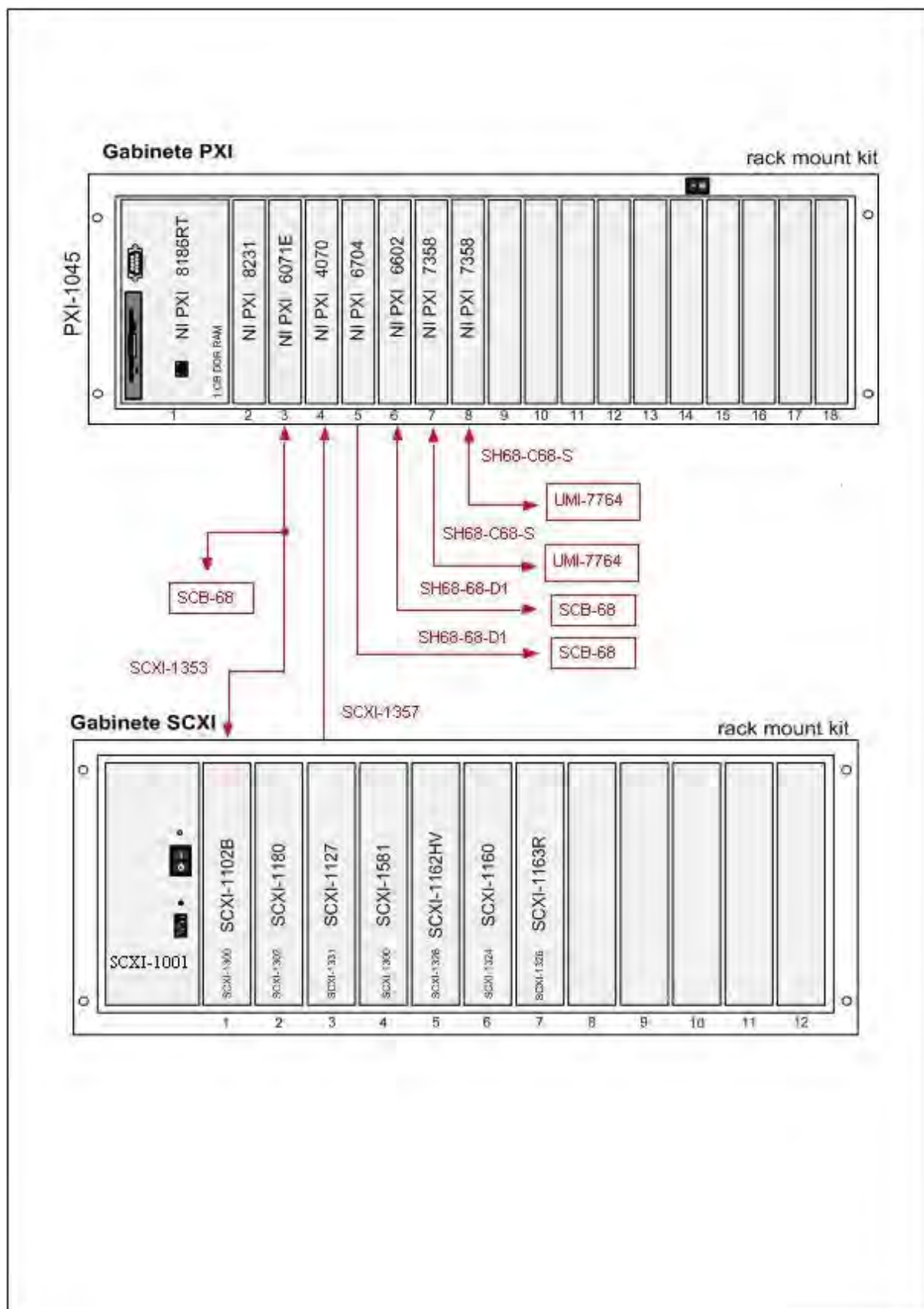


Figura 6.17 - Arquitetura de hardware para o sistema de controle do TA-2

Foram analisados cada um dos sinais de estímulo e leitura segundo as suas características, e alocados aos canais dos módulos ou dispositivos eletrônicos pertinentes, levando em consideração a característica do sinal ou estímulo, para relacioná-los ao tipo de condicionamento adequado. A arquitetura proposta baseia-se na plataforma *compactPCI (PXI)*, que permite uma flexibilidade para agregar novas funcionalidade e expansões. Associada a esta plataforma, estão os dispositivos eletrônicos para condicionamento de sinais, com recursos para os mais variados tipos de sensoriamento e coleta de dados das grandezas físicas do sistema de medidas do TA-2.

Pode ser vista no Apêndice B deste trabalho, a lista de controle de instrumentação relacionada ao subproduto SCB, e definida pelo arquiteto de instrumentação. Esta lista é composta dos dispositivos eletrônicos, tipos de sinais e subsistema de origem dos sinais agrupados em forma de tabela. Para a definição, adaptação e refinamento da arquitetura de hardware do SCB, se fez reuso da arquitetura anterior e do conhecimento com experimentos de aplicações similares, em projetos e implementações de sistemas de medidas no escopo industrial, e com a obtenção dos subprodutos das iterações anteriores da fase de construção.

Com a definição da arquitetura geral de hardware, se prossegue o desenvolvimento com o processo de refinamento desta, e também com o processo de projeto da arquitetura de software. Ambos processos de arquitetura representam um marco importante, para prosseguir com a implementação e testes dos subprodutos gerados, atividades nas quais será comprovado o cumprimento dos requisitos especificados para cada subproduto de software. Estes subprodutos possibilitaram a operação do TA-2, com as leituras dos sensores e os estímulos de atuadores dos subsistemas especificados, para realização dos ensaios aerodinâmicos.

As implementações de arquitetura de software representam um vínculo importante entre o projeto e os processos de engenharia, que neste trabalho dependem das arquiteturas de hardware e configurações das instrumentações da seção de ensaio,

modelo aerodinâmico e seção de controle do túnel de vento, aplicáveis para o desenvolvimento de cada subproduto. Para projetar e documentar explicitamente uma arquitetura de software do sistema de medidas do TA-2, é necessário conhecer e administrar a arquitetura de hardware, para permitir que cada subproduto de software cumpra seus requisitos.

Para cada subproduto de software desenvolvido durante o ciclo de vida de desenvolvimento de software foi gerado o documento de projeto de arquitetura. Este documento consiste em uma série de representações gráficas dos modelos aplicados aos subprodutos, com texto descritivo associados a eles. Foi adotado o fluxo de trabalho do RUP (RATIONAL CORPORATION, 2002) mostrado na figura 6.18, para obtenção da documentação e de diferentes modelos gráficos das diferentes perspectivas sobre a arquitetura dos subprodutos.

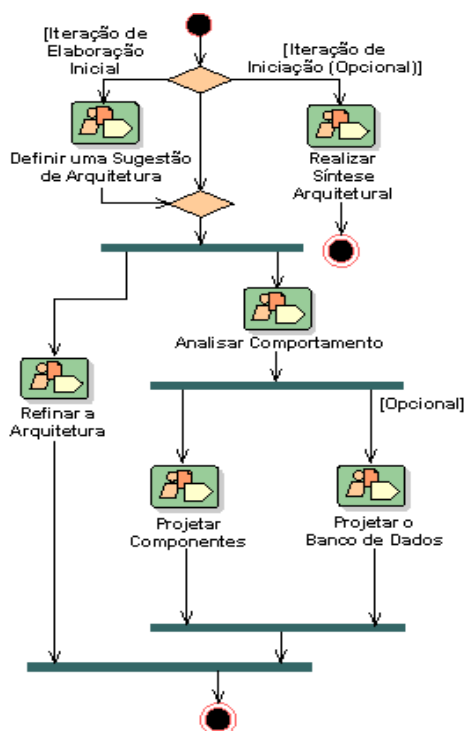


Figura 6.18 - Fluxo de trabalho de análise e projeto do RUP.

A função da etapa de análise e projeto é determinar a viabilidade do sistema, e avaliar as tecnologias possíveis para a solução. O fluxo se inicia com a atividade de realização de síntese arquitetural do sistema de medidas do TA-2, apresentado neste

trabalho no capítulo 3, como parte da fase de iniciação. Nesta etapa foram avaliados os riscos envolvidos no desenvolvimento, que foram dirimidos, principalmente aqueles quanto à compreensão do domínio e às novidades do sistema de software. Esta não foi considerada uma etapa opcional, devido à fragmentação do conhecimento e pouca documentação disponível do sistema de software.

A arquitetura para o sistema de medidas e seus subsistemas é desenvolvida através da atividade de definição de uma sugestão de arquitetura, e é enfatizada na primeira iteração da fase de elaboração, que fornece o ponto de partida para o trabalho de análise inicial. A arquitetura de cada subproduto de software é elaborada e produzida na segunda iteração da fase de elaboração e complementada e refinada nas iterações da construção. Na atividade de projeto preliminar é analisado o comportamento, com a criação de um conjunto inicial de elementos que fornecerão o comportamento apropriado dos subprodutos gerados para o sistema de software.

Na atividade de detalhamento do projeto são praticados os refinamentos necessários nos quais são analisados mais intensamente estes comportamentos, e projetados os componentes ou partes dos subprodutos de software, que fornecem o comportamento apropriado para atender aos requisitos do sistema. Assim se prossegue simultaneamente com as atividades de implementação e teste da fase de construção.

A proposta técnica, em relação à arquitetura de hardware e de software, é concluída e refinada pela equipe de desenvolvimento, com base na disponibilidade de pessoal, nos requisitos do cliente em termos de subprodutos necessários para os ensaios, na disponibilidade das ferramentas e dos produtos de terceiros (COTS) e nas necessidades de outros subprodutos do sistema de software. Estes subprodutos são desenvolvidos ou aplicados na fase de construção, e neste trabalho se descreve sucintamente cada subproduto e suas arquiteturas de hardware e de software.

No desenvolvimento do software de aquisição de dados foram previstas quatro iterações na fase de construção, ou seja, abrange as construções denominadas C1, C2,

C3 e C4. As práticas implantadas via METAPLI, provêem as seguintes atividades para que cada construção de software seja um subproduto operacional:

- Alteração e complementação das Especificações de Requisitos de Software no Documento pertinente da norma especificada.
- Projeto preliminar, detalhado e implementação das capacidades pertinentes a cada construção, com a modelagem e descrição no documento preconizado pela norma.
- Conclusão das propostas de testes de aceitação e aplicação dos testes para cada construção, e a documentação nos documentos preconizados pela norma.
- Realizar o acompanhamento dos problemas abertos em reuniões, revisões e testes do software.
- Realizar acompanhamento dos riscos identificados como abertos e controlá-los.
- Revisar os planos e documentos do projeto de software, como: documento de visão, especificação de requisitos, arquitetura de software, planos de desenvolvimento, garantia da qualidade, gerenciamento de configuração, plano de teste, e demais documentos e planos do projeto.
- Analisar a confiabilidade dos dados a serem armazenados em mídia, pelos critérios estabelecidos para cada construção.
- Rever os resultados alcançados e as lições aprendidas.

Para desenvolvimento dos subprodutos da fase de construção foi definido o papel de arquiteto de software, responsável por liderar e coordenar as atividades de arquitetura dos subprodutos de software, durante todo o desenvolvimento do novo sistema de medidas.

O arquiteto de software é responsável pelas atividades tais como: estabelecer a estrutura geral de cada arquitetura do subproduto a ser gerado, e também a decomposição com a verificação de reuso de arquitetura, o agrupamento dos elementos pertinentes ao subproduto de software de acordo com as interligações com hardware. Também deve formar a ordem das iterações sucessivas, cumprir os requisitos

estabelecidos para liberação de cada subproduto, refinar a arquitetura do produto final a cada iteração dos subprodutos entregues. As atividades e artefatos do papel do arquiteto de software podem ser vistas na figura 6.19.

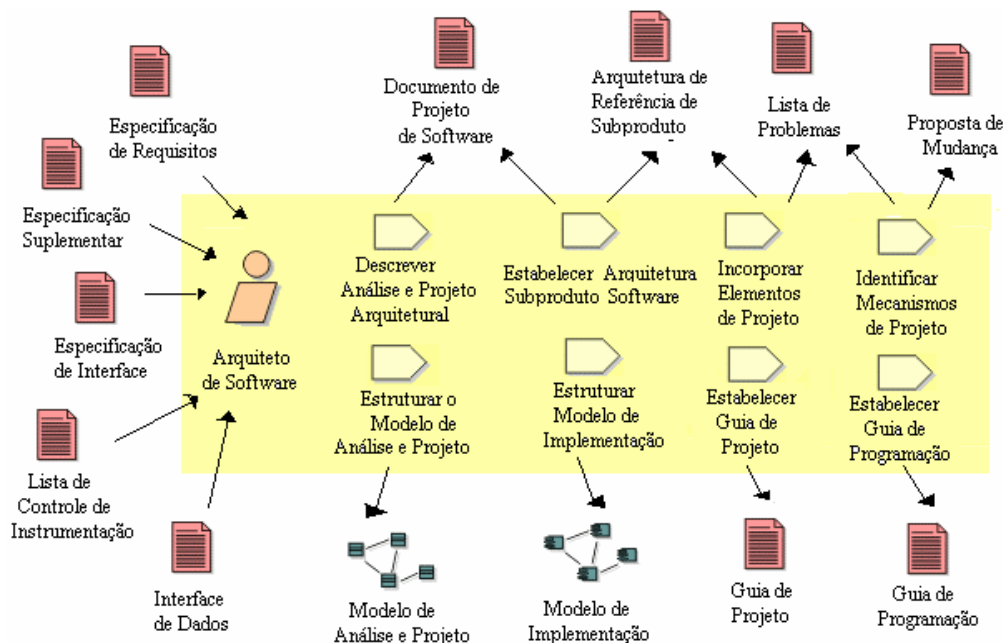


Figura 6.19 - Papel do arquiteto de software do novo sistema de medidas.

O projeto de arquitetura de software não distingue rigidamente a decomposição dos subsistemas em módulos ou subprodutos. Assim, suas funcionalidades permitiram o uso de modelos alternativos. Estes modelos transformam os dados de entrada em dados de saída, e também definem as operações em cada estado referentes ao subproduto, as quais podem ser implementados como componentes sequenciais ou como processos, com possibilidade de reuso de partes da modelagem de arquitetura, partes da “vi” e interligações de funções e sub-funções, em outros subprodutos.

A modelagem de requisitos, análise e projeto ou arquitetura de cada subproduto, definidos em projeto do sistema de software para a fase de construção do software de aquisição de dados, e processo de reengenharia do software de redução de dados, são descritas neste trabalho separadamente.

A descrição de arquitetura tem início com o software de aquisição de dados do Laboratório de Força (SLF) que é considerado um tipo de subproduto de software de

ensaios em túnel (SWE). Este usa um tipo de subproduto de software aplicativo de interface (SWI) abrangendo o software aplicativo de interface GPIB (SIG), com uma das modelagens representada pelo diagrama de contexto com elementos de interface, na modelagem foi dada ênfase ao reuso dos dados e funções estatísticas aplicáveis em calibração.

A seguir foi feita a descrição do software de redução de dados do túnel de vento (SRD) que é considerado um tipo de subproduto do software de validação da redução de dados (SWR), representado pelo diagrama de contexto com as funcionalidades necessárias e pelos diagramas de fluxo de dados de alto nível (DFD). Foi dada ênfase também no reuso dos dados obtidos pela plataforma HP, para os quais foi elaborado o software de transferência de dados da HP para o PC (STD).

Para o subproduto SRD foi aplicada a reengenharia dos dados, métodos matemáticos e funções aplicáveis em calibração para o cálculo dos coeficientes aerodinâmicos, com abordagem em reuso do conhecimento sobre os subsistemas do sistema de medidas e suas ligações com a mecânica dos fluídos, para obtenção da reestruturação dos dados e do projeto, e redocumentação do sistema de software.

Os demais subprodutos de software são descritos no detalhamento da modelagem de projeto, implementação e teste, sendo que os casos de teste aplicados ao software de redução de dados foram também detalhados, os quais representam os resultados finais deste trabalho após o desenvolvimento de todos os subprodutos. Este sistema de software integrado representa hoje o novo sistema de medidas do TA-2, que culminou na finalização do projeto de modernização.

6.4.2.1 Software de aquisição de dados do laboratório de força (SLF)

O SFL foi projetado para atender aos requisitos dos usuários ou operadores do processo de calibração do laboratório, que executam suas atividades manualmente

através da operação de medidas e registro dos resultados em papel, para posteriormente inseri-los em planilhas do aplicativo Microsoft Excel e obter os coeficientes de calibração. Os registros desses valores estavam sujeitos a erros, pois caso fossem executados por usuários não familiarizados, estes poderiam anotar valores discrepantes e digitá-los de forma incorreta no computador, além de poderem despertar dúvidas em relação aos cálculos a serem realizados quando em posse dos dados.

Para início da modelagem de requisitos, análise e projeto adotou-se o diagrama de contexto de cada subproduto do sistema de software. No diagrama do SLF e dos demais subprodutos, destacam-se os elementos essenciais ligados ao software de aquisição e redução de dados. Na representação destes diagramas constam as funções e interfaces, que cada subproduto de software irá ter em seu desenvolvimento, iniciando com o diagrama de contexto com os elementos de interface e funções do software de aquisição de dados do laboratório de força.

O SLF foi projetado usando o software de apoio ao desenvolvimento (SWA) *Microsoft Windows 3.1* (SOP), o *LabView* versão 3.1 (SAP), o software aplicativo de interface da *National Instruments* ou *driver* de interface GPIB (SIG), o protocolo de comunicação IEEE 488.2 e o multímetro HP3458A. Além destes acrescenta-se outros elementos e funções para a calibração de células de carga, interligados ao microcomputador PC com configuração mínima estabelecida em requisitos, contendo uma placa de aquisição GPIB do mesmo fabricante da interface, a qual permite a execução das operações necessárias para obtenção dos dados de calibração.

Para cada célula de carga calibrada são armazenados os resultados da calibração, dados estes que serão usados para comparação com os obtidos em cada componente de força ou momento da cadeia de medidas do túnel de vento. Os detalhes de partes deste subproduto podem ser vistos no diagrama de contexto da figura 6.20, o qual representa uma visão do contexto das interfaces.

O processo de projeto do SLF foi definido com base nas funções apropriadas para a iniciação, leitura, apresentação e armazenagem dos dados da calibração em tração e compressão, e para os cálculos de regressão, visando à obtenção dos coeficientes de calibração de cada célula de carga. As características deste software englobam o uso de cálculo estatístico e matemático para calibração das células de carga com massas ou cargas pré-definidas, que vão sendo acrescentadas na máquina de calibração e computados seus efeitos via subproduto de software interligado aos demais dispositivos.

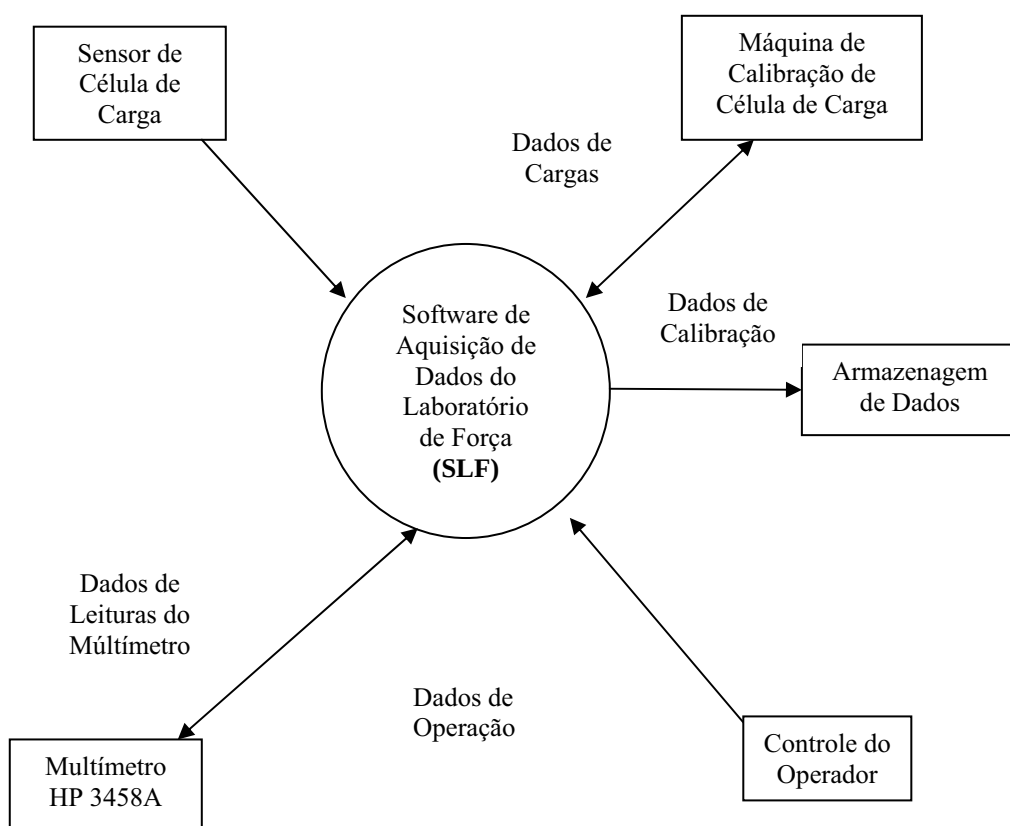


Figura 6.20 - Diagrama de contexto de interfaces do subproduto SLF.

A interface do sensor de célula de carga é responsável por obter dados das forças de tração e compressão do sensor em calibração. A interface do multímetro HP realiza a leitura destas forças em comunicação com o software. A interface da máquina de célula de carga é usada para fornecer os dados sobre a massa alocada durante o processo de calibração. A interface do controle do operador é onde transitam as

informações necessárias para o processamento dos dados da calibração e obtenção de resultados. A interface de armazenagem dos dados é usada para salvaguardar os dados de calibração, para reuso na calibração de forças e momentos pelo sistema de medidas do TA-2.

Usando modelagens em forma de diagramas de contexto, fluxo de dados entre outros, foi descrito o processo de calibração do SLF, incluindo a definição de arquiteturas deste subproduto usando os dispositivos, hardware e software, juntamente com os detalhes funcionais de outros tipos de serviço. Assim foi desenvolvido este subproduto para uso no laboratório de força e os demais subprodutos deste trabalho, os quais atendem aos requisitos do novo sistema de medidas do TA-2.

Os resultados dos coeficientes e de armazenagem do processo de calibração, obtidos com este subproduto de software projetado integralmente, podem ser vistos neste trabalho na modelagem de projeto detalhado, implementação e teste, a qual abrange as interfaces e funções para detecção do emprego das massas calibradas, montadas na máquina de calibração de célula de carga, para as medidas utilizando o multímetro HP3758A de 5 e meio dígitos e sua ligação a um computador pessoal, para uso da placa de aquisição AT-GPIB/TnT e o driver de interface NI-488.2 para transitar informações sob o controle do operador, e para obter dados do sensor de célula de carga.

6.4.2.2. Software de redução de dados do túnel de vento (SRD)

Na reengenharia do software de redução de dados é necessário obter as bases de dados das calibrações com as instrumentações, modelo aerodinâmico, dispositivos eletrônicos, e acessórios adequadamente instalados na seção de ensaios do túnel de vento. O processo de extração dos dados e o processamento das correções da pressão dinâmica, zero inicial, forças e momentos, atitude do modelo, tara e interferência, são macro-funções para o cálculo da matriz de calibração. A reengenharia foi aplicada

considerando todos os processos técnicos com seus métodos de cálculos utilizados no sistema anterior.

Estas macro-funções antes desenvolvidas para a plataforma HP eram executadas pelos programas “CALSENSORQ”, usado na calibração do sensor da pressão dinâmica, “PRDINREF”, para calibração da pressão dinâmica versus pressão de referência, “ZAB1701MC”, usado na obtenção da matriz ZAB, “ZQ1701MC”, para obter a matriz ZQ, entre outros que ajustam os valores obtidos em cada processo. Para entender melhor o processo de redução de dados através da METAPLI, se considerou a geração da documentação do software com modelagens via método estruturado.

Os resultados da redução dos dados têm relação com as abordagens técnicas dos subsistemas de medidas, e também com o tratamento dos dados de calibração. Este tratamento dos dados ou redução inclui métodos de calibração, para a correção dos valores e obtenção da matriz de calibração, que contribuem para obtenção dos coeficientes aerodinâmicos. Para estes cálculos se faz uso dos coeficientes de calibração das células de carga, pressão dinâmica, temperatura, entre outros, que neste trabalho são calculados e comparados com os obtidos pela plataforma HP, e também pelo software de aquisição do laboratório de força, os quais são armazenados em forma de matriz, como por exemplo, a denominada “Matriz K”.

Nesta atividade aplicaram-se métodos como para calibração da balança externa [NOGUEIRA, 1980], pelo qual se obtém a matriz de calibração, válida por um longo período de tempo de ensaio aerodinâmico, e outros métodos de calibração para ajustes e correções dos dados aerodinâmicos. O software de redução de dados foi desenvolvido abrangendo os cálculos especificados pelos especialistas, relacionados com as cargas e pressões aerodinâmicas, que após passar pelo subsistema de redução de dados, fornecem os coeficientes aerodinâmicos documentados nos relatórios finais, entregues aos clientes. Esses ajustes e correções são processados com os seguintes elementos:

- Dados de zero inicial.
- Dados do efeito da pressão dinâmica (ZQ).
- Dados de variação da atitude do modelo (ZAB).
- Dados de forças e momentos.
- Dados de tara e interferência (TARA).

Os dados do zero inicial são adquiridos sem a presença do vento, através dos sensores de célula de carga do subsistema de sensoriamento em conjunto com os subsistemas balança, mastro, modelo e aquisição de dados. O objetivo desta correção é descontar os valores residuais que podem ser variáveis durante os ensaios, em relação as forças e momentos das diversas configurações e atitudes do modelo aerodinâmico. Os valores de zero inicial são armazenados em arquivos de dados, que são lidos e processados por um componente de software, gerado com a reengenharia do software de redução de dados.

No trabalho de reestruturação dos dados foram criados o dicionário e o repositório ou depósito de dados, os quais estarão disponíveis para os demais componentes de software do subsistema de redução de dados. Procurou-se organizar os dados e gerar a estrutura de variáveis, de forma que pudessem expressar as funções de ajustes e de correções para os quais estaria sendo executado o processamento, mas esta forma de projeto e de programação não é considerada neste trabalho com componentes orientados a objetos.

A correção do efeito da pressão dinâmica (ZQ) é feita através dos dados adquiridos do tubo de pitot padrão, instalado na posição do modelo aerodinâmico, com o vento gerado pelo subsistema motor e hélice em conjunto com os subsistemas sensoriamento e balança. Trata-se das medidas simultâneas do pitot padrão e dos pitots fixos na seção de ensaio, que juntamente com a medida da carga da balança, constituem dados os quais se ajustam as curvas de carga versus pressão dinâmica, através do método dos mínimos quadrados, que relacionam a carga medida no ensaio

com a variação da pressão dinâmica, gerando os dados armazenados em arquivos disponíveis para os componentes do software de redução de dados.

Com a variação dos ângulos, altera-se a posição do centro de gravidade do modelo em relação à balança, processo que envolve os subsistemas de sensoriamento, balança, aquisição de dados e mastro, denominado Zalfa para atitude em α , e Zbeta para atitude em β , denominado ZAB, o conjunto das duas atitudes. Este ensaio é feito com o modelo instalado no TA-2 e sem a presença do vento. Nos cálculos de correção da atitude usando os dados obtidos na calibração, são processados métodos de ajustes e regressões para posterior armazenagem em arquivos em forma de matriz, que são então usados para a correção dos dados do ensaio em relação a atitude.

Para os cálculos das forças e momentos são utilizados os arquivos de dados contendo as matrizes da saída dos sensores, a “Matriz K” e a “Matriz E”, com as quais são ajustados as forças e os momentos e processadas as correções. O processo de ajuste para obtenção da matriz de calibração é através de método para calibração de medidas de múltiplas componentes, aplicável para balança externa. A matriz de calibração e os resultados de ajustes de forças e momentos usando o SRD serão apresentados como caso de teste deste trabalho.

Também faz parte de ajustes e correções de dados o processo do sistema de transferência e de adimensionalização. Os cálculos do sistema de transferência são utilizados tanto para ensaios com ângulos de ataque em alfa quanto em beta. A transferência trata os dados com relação ao centro de referência dos momentos aerodinâmicos. Os valores são medidos na posição do centro de gravidade (CG) do túnel de vento, que via cálculos são transferidos para a posição do CG de fixação do modelo, e posteriormente sendo processados os cálculos dos momentos com referência à posição do CG especificado pelo cliente, o qual é o centro de referência do momento.

Os especialistas em ensaios em túnel apresentam os resultados das forças aerodinâmicas através de coeficientes adimensionais, que são calculados em função das relações geométricas do modelo aerodinâmico, do número de Mach, e do número de Reynolds. Estes cálculos influenciam de forma relevante na obtenção dos coeficientes aerodinâmicos, em que os coeficientes de forças aerodinâmicas são adimensionalizados pela pressão dinâmica e pela área de referência do modelo. Para estes cálculos se faz uso das coordenadas do modelo obtidas dos clientes juntamente com o programa de ensaios.

Os dados de ensaios e de calibração foram reestruturados em um repositório de dados específico no software para PC, pois o programa em BASIC denominado “RED170FS2” por exemplo, era alterado em cada ensaio para modelos aerodinâmicos diferentes. Este processo de alteração muitas vezes era esquecido causando erros nos resultados da redução, ou após alterá-lo se deveria gerar uma nova versão do programa em questão, que seria compilado e executado novamente apenas pela mudança de alguns dados.

O processo de correção de tara e interferência é iterativo e são necessários os dados de carregamento de tara para o ajuste dos carregamentos da calibração. Neste processo são ajustados os valores baseados em resultados dos ajustes anteriores. O software de redução efetua os cálculos de tara e interferência, os quais são processados para tirar o efeito dos mastros e ajustar os coeficientes aerodinâmicos às condições de referência aerodinâmica. São medidas as forças e pressões atuantes no modelo durante os ensaios, para os cálculos de tara de algumas configurações que possam interferir de forma relevante na obtenção dos coeficientes.

O software de redução de dados de um ensaio em túnel de vento constitui um subproduto complexo, e dependente dos resultados dos processos e dados anteriores. Estes processos podem ser vistos nos diagramas de contexto e de fluxo de dados. O diagrama de contexto mostrado na figura 6.21 representa as funções do software de redução de dados, no qual são identificadas as dependências do subproduto de

software e as relações com os dados produzidos, transferidos ou armazenados, os quais o sistema de software processa de acordo com os requisitos, sendo o processo representado pelo círculo com a declaração do principal objetivo da redução de dados. Este diagrama de contexto em sua arquitetura geral é representado através de blocos, modelados com a abordagem abstrata dos propósitos declarados deste software com relação ao sistema de medidas do TA-2.

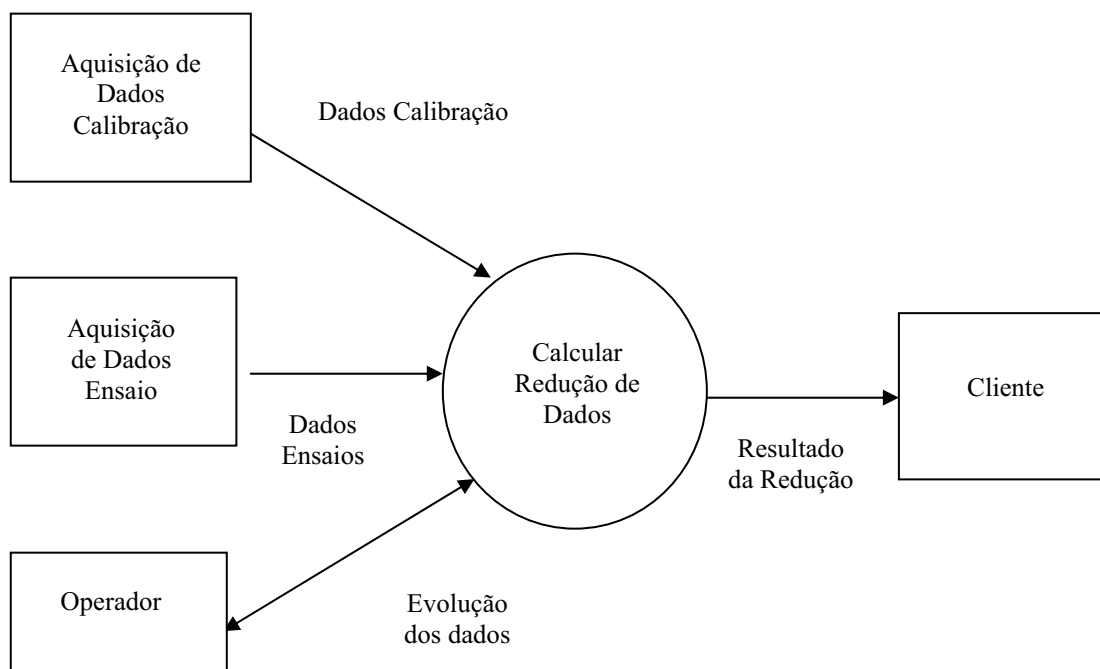


Figura 6.21 - Diagrama de contexto do software de redução de dados do túnel de vento

O único processo do diagrama de contexto do software de redução de dados é “Calcular Redução de Dados” que representa a função puramente lógica deste processo, ou seja, a declaração funcional que é desenvolvida na estrutura do projeto.

Os diagramas de fluxo de dados são maneiras de representar como são processados os dados pelo SRD, nos quais a notação utilizada representa o processamento funcional, o repositório de dados e os movimentos dos dados entre as funções ou processos. Estes dados são transformados a cada etapa de processamento, e são mostrados nos diagramas das figuras 6.22 a 6.25, para documentar o projeto do software de redução de dados.

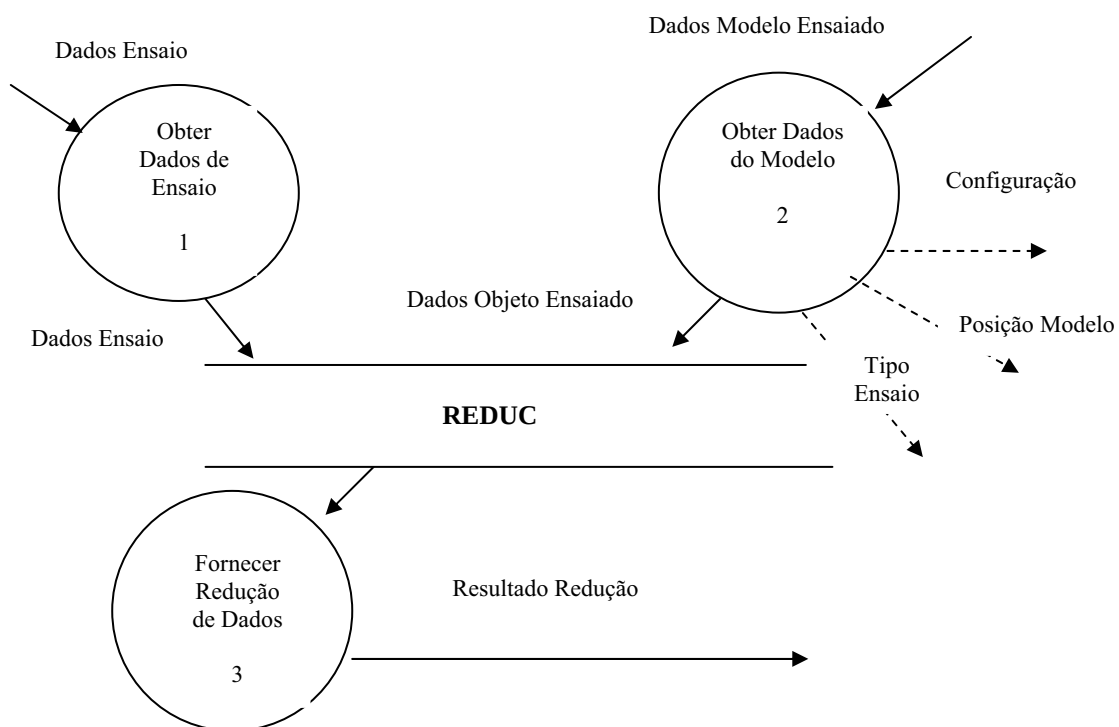


Figura 6.22 - DFD0 – Calcular redução de dados.

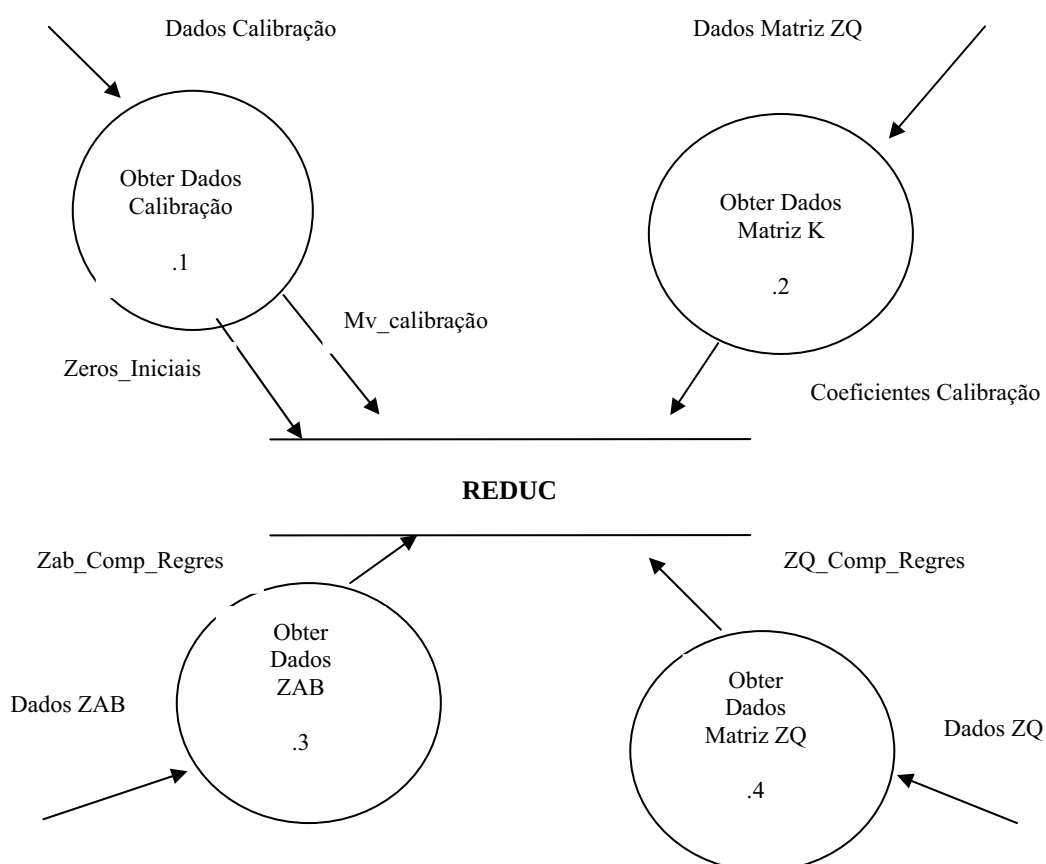


Figura 6.23 - DFD1 – Obter dados de ensaio.

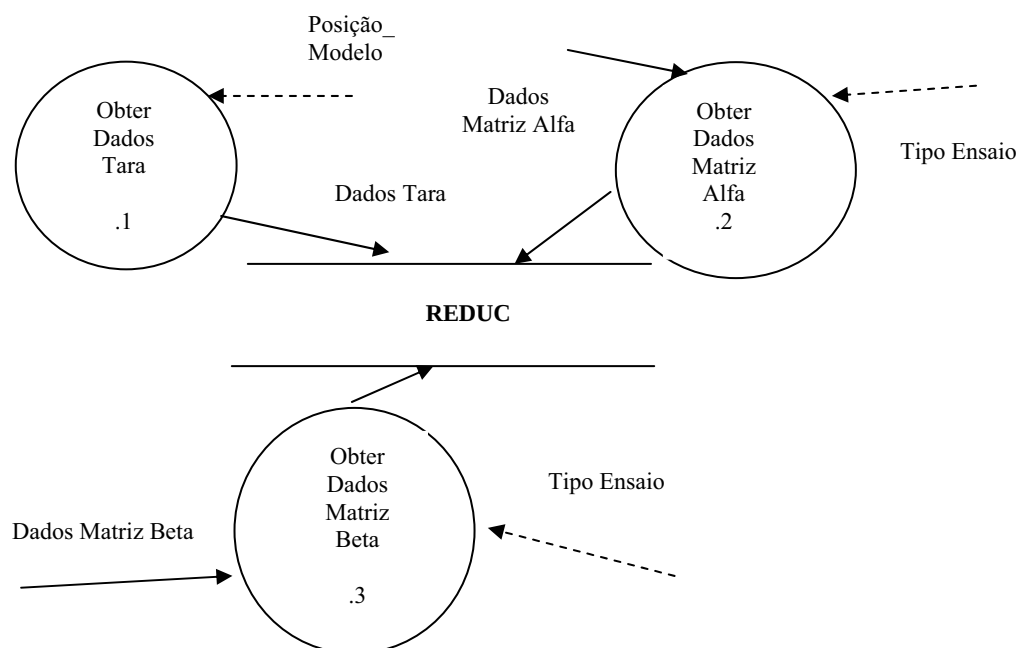


Figura 6.24 - DFD2 – Obter dados do objeto ensaiado.

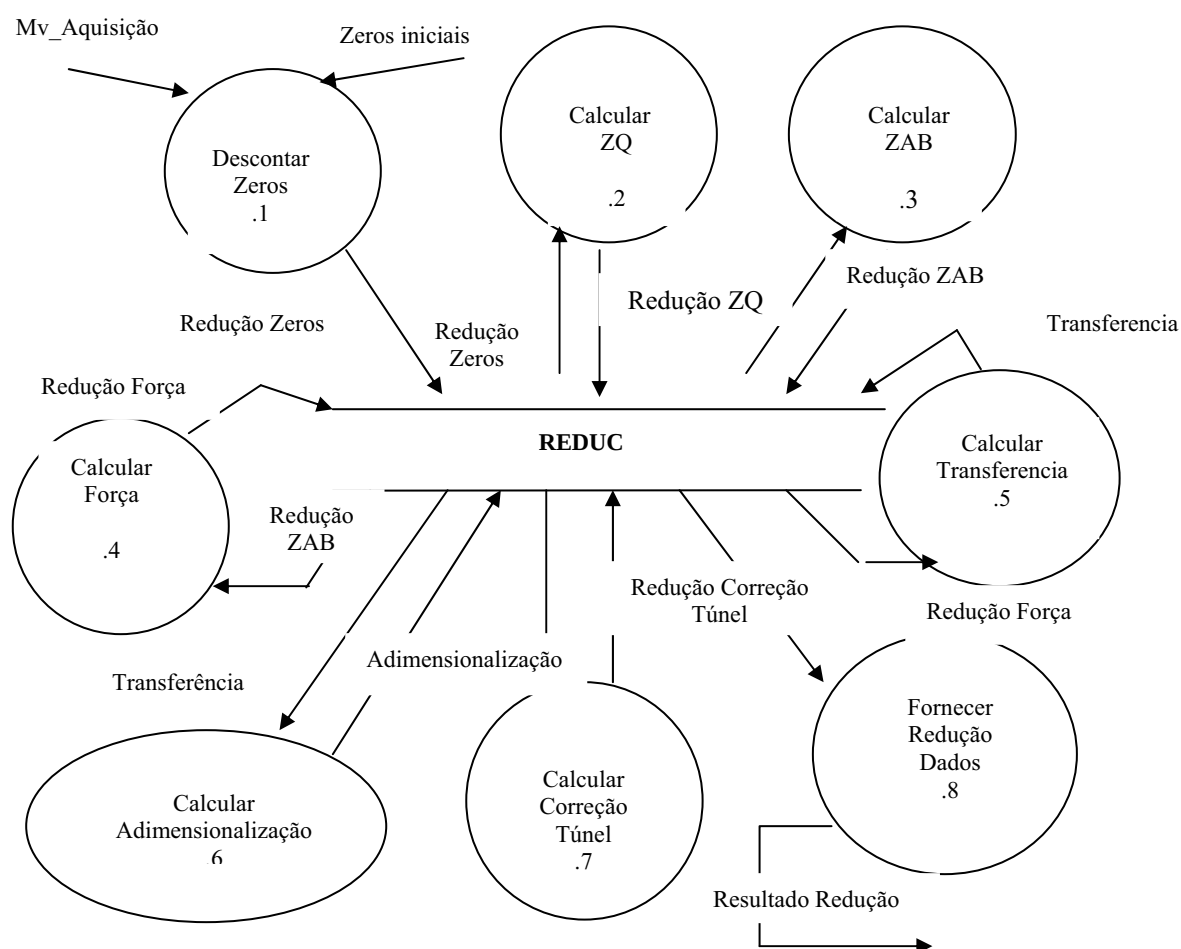


Figura 6.25 - DFD3 – Fornecer redução de dados.

As figuras de 6.22 a 6.25 mostram os processos envolvidos no processamento de dados do software de redução de dados do TA-2, os quais foram obtidos na reestruturação do projeto, através de engenharia reversa dos códigos existentes em linguagem BASIC. Com a identificação dos dados de cada processo, se gerou a reestruturação dos dados contidos no repositório ou depósito de dados. O processo “Calcular a Redução de Dados” faz interface com o operador e com os dados de ensaio e de calibração no modelo geral, o qual é desmembrado em três outros processos constantes dos diagramas de fluxo de dados. Estes processos indicam o repositório denominado “REDUC”, no qual estão armazenados os dados associados a cada processo.

Os diagramas de fluxo de dados “Obter Dados de Ensaio”, “Obter Dados do Objeto Ensaado” e “Fornecer Redução de Dados” mostram as perspectivas funcionais em níveis mais detalhados, dos quais são gerados outros níveis até que representem uma única função, de modo a detalhar como é descrito o processamento dentro do software de redução de dados, para possibilitar a tradução do código em linguagem BASIC para a linguagem C.

Todos esses diagramas de fluxo de dados fizeram parte da atividade de análise e projeto, do software de redução de dados. Cabe lembrar que para o desenvolvimento deste software, aplicou-se a técnica de reengenharia, portanto a documentação é considerada a atividade mais importante, principalmente as informações sobre os cálculos que constam no projeto do diagrama “Fornecer Redução de Dados”.

Para a obtenção dos dados visando a recuperação rápida daqueles ensaios armazenados com a plataforma HP, e a validação do software de redução de dados foi construído o software de transferência dos dados da HP para o PC (STD), o qual possibilitou passar os dados de aquisição, calibração e algumas matrizes já processadas pelos softwares da HP. O objetivo principal da construção deste produto é a armazenagem dos dados para recuperação de resultados de ensaios, com uso da plataforma PC.

O subproduto STD foi desenvolvido em linguagem de programação C, usando os recursos da interface e protocolo GPIB, interligando as plataformas HP e PC, para a transferência dos dados. Hoje a plataforma HP está desativada, entretanto os dados dos ensaios que foram executados estão recuperados e salvos em mídia, trabalho este executado pelo STD usando a plataforma PC. O diagrama de contexto do STD da figura 6.26 descreve de forma simplificada o ambiente deste subproduto de software.

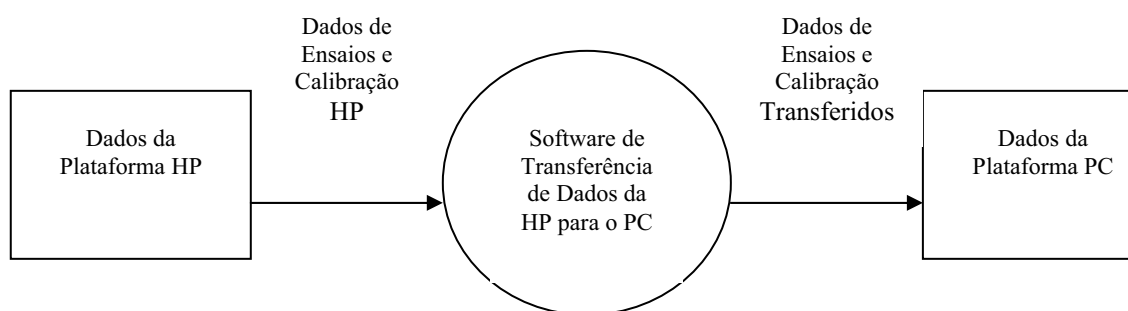


Figura 6.26 - Diagrama de contexto do STD

Na figura 6.26 observamos que o STD está conectado aos depósitos de dados das plataformas HP e PC, que produzem ou consomem os dados de ensaios e calibração. Estes dados quando transferidos para a plataforma PC são processados pelo SRD ou armazenados em mídia, visando garantir a recuperação e reuso dos dados de serviços prestados de ensaios no TA-2. A maior contribuição do STD para o novo sistema de medidas foi na disponibilidade dos dados, possibilitando os testes e a obtenção de resultados para validação do subproduto de software de redução de dados.

Para possibilitar o reuso ou a reengenharia em projetos futuros, a documentação associada deve auxiliar o desenvolvedor a compreender e adaptar o sistema de software a uma nova aplicação. Os diagramas, fluxogramas, fluxos de trabalho e outras práticas, auxiliam ao desenvolvedor com pretensões de reuso e reengenharia, a localizar as informações sobre onde os componentes ou funções foram utilizadas, e sobre os problemas que tenham sido encontrados.

A visão gerencial e técnica contida nas documentações deste trabalho, implantada via METAPLI, foram consideradas reuso do conhecimento. A obtenção da documentação de cada subproduto de software, na qual consta a descrição da aplicação específica de forma abstrata, produz informações que permitiram que partes dos subprodutos e artefatos fossem selecionados, visando o reuso do conhecimento que, quando combinadas, permitiram a um desenvolvedor compor e controlar as abstrações para o novo aplicativo ou no caso do projeto do TA-2, para um novo subproduto.

Os critérios adotados e implantados via METAPLI, para análise e seleção de componentes de software ou subprodutos previamente desenvolvidos, e disponíveis em bibliotecas de funções de aplicativos, com objetivo de reuso ou de alterações para torná-los reutilizáveis, são os seguintes:

- Facilidade de acesso ao subproduto ou componente.
- Funções e interfaces providas.
- Demonstração da operação correta.
- Condições de instalação, preparação, treinamento e uso.
- Identificação e o registro pelo gerenciamento de configuração.
- Condições de manutenção incluindo as possibilidades de alteração.
- Restrições de copyright.

A garantia da qualidade, voltada para o reuso de software, é provida pela atividade de análise da durabilidade e validade dos métodos e ferramentas usadas no desenvolvimento do software, visando o uso futuro. Para cada componente ou subproduto selecionado para reuso são avaliados:

- A validade ou melhoria operacional.
- O estado de documentação gerencial e técnica.
- A qualidade funcional, ou seja, não conformidades, complexidade, confiabilidade dos dados, entre outros.

Esses critérios e avaliações abrangem todas as etapas do ciclo de vida de desenvolvimento, às quais são aplicadas as atividades de reuso e de reengenharia. A visão para o reuso do conhecimento é mais evidente quando finalizadas as modelagens do projeto detalhado, implementação e teste dos subprodutos de software, etapas em que as funcionalidades de cada subproduto foram comprovadas.

6.4.3 Modelagem de projeto detalhado, implementação e testes

Para modelagem de análise e projeto do sistema de software foram decididas quais as capacitações ou certas funções do sistema de medidas que seriam implementadas via software, e quais via hardware especializados ou dispositivos eletrônicos e suas interligações, para cada subproduto de software a ser construído. Partes da arquitetura geral de cada subsistema foram descritas no capítulo 3 deste trabalho, no qual constam as descrições simplificadas da arquitetura e da visão dos subsistemas para a aquisição e redução de dados.

Neste trabalho, para facilitar a disponibilidade operacional do túnel de vento, foram definidos alguns papéis, os quais têm por finalidade atender a solicitações de serviços de ensaios aerodinâmicos. Todos os subprodutos de software foram desenvolvidos ou construídos sob coordenação e controle destes papéis especificados. As atividades e os artefatos de cada papel específico são para o cumprimento do programa de ensaio aerodinâmico, para o qual os subprodutos de software são desenvolvidos ou experimentados, e considerados aceitos para o propósito operacional.

Demandou-se um esforço considerável para o projeto detalhado da arquitetura do sistema de software, a qual foi usada nas construções dos subprodutos de software específicos. Estes subprodutos quando implementados irão processar a troca de dados e de controle com os subsistemas. Para obter este projeto de arquitetura detalhado foram implantados os seguintes passos via METAPLI:

- Definição de arquiteturas do sistema de medidas com os dispositivos eletrônicos, os hardwares, e componentes ou subprodutos de software.
- Análise para melhorias nas arquiteturas em cada iteração da fase de construção do subproduto.
- Desenho esquemático com os componentes dessas arquiteturas.
- Refino das arquiteturas para abranger os subsistemas e seus subprodutos de software.

A modelagem desta arquitetura detalhada do sistema de software do TA-2, obtida no final da fase de elaboração, aborda cada subproduto de software a ser desenvolvido. Esta foi obtida seguindo os passos estabelecidos anteriormente e pode ser vista na figura 6.27. As funções referentes aos comandos dos atuadores, monitoramento dos subsistemas, aquisição de dados dos sensores, leituras e armazenagem de dados, entre outras, são implementadas em cada subproduto de software apresentado nesta arquitetura detalhada, provendo a disponibilidade do TA-2 para operação, conforme as necessidades de ensaios solicitadas pelos clientes.

O processo de projeto detalhado e de implementação do sistema de software utiliza as abordagens deste trabalho, para o desenvolvimento dos subprodutos de software, as quais minimizam os erros humanos e auxiliam na descoberta de defeitos no sistema de medidas, antes de colocá-lo em pleno uso. À medida que os subprodutos de software são colocados em funcionamento para os ensaios em túnel de vento, são observados os problemas através dos testes para encontrar os defeitos e comparação de resultados com os obtidos usando a plataforma HP. A aplicação do reuso, da reengenharia e dos testes, auxiliam a tornar o sistema de software mais confiável.

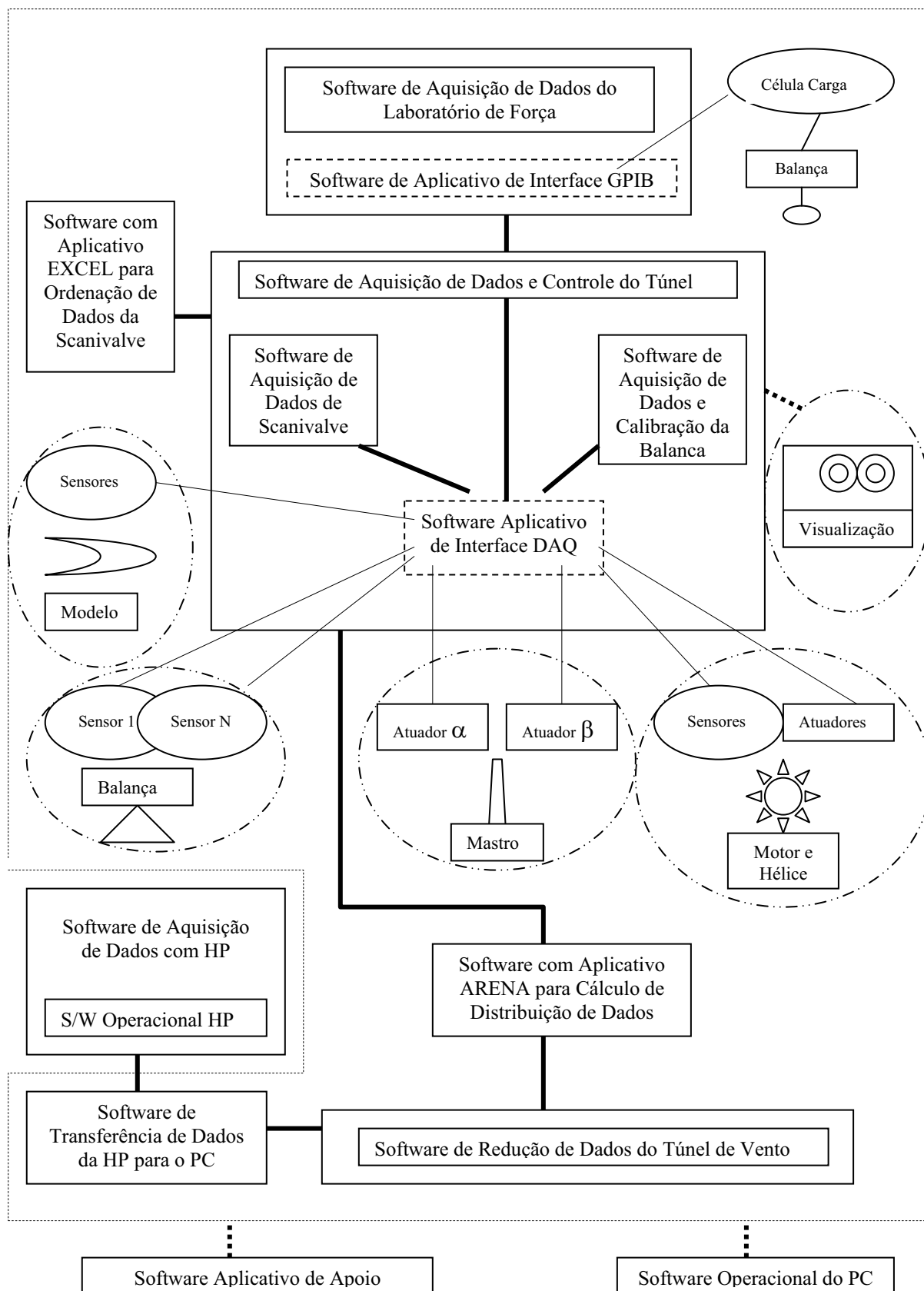


Figura 6.27 - Arquitetura detalhada do sistema de software.

Neste trabalho, com a inviabilidade do uso de ferramentas CASE para executar a modelagem do projeto detalhado, e a codificação em diagramas de blocos do ambiente de programação visual comercial, se fez uso dos recursos da própria ferramenta *LabView*, para a modelagem dos subprodutos de software de aquisição de dados, como uma equivalência ao diagrama de estado de um projeto detalhado da programação orientada a objetos, os quais são codificados considerando estas equivalências com uso de botões e rótulos visuais para representar as mudanças de estado.

Para execução deste trabalho é definido o papel do implementador, o qual é responsável pela codificação e implementação dos subprodutos ou componentes, com execução de atividades e geração de artefatos referentes às construções dos subprodutos de software e testes das operações destes com os subsistemas. Estas atividades requerem uma atenção especial nos testes com relação à aquisição e monitoramento dos dados, as instalações do modelo aerodinâmico, a calibração de instrumentos no laboratório de força e outras atividades. A figura 6.28 mostra as responsabilidades do implementador.

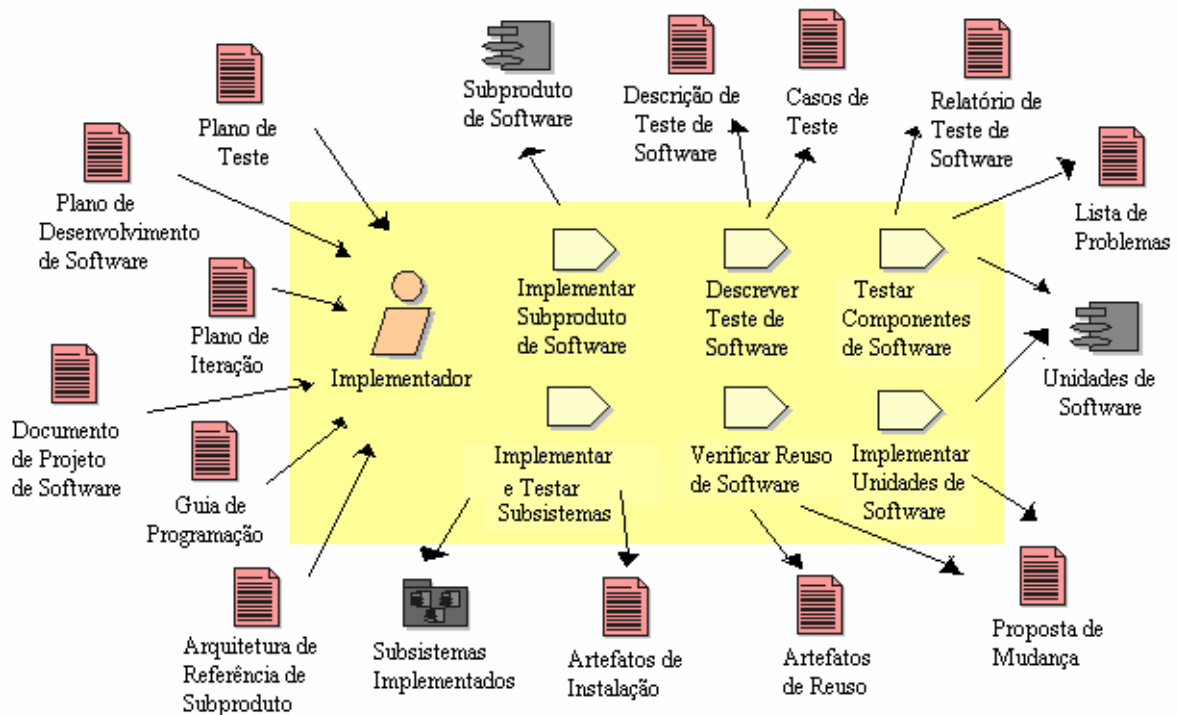


Figura 6.28 - Papel do implementador no TA-2.

O implementador é responsável pelas atividades de codificação e testes de partes ou unidades dos subprodutos, como por exemplo, daquelas referentes às configurações do modelo de ensaio, identificação dos ensaios, carregamentos das forças, entre outras partes de cada subproduto de software especificado. Estas partes dos subprodutos com uso da ferramenta *LabView* correspondem às unidades com as extensões “.vi” e “.ctl”, que irão compor os códigos fontes do subproduto completo.

Para a modelagem do projeto detalhado em conjunto com a implementação dos subprodutos de software de aquisição de dados, foi adotado o próprio código de programação em *LabView* como um modelo de diagrama de estados, nos quais as funções e os elementos que formam o código são definidos, e através da verificação da ocorrência de um possível evento, disparar uma transição de estado, conforme mostra a figura 6.29.

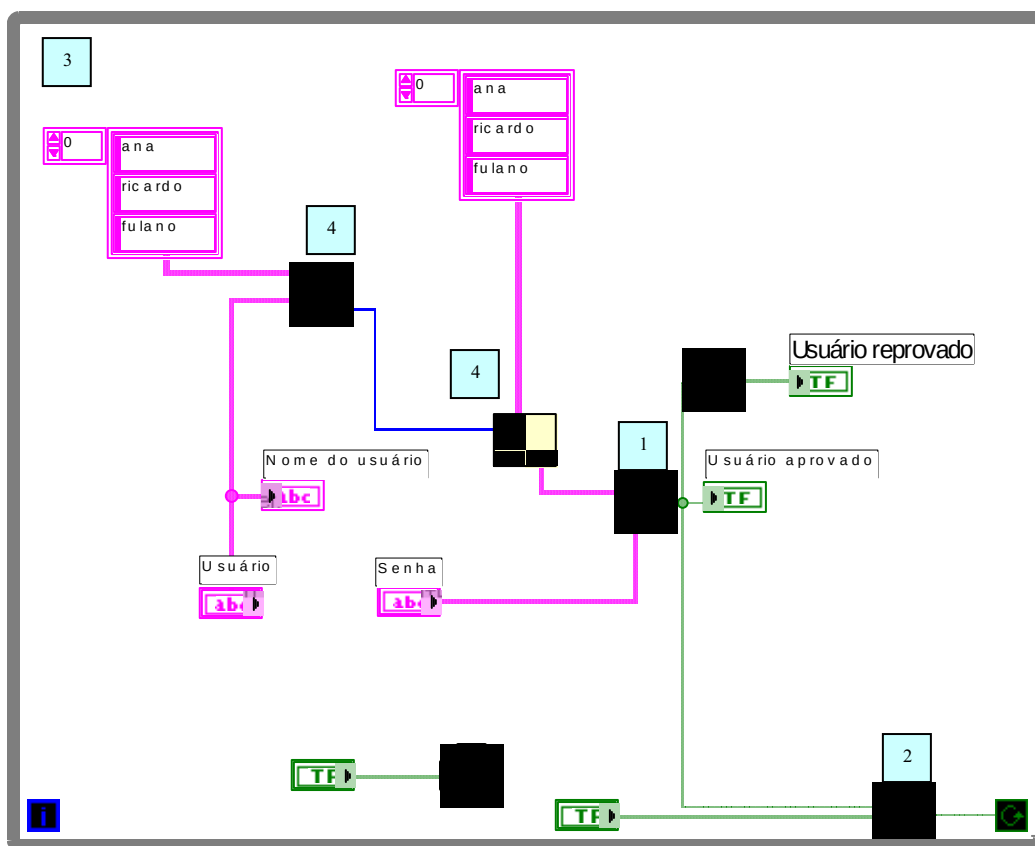


Figura 6.29 - Diagrama de estado para os subprodutos SWE em *LabView*.

A figura 6.29 mostra uma parte funcional do subproduto SSV denominado “Confere usuário e senha.vi”, a qual é a codificação da equivalência de estados de parte do subproduto de software, ou seja, aquelas programações que constituem uma “vi” ou “ctl” do *LabView*. Este exemplo é usado para ilustrar os estados ou condições que levam a uma transição para um estado diferente a partir de um estímulo, que são definidos conforme o seguinte:

- 1 Estado inicial para conferência de usuário e senha
- 2 Condição de parada que define o estado final para esse diagrama de estado
- 3 Estrutura “*While*” que define os limites do diagrama de estados, no qual O código em seu interior será executado ciclicamente, até que ocorra a Condição de parada.
- 4 As atividades a serem executadas neste estado

Esta modelagem de projeto para mudança de estado equivale aos botões e rótulos, que representam os estímulos contidos nos subprodutos de software, que quando acionados forçam a transição de um estado para outro, conforme mostrado na figura 6.30, em que os estímulos são representados pelos botões com rótulos “ENTRE” e “DESISTO”. Esta operação da mudança de estado acontece assim que o operador acionar os botões rotulados, movimentando o *mouse* até que indique estar sobre o botão virtual e pressionando-o, quando então a operação é habilitada e assim é alterado o estado inicial, ao qual se retorna depois de finalizada a operação, ou através da escolha de outra operação.

TÚNEL AERODINÂMICO - ENSAIO

Usuário

Senha

ENTRE

Usuário aprovado

Usuário reprovado

Nome do usuário

DESISTO

Figura 6.30 - Diagrama de estado com botões em *LabView*.

O uso de funções disponíveis em conjunto com a ferramenta *LabView*, facilitou o desenvolvimento dos subprodutos de software de aquisição de dados do túnel de vento, ou seja, o desenvolvimento dos subprodutos das construções C1, C2, C3 e C4. As funções disponibilizadas através de bibliotecas de funções, com códigos escritos e testados, possibilitaram aos desenvolvedores a escolha daquelas necessárias, e também a adaptação, criação, modificação e implementação de códigos complementares, para compor e completar o desenvolvimento do novo software de aquisição de dados do TA-2, com as seleções daquelas funções com características de reusabilidade, para auxiliar no desenvolvimento dos códigos e nas evoluções de uma iteração da fase de construção para a próxima.

Muito esforço foi feito para que o sistema de medidas do TA-2 e seus subprodutos de software atingissem a funcionalidade esperada o mais rápido possível, para que a implementação obtivesse sucesso nos detalhes de operação de um ensaio em túnel de vento. As mudanças dos subprodutos entre uma iteração e outra da fase de construção seguiriam o processo de controle de mudança, no qual são gerados as lista

de problemas e proposta de engenharia, e submetidas ao revisor de engenharia para apreciação e avaliação.

Para facilitar a implementação ou codificação dos subprodutos de software da construção C2 em diante, um dos critérios estabelecidos via METAPLI é que se investigasse o reuso, baseando-se nas principais funções usadas para a leitura dos sensores, comunicação entre hardwares e instrumentações, configuração dos canais dos dispositivos, estímulos dos atuadores, apresentações em tela dos principais dados para monitoramento, usadas no subproduto anterior. Estas devem ser definidas de forma clara, ficando a critério do desenvolvedor esta definição, com adaptações e refinamentos em consenso com seus parceiros, para compor as unidades dos subprodutos.

Como já descrito anteriormente, a declaração formal de unidade como uma UML (Unified Modeling Language), não era suportada pelo *LabView* versão 3.1, razão pela qual para este trabalho foram criadas as unidades de forma manual. Estas unidades se constituem dos chamados “controles”, armazenados em arquivos “.ctl”, e de outras unidades, armazenadas como arquivos “.vi”, todos agrupados dentro de bibliotecas “.llb”. Para a codificação de cada uma destas unidades se adotou o critério estabelecido pela METAPLI para o reuso de funções.

Todos os papéis definidos para a modelagem de requisitos, análise e projeto, implementação e teste dos subprodutos de software, têm como responsabilidade liderar e coordenar o processo de desenvolvimento, até que se cumpram os requisitos estabelecidos no projeto do novo sistema de medidas. A demonstração do cumprimento destes requisitos é feita através dos documentos, dos códigos e resultados de testes gerados. A maioria destas demonstrações depende das programações das funcionalidades do sistema de software constante em cada unidade, e das delimitações deste sistema implementadas nos subprodutos.

Todo o desenvolvimento dos subprodutos foi feito visando a melhoria da qualidade com a minimização de defeitos do sistema de medidas, e a disponibilidade operacional deste sistema durante todo ciclo de desenvolvimento. A figura 6.31 ilustra algumas unidades contidas na biblioteca “TA-2_SV.llb”, codificadas no subproduto de software de aquisição de dados de *scanivalve* (SSV).

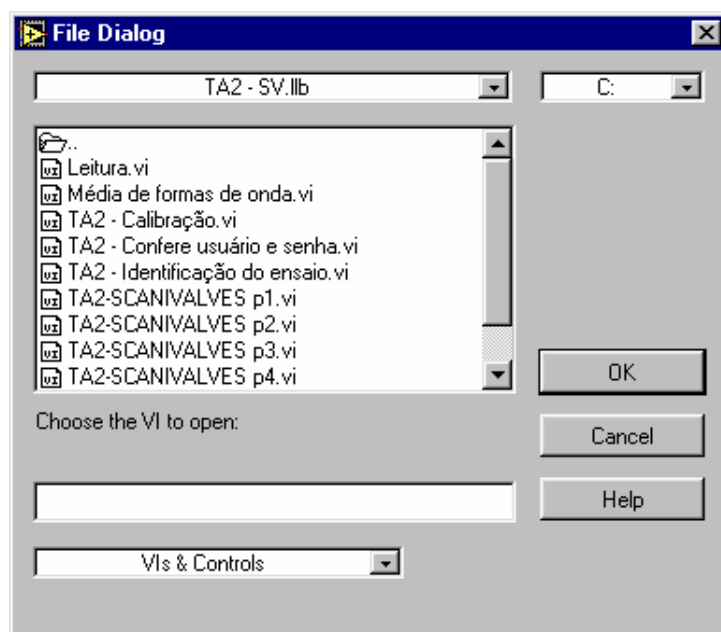


Figura 6.31 - Declaração de unidades do software de aquisição de dados de *scanivalve*.

Neste trabalho são codificadas pelo implementador, diversas unidades para compor os subprodutos, dentre as quais aquelas com características de reuso. Estas são definidas e adaptadas pelos desenvolvedores para o reuso nos demais subprodutos. Dentre estas unidades para reuso, este trabalho destaca aquelas que lêem sinais ou dados, as que armazenam dados em forma de planilha e as que processam o cálculo de ganho.

Os reusos dos softwares para leitura de sinais ou dados, e armazenagem de dados em forma de planilha, podem ser observados na codificação das unidades do subproduto SSV, as quais correspondem às unidades “leitura.vi” e “TA-2-Scanivalves P1.vi”, para aquisição de dados de pressão. Estas unidades do subproduto SSV foram

adaptadas e refinadas para o reuso no subproduto SCB, e podem ser observadas nas figuras 6.32 e 6.33.

1D Linear Evaluation	vi	Encoder RS-485 ver2	vi
6528_task_alarme	vi	envia_status_dadosvi	vi
Add Action	vi	Error Code Database	vi
alarme_6528	vi	ErrWarn	ctl
aquisicao_suplementar	vi	eventvkey	ctl
Auto Nulling	vi	fila_ensaio_camp	vi
baixo_nivel_close	vi	fila_ensaio_camp_fechar	vi
baixo_nivel_config	vi	fila_inicial_ensaio_camp	vi
baixo_nivel_init	vi	Find First Error	vi
baixo_nivel_read	vi	Find Tag	vi
BuildHelpPath	vi	Format Message String	vi
built_pc_cmd	vi	fuso_sentido	vi
calculo_ganho	vi	General Error Handler CORE	vi
calculo_K_zeragem	vi	General Error Handler	vi
calculo_Pulso	vi	gera_angulos_simulados	vi
Calibracao Encoder RS485	vi	Geracao Pulsos_Zeragem Balaca_mod	vi
Check New	ctl	Gerenciador_ver3	vi
Check Special Tags	vi	Get Text Rect	vi
checksum	vi	GetHelpDir	vi
Clear Errors	vi	grava_fine_course	vi
Close File+	vi	Gravar PSI	vi
close_6602	vi	Gravar PSI_MEDIA	vi
comp_generation	vi	Init RS-485 Slin Mode	vi
comp_zerada	vi	Interface Hardware	vi
compara_zeragem_balança	vi	Interface Usuario	vi
Conf Suplemetar	ctl	KB	vi
Config Channel Suplementar	vi	Leitura RS-485	vi
Config Channel Suplementar_array	vi	Ler Arquivo PSI	vi
config_6602	vi	ler_alto_nivel	vi
Config_AS_task	vi	ler_mux	vi
Configura_ganho	vi	ler_psi	vi
configura_hw	vi	Linear Fit Coefficients	vi
Configura_Null	vi	Linear Fit	vi
Controla_tempo	vi	Longest Line Length in Pixels	vi
Convert property node font to graphics font	vi	media_alimentação	vi
Convert to String	vi	media_componentes	vi
Counter Gate (NI-T10)	vi	media_componentes_alto_nivel	vi
Define_Dev2_task	vi	media_mux	vi
Details Display Dialog	vi	mens_zeragem	vi
Device 1_leitura	vi	Mensagem Usuario	vi
Device 2_leitura	vi	Merge Errors	vi
DialogType	ctl	MSE	vi
DialogTypeEnum	ctl	Mux_index	vi
elimina_fusos	vi	Mux_pressao	vi
Encoder RS-485 ver2	vi	Mux_pressao_Calibracao	vi
		Mux_pressao_last_version	vi

Figura 6.32 - Declaração de unidades do software de aquisição de dados e controle do túnel de vento.

Mux_pressao_last_version_mod	vi	Table to File	vi
Mux_pressao_novo	vi	TaqReturnType	ctl
Mux_pressao_valores	vi	task_1160	vi
Not Found Dialog	vi	Tela Conf Suplementar	vi
Novo Valor	vi	temp Daq CH Manipulado	vi
Open File+	vi	tempo PSI	vi
Operate Nulling	vi	teste DAQmx valid device	vi
Passagem Parametros Variant	vi	teste	vi
PB		teste_da_6602vi	vi
QSMTemplate	vi	Three Button Dialog CORE	vi
queue second	vi	Three Button Dialog	vi
Read File+ (string)	vi	Trat_dados_Array	vi
Read From Spreadsheet File	vi	Trat_dados_wave	vi
Read Lines From File	vi	Trim Whitespace	vi
Read_curva_mux	vi	USer event Demo	vi
Read_encolder	vi	Verifica Canais Selecionados	vi
Read_fine_coarse	vi	VISA Configure Serial Port	
Read_K_zeragem	vi	VISA Configure Serial Port (Instr)	vi
Search and Replace Pattern	vi	VISA Configure Serial Port (Serial Instr)	vi
Separa Canais Selecionados	vi	whitespace	ctl
Set Bold Text	vi	Write File+ (string)	vi
Simple Error Handler	vi	Write To Spreadsheet File	vi
Slin translate	vi	write_6602	vi
subSCXI Multi-Chan Null - volts with excitation	vi	Write_encolder_curve	vi
subSCXI Multi-Chan Null - volts with excitation_mod	vi	Write_fine_coarse	vi
subSCXI Multi-Chan Set Potentiometer - volts with excitation	vi	Write_K_zeragem	vi
		Write_mux_curva	vi

Figura 6.33 - Continuação da declaração de unidades do software de aquisição de dados e controle do túnel de vento.

Na implementação e teste do subproduto de software de calibração do laboratório de força (SLF) são apresentadas diversas unidades implementadas e testadas, os quais apresentam como resultados os coeficientes de calibração das células de carga. O reuso destes dados da calibração é o foco principal deste subproduto de software da construção C1, que também contribuiu para o exercício das práticas da METAPLI e a gestão do conhecimento, para a construção dos demais subprodutos.

Os detalhes das implementações e testes de unidades de leitura de sinais ou dados de pressão observados no subproduto SSV, apresentadas na figura 6.31, são descritos no tópico de implementação e teste do subproduto de software de aquisição de dados com Scanivalve, complementado com a implementação e teste do subproduto de software com aplicativo EXCEL para ordenação de dados da Scanivalve (SVE).

O software de transferência de dados da HP para o PC (STD) foi implementado e testado, juntamente com o software de aquisição de dados e calibração da balança

(SBA) da construção C3. O STR é implementado seguindo as ações do processo tradicional. No entanto, faz uso dos dados adquiridos com SBA e com o programa em *BASIC* “AQU1701MC” da HP. Esses dados transferidos são depois processados pelo software de redução de dados do túnel de vento (SRD), para cálculo da matriz de calibração, cujos resultados são apresentados pelo tópico implementação e teste do subsistema de redução de dados.

As implementações e testes das unidades refinadas para cálculo de ganho apresentadas nas figuras 6.32 e 6.33, as quais foram codificadas no subproduto SBA, são apresentadas no tópico de implementação e teste do subproduto de aquisição de dados e calibração da balança (SBA). E o reuso destas unidades, nas implementações e testes do subproduto SCB, é apresentado no tópico de implementação e teste do subproduto de software de aquisição de dados e controle do túnel de vento (SCB).

6.4.3.1 Implementação e testes do subproduto de software de aquisição de dados do laboratório de força - SLF (C1)

Na implementação do software de aquisição de dados do laboratório de força (SLF) com o ambiente de programação visual comercial *LabView*, foram obtidos os diagramas de dados ou códigos e os painéis ou telas deste subproduto. Fez-se uso de funções estatísticas, de comunicação com o multímetro, funções de leitura e armazenagem dos dados, entre outras, alocadas especialmente para a tecnologia de medição das forças aerodinâmicas.

Este subproduto de software foi desenvolvido na primeira iteração da fase de construção (C1), na qual se aplicou reuso de funções da biblioteca disponíveis na ferramenta, como *driver* para comunicação GPIB com o multímetro HP3458A, e de várias outras funções, com as quais foram desenvolvidas as diversas unidades deste subproduto, apresentadas em forma hierárquica na figura 6.34.

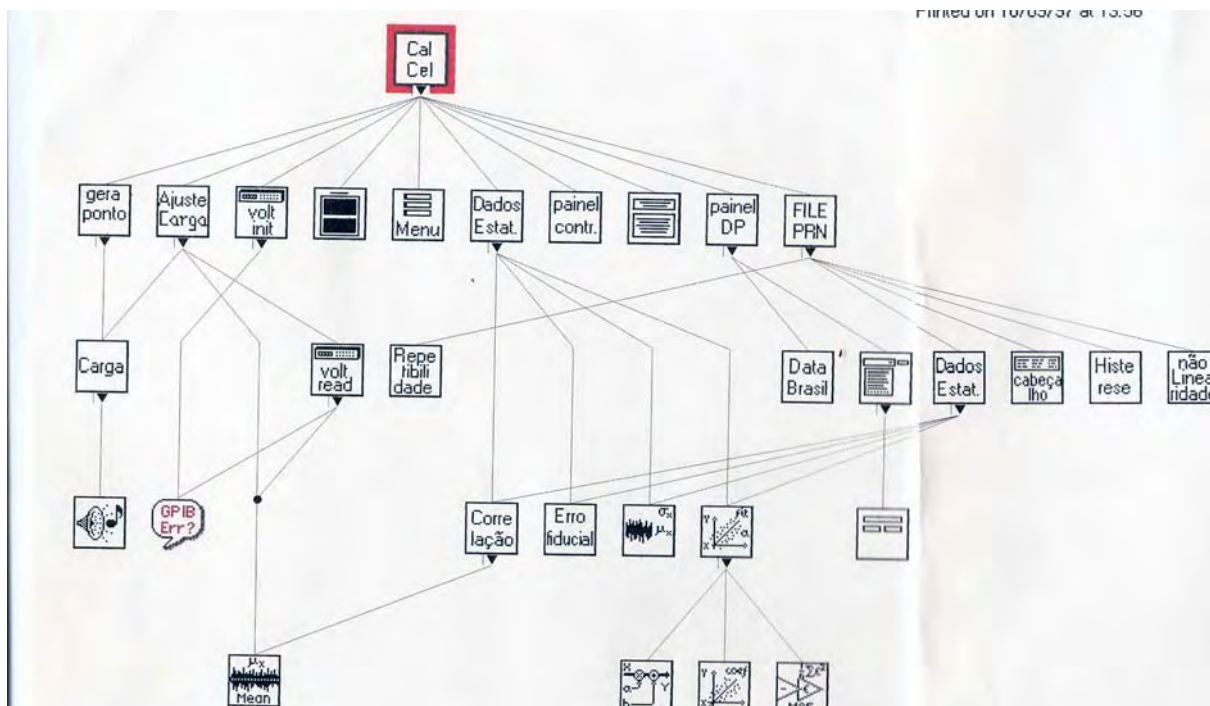


Figura 6.34 - Hierarquia de unidades do subproduto SLF.

Os coeficientes de calibração das células de carga são obtidos através destas unidades, por intermédio de definições de cargas e usando hardwares com interfaces de software devidamente instaladas, para uso com SLF. Este software tem por função registrar, armazenar e processar os dados a fim de calcular os coeficientes de calibração, que são usados como referência nos ensaios do TA-2, para comparação com aqueles obtidos na calibração da balança externa com essas mesmas células de carga.

A execução da calibração é conduzida através dos painéis virtuais do SLF, com inserções de informações sobre as células de carga, tipos de calibração entre outras, pertinentes a cada unidade, com as quais são obtidos os coeficientes de calibração reutilizados pelo subproduto de software de aquisição de dados da balança. Partes do subproduto SLF são mostradas nas figuras 6.35 a 6.39, as quais incluem as unidades para processar as funções para cálculo da média de uma série de leituras, ajustes de curva de força versus sinal de saída, cálculo de correlação, determinação de erro fiducial, entre outras unidades implementadas.

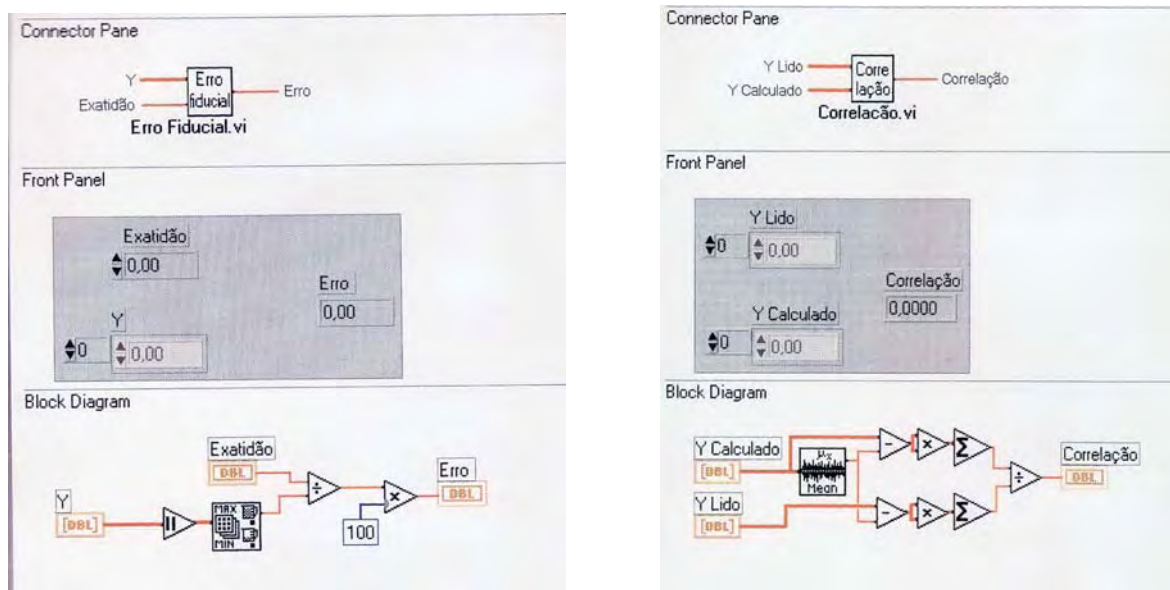


Figura 6.37 - a) Unidade de cálculo de erro fiducial b) Cálculo de correlação

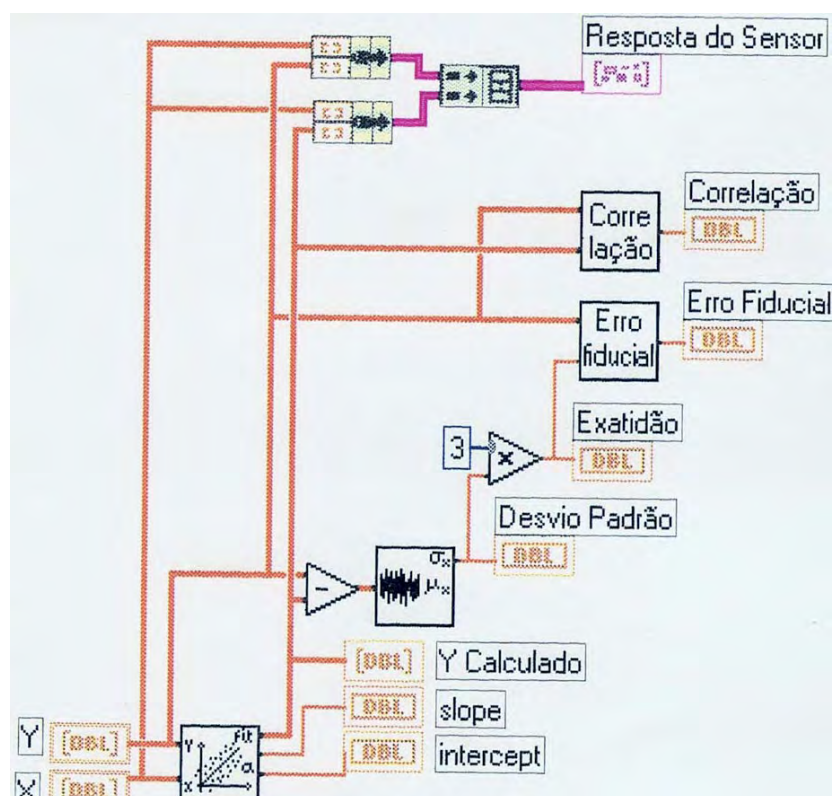


Figura 6.38 - Unidade de software de cálculo de ajuste de curva

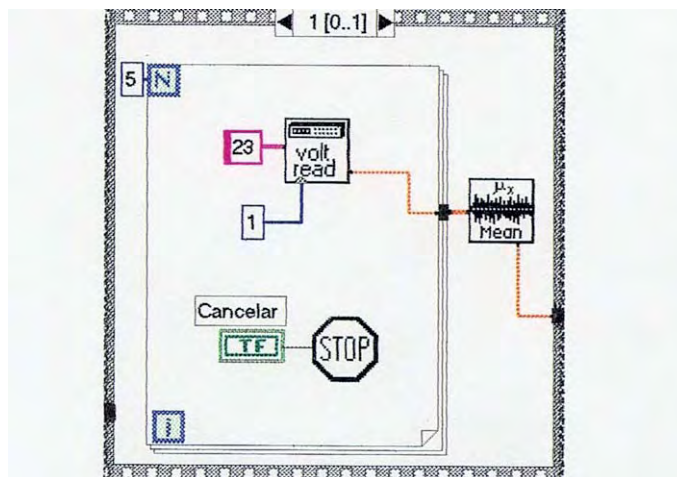


Figura 6.39 - Unidade de software de leitura dos sinais do multímetro.

Na construção do SLF, se confirma a relação da METAPLI com ações do processo unificado em relação ao envolvimento de pessoas na programação e revisão em par do processo de calibração, para resolver o problema específico de calibração de célula de carga. Com a análise do resultado dos coeficientes é evidente que o processo de calibração automático através de um mecanismo computacional e de documentação, é melhor compreendido pelo usuário e assim é possível minimizar os erros humanos nas demais construções dos subprodutos.

O desenvolvimento deste subproduto auxiliou na familiarização dos desenvolvedores com as diversas funções do ambiente de programação visual comercial, através de implementações de funções para leitura e armazenagem de dados, acionamento do multímetro e cálculos estatísticos. A sua implementação contribuiu para validação dos processos de engenharia aplicáveis na calibração de células de carga, que servem de base para a codificação dos demais subprodutos. Assim com a implementação deste subproduto se obteve:

- A documentação dos cálculos estatísticos aplicáveis.
- A definição do processo para calibração de células de carga.
- A automatização deste processo, com registro e armazenagem dos dados de calibração.
- A qualidade assegurada dos dados.

- A documentação dos dispositivos e hardwares envolvidos.
- Os dados dos coeficientes de calibração mais corretos e precisos para os ensaios de modelos aerodinâmicos com as células de carga calibradas no laboratório de Força.

Os testes neste subproduto foram aplicados usando o critério de comparação dos resultados com células de carga com libras iguais. Foram testadas primeiramente todas as unidades individualmente, com tensões inseridas por intermédio de uma fonte de alimentação e informações textuais. Por ser o primeiro subproduto gerado foram concebidas diversas versões de cada unidade, até a obtenção daquelas consideradas uma versão completa pelo implementador, neste trabalho considerando sempre dois elementos da equipe, as quais são integradas e aceitas pelo revisor de engenharia.

Os coeficientes obtidos com aplicação de um teste para a calibração de células de carga, com o subproduto SLF, são mostrados na tabela 6.1. Estes coeficientes são comparados com os obtidos com cada célula de carga na calibração da balança externa do TA-2, e são armazenados no arquivo que contém a matriz K. As capacidades das células de carga são apresentadas em libras (lb) e segue o especificado pelo fabricante.

Tabela 6.1 - Coeficientes obtidos com SLF.

Capacidades (lb)	Coeficientes (mV)
100	-1,567723
100	-1,568747
100	-1,586274
300	-4.532817
250	-3.591140

Pode-se observar nesta tabela que os coeficientes obtidos com células de carga com libras iguais, apresentaram os coeficientes com valores aproximados, o que demonstra atingir o objetivo de um teste de software, aplicado pelo implementador ao

subproduto SLF, por intermédio do documento de descrição de teste de software. Obviamente valores aproximados a estes para libras iguais devem se repetir com o processo de calibração da balança externa, onde são instaladas estas células de carga.

6.4.3.2 Implementação e testes do subproduto software de aquisição de dados de *scanivalve* – SSV (C2)

Este subproduto SSV foi construído para obtenção de dados das *scanivalves* instaladas para tomadas de pressão de perfil NACA 0012 e usado na validação do modelo aerodinâmico de um cliente. A arquitetura de hardware que faz interface com o SSV, definida pelo papel do arquiteto de instrumentação e apresentada anteriormente, inclui a identificação das placas da *National Instruments* alocadas para aquisição dos dados, de cada canal das *scanivalves*, e de demais sensores e instrumentações envolvidas. As identificações das placas, canais e sensores são usados para conferência dos dados advindos do modelo aerodinâmico em ensaio.

Foi prevista a construção deste subproduto para resolver outro problema específico, este solicitado pelo cliente, para aquisição de dados de pressão visando à validação de um modelo aerodinâmico. Para esta implementação são usadas funções para leitura dos sensores de pressão e para atuação das *scanivalves*. É prevista a análise e verificação de reuso de funções ou unidades, em que seriam avaliadas as partes reutilizáveis dos processos de comunicação com os dispositivos de hardware, comando de atuação das *scanivalves*, aquisição de tomadas de pressão, aquisições de calibrações, entre outros componentes ou unidades de software as quais são propensas a serem reutilizáveis.

Alguns exemplos destas unidades, com seus painéis e diagrama de blocos são mostradas nas figuras de 6.40 a 6.42, e foram codificadas e testadas pelo implementador, com a verificação de reuso de funções também voltadas para a identificação dos ensaios. As informações sobre os ensaios são requisitos do cliente

que deveriam ser cumpridos, e também a armazenagem em arquivo de parte desta identificação com respeito aos ensaios aerodinâmicos.

TÚNEL AERODINÂMICO - ENSAIO

Título do Ensaio:

Descrição do Ensaio:

Responsável Técnico:

CONTINUE

Arquivo de Dados Criado:

Data: Horário:

CONFIRME **RE-INICIE**

Nome do usuário:

Título do ensaio:

Figura 6.40 - Unidade de software de identificação do ensaio.

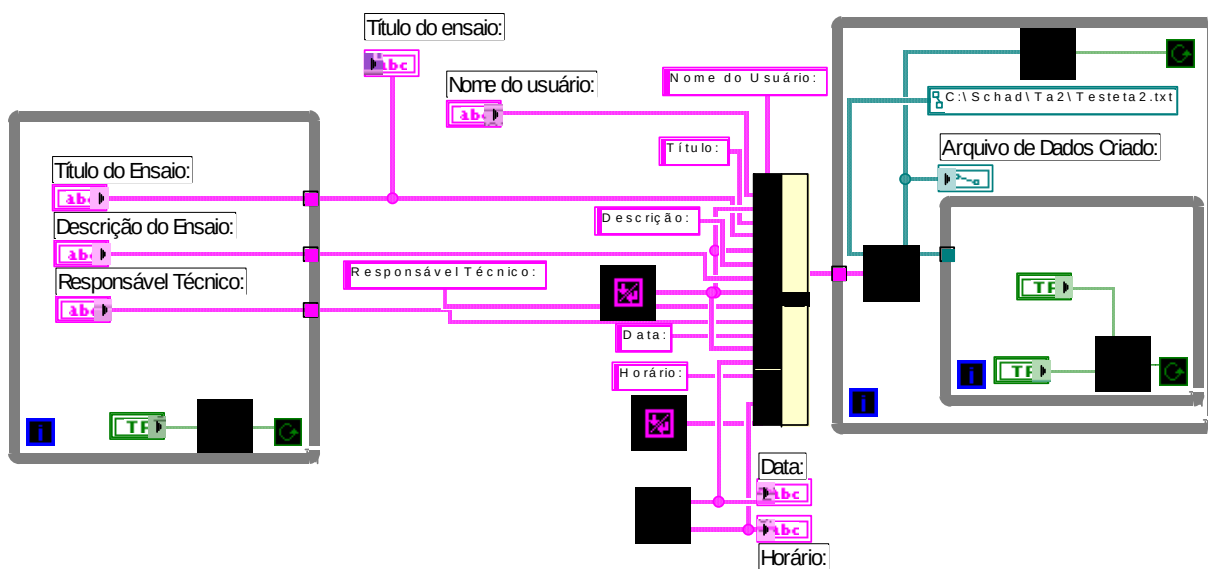


Figura 6.41 - Diagrama de blocos e funções da unidade identificação de ensaio.

INSTITUTO DE AERONÁUTICA E ESPAÇO
ASA-L TÚNEL AERODINÂMICO
ENSAIO COM SCANIVALVES

Nome do operador: Responsável Técnico:

Título do Ensaio: Descrição do Ensaio:

CONFIRMAR DADOS

Data: Horário:

Arquivo de Dados Criado:

Ângulo (graus): Novo ângulo (graus):

CONFIRMAR NOVO ÂNGULO

Canal number read

PARE

Média das saídas das scanivalves: Pressão relativa: scan backlog:

0,34 -0,59 0,00 0,45 0,05 0,84

Saídas das scanivalves:

0,34 -0,59 -0,00 0,45 0,05 0,82

Figura 6.42 - Unidade de software para aquisição de dados de *scanivalves* P1.

As implementações do subproduto SSV foram apresentadas nas figuras 6.40 a 6.42, e constituem uma combinação dos dados das unidades “info de ensaios” e “scanivalves p1,p2,p3,p4,p5,p6”, com tomadas de pressão das *scanivalves*, as quais foram obtidas com a plataforma PC, com relação aos dados de cada experimento e configuração estabelecidos no programa de ensaio.

Os resultados dos testes aplicados para a unidade identificação de ensaios no subproduto SSV correspondem às saídas armazenadas em arquivos, as quais são mostrados nas figuras 6.43 e 6.44 conforme aquisição para mesmo ângulo de ataque, porém por questão de sigilo, não são indicadas a quais tomadas de pressões estes valores correspondem.

```

Nome do operador:      TAKESHI
Título do ensaio:      A0045
Descrição do ensaio:    Rake Azul pos. 2.5      Scan 06 (1, 2, 3, 6 e 5)
Responsável técnico:    Pressao Barometrica = 709.7
Data:                  29/04/99
Horário:               21:52
Pressão dinâmica:      0,000098
0,0000000 0,0000000 -0,0003395      -0,0003739      -0,0001820      -0,0003886
      -0,0006667      2,7312608 0,0001065
0,0000000 1,0000000 -0,0003409      -0,0004205      -0,0001804      -0,0003855
      -0,0006651      2,7311294 0,0001061
0,0000000 2,0000000 -0,0003438      -0,0003901      -0,0001806      -0,0003764
      -0,0006632      2,7310836 0,0001061

```

Figura 6.43 - Resultados de testes de ensaio A0045 e armazenagem de dados.

```

Nome do operador:      ANA LUCIA
Título do ensaio:      A0046
Descrição do ensaio:    Rake Arrasto. FLAP=0. X=5. Z=0. SLAT2=0.X=5 .Z=0
Responsável técnico:    Pressao Barometrica = 710.655
Data:                  30/04/99
Horário:               08:04
Pressão dinâmica:      0.000058
0.0000000 0.0000000 -0.0003362      -0.0003752      -0.0001634      -0.0004248
      -0.0006706      2.6096816 0.0000922
0.0000000 1.0000000 -0.0003412      -0.0003723      -0.0001564      -0.0004155
      -0.0006678      2.6097977 0.0000927
0.0000000 2.0000000 -0.0003363      -0.0003762      -0.0001612      -0.0004272
      -0.0006738      2.6099763 0.0000917

```

Figura 6.44 - Resultados de testes de ensaio A0046 e armazenagem de dados.

No ensaio de um modelo NACA0012, foi utilizado o *rake* de arrasto (tubos posicionados em fila montados e fixados em intervalos fixos, interligados por tubos flexíveis a transdutores de pressão) composto de 92 tomadas de pressão, incluindo as medidas de três *pitots* de pressão estática. Para validação dos dados deste ensaio, todas as medidas foram feitas em conjunto usando ambas plataformas HP e PC, ou seja, com os sistemas de softwares existentes da HP e a biblioteca “TA2_SV.llb” instalada no PC. As *scanivalves* e sensores usados para estes ensaios são mostrados na tabela 6.2, os quais são as mesmas para ambas as plataformas.

Tabela 6.2 - *Scanivalves* usadas no ensaio do modelo NACA0012 do TA-2

Material	Modelo	Num. Série
Scanivalve 1	48D3	1357
Sensor	PDCR23D	B11192 (± 1 PSI)
Scanivalve 2	48D3	1360
Sensor	PDCR23D	B9539
Scanivalve 3	48D3	1358
Sensor	PDCR23D	B11193
Scanivalve 4	48D3	1362
Sensor	PDCR23D	B9533 (± 1 PSI)
Scanivalve 5	48D3	1359
Sensor	PDCR23D	B11191 (± 1 PSI)
Scanivalve 6	48D3	1361
Sensor	PDCR23D	B9536 (± 1 PSI)

Este ensaio com ambas as plataformas serviu de caso de teste para validação do subproduto SSV. Para execução deste caso de teste foram definidos os dispositivos eletrônicos do ambiente de ensaio, para executar as funcionalidades e operações de leitura de sinais e atuações de ângulo de ataque, pressão dos 48 canais das *scanivalves* 1,2,3,4,5,6, e também pressão dinâmica do *pitot*, cumprindo com os requisitos para aquisição dos seguintes dados:

- Dados dos três *pitots* de tomadas estáticas.
- Dados instalados no *rake* de arrasto.
- Dados de dentro do caixão do modelo aerodinâmico, que corresponde ao bordo de ataque, intradorso e o extradorso do modelo aerodinâmico, e os dados da calha.
- Dados de pressão do *spoiler* e *slat*.
- Dados das pressões estática e total do *pitot*.
- Dados das pressões estáticas da parede à frente, à direita e à esquerda do perfil.

- Dados da estática externa.
- Dados da varredura da pressão dinâmica local dentro do espaço entre as paredes, com relação a saída dos tubos do *rake* de arrasto.

Os componentes ou unidades de software do SSV são testados individualmente e integrados para criar um sistema de software parcial. Os resultados obtidos são comparados com aqueles obtidos com a plataforma HP. Neste caso de teste, pelo fato da aquisição ser feita quase que instantaneamente com ambas as plataformas, os dados obtidos também foram avaliados simultaneamente, validando-os por comparação visual feita pelos especialistas durante o ensaio. Pode-se observar neste ensaio que o tempo de aquisição com SSV em relação aos obtidos com HP, era menor, ou seja, cerca de dois segundos mais rápido.

Este fato ocorreu devido aos equipamentos HP interligados, necessitarem de um *delay* para o sincronismo entre eles, para prosseguir com a aquisição de dados. Este fato comprova um melhor desempenho na aquisição de dados com o SSV, instalado na plataforma PC e dispositivos eletrônicos interligados, definidos e revisados pelos papéis competentes.

Como requisito de software e de interface para este subproduto, está aquele referente à modelagem dos dados para ser possível executar o tratamento destes dados, conforme solicitado pelos especialistas em ensaios de túnel de vento e pelos clientes, dadas as necessidades de urgência dos resultados e para manter a operacionalidade do TA-2. Para cumprir este requisito se desenvolveu o subproduto de software com aplicativo *EXCEL* para ordenação de dados da *scanivalve* (SVE). O desenvolvimento deste seria uma solicitação do cliente e do especialista em ensaios em túnel, para que o processamento dos dados pudesse ser realizado de forma correta.

6.4.3.3 Implementação e testes do subproduto de software com aplicativo *EXCEL* para ordenação de dados da *scanivalve* – SVE

A ferramenta *Microsoft Excel* disponibiliza os recursos de programação de edição e gravação de macro, como acesso direto à linguagem de programação *Visual Basic*. Foi solicitada pelo cliente a ordenação dos dados das *scanivalves* e dos *rakes*, usados nos ensaios do modelo aerodinâmico da aeronave 170, com o NACA 0012 em 1999, correspondentes ao contorno do perfil aerodinâmico, para uso destes dados posteriormente, na validação de um projeto experimental.

Para este trabalho foi construído o subproduto SVE utilizando os recursos disponibilizados pela ferramenta *EXCEL* para criar uma macro, com uso do mecanismo de gravação de macros, e quando necessário utilizava-se o recursos de edição, para executar as modificações necessárias na estrutura dessa macro, e adequar às reais intenções de ordenação dos dados.

O uso de macros para ordenação dos dados exigiu do desenvolvedor mais conhecimento do que normalmente ele dispunha, devido ao fato deste ter que conhecer todo o subsistema do modelo de ensaio, para estabelecer as respectivas ordenações. Trata-se de saber identificar onde estavam as tomadas de pressão estáticas da parede, os zeros iniciais das *scanivalves*, as pressões dinâmicas médias, os sinais dos *flaps*, da calha, do corpo central, dos *spoilers*, do *slat* e dos *rakes* envolvidos nos ensaios. Algumas partes deste subproduto de software podem ser vistas no Apêndice D, deste trabalho.

Como pode ser observado o SVE obtém os dados das planilhas a partir dos valores adquiridos e armazenados pelo subproduto SSV, os ordena e armazena em arquivos. Estes dados ordenados das tomadas de pressão das *scanivalves* de todo o perfil aerodinâmico ficam disponíveis para o processamento e cálculos de redução de dados, conforme solicitação do cliente.

Com os subprodutos SSV e SVE pode-se comprovar o reuso dos dados de pressões das *scanivalves*, e o cumprimento dos requisitos dos clientes. Os testes aplicados em tempo de execução dos ensaios serviram de evidências, para comprovação do funcionamento correto destes subprodutos. Os dados obtidos foram ajustados com o SVE, para possibilitar o processamento destes pelos clientes juntamente com os especialistas em ensaios em túnel.

6.4.3.4 Implementação e testes do subproduto de software de aquisição de dados e calibração da balança – SBA (C3).

A implementação do SBA também foi feita usando o *LabView*, fazendo reuso de funções desta ferramenta. Também se fez reuso de funções e interfaces com usuário com algumas adequações, processos de engenharia comuns, dispositivos eletrônicos, e alguns hardwares aplicados no SSV com exceção da placa de aquisição instalada no PC, que foi substituída pela placa “AT-MIO-16E-10”.

No desenvolvimento deste subproduto os requisitos em destaque, neste trabalho, são aqueles correspondentes à calibração da balança externa, na qual o cálculo de ganho é uma funcionalidade importante. Assim como a coleta dos demais dados de zero inicial, forças e momentos aerodinâmicos, entre outros, estes cálculos são implementados em diagramas de blocos e painéis, conforme apresentado nas figuras 6.45 a 6.50.

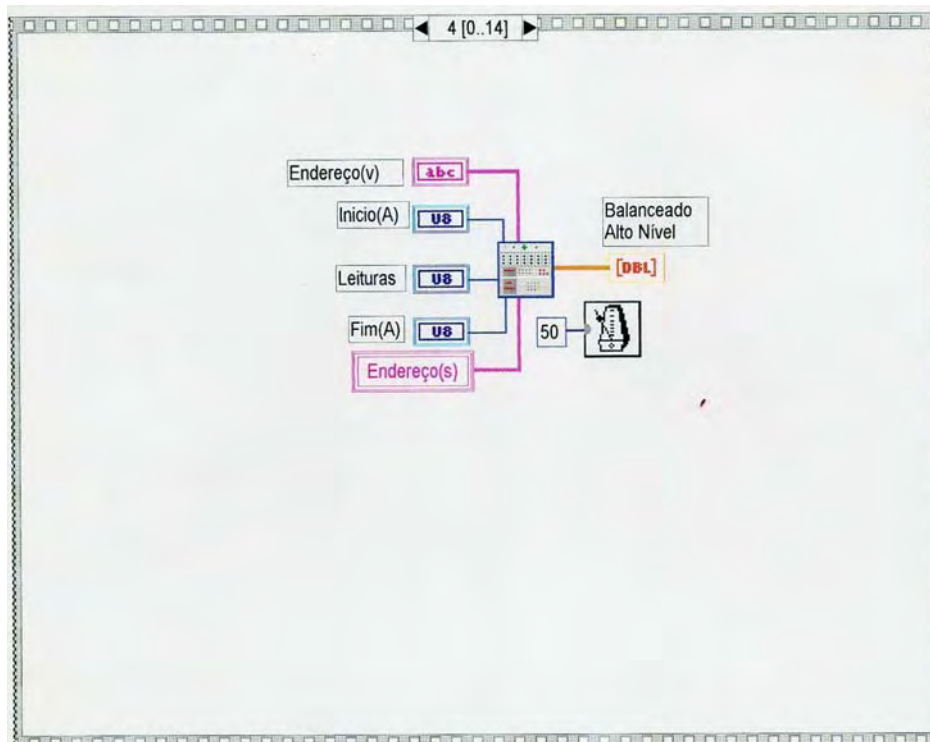


Figura 6.45 - Implementação do SBA alto nível balanceado.

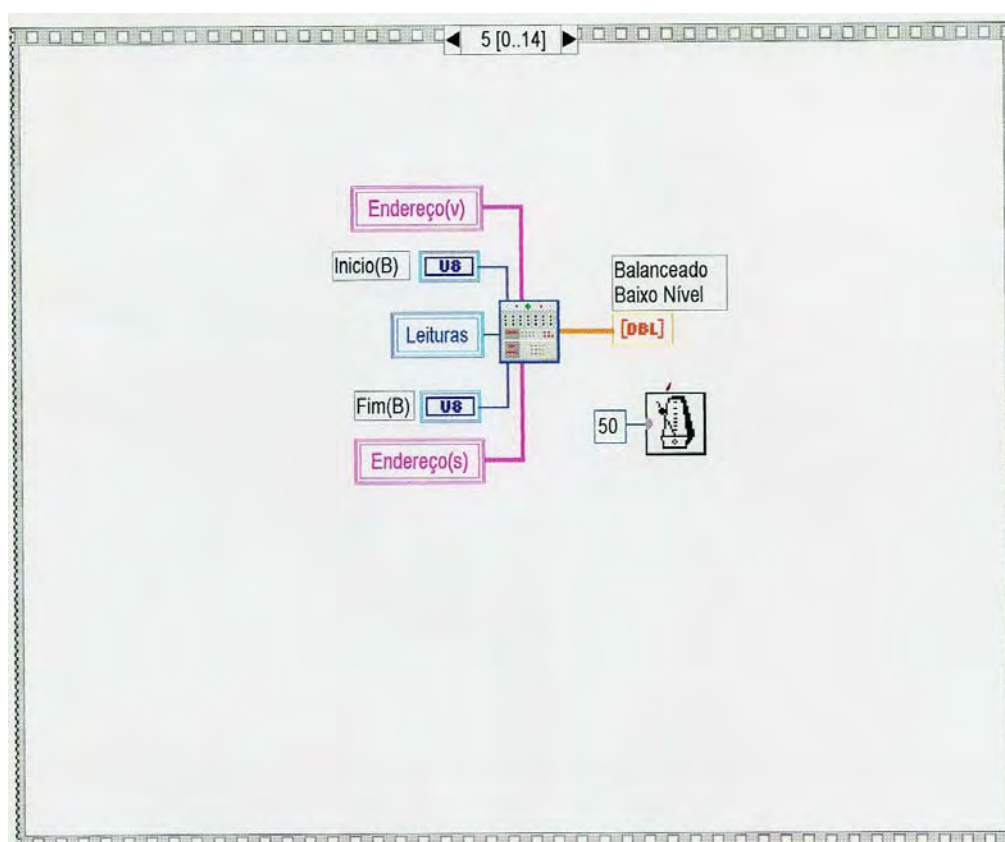


Figura 6.46 - Implementação do SBA baixo nível balanceado.

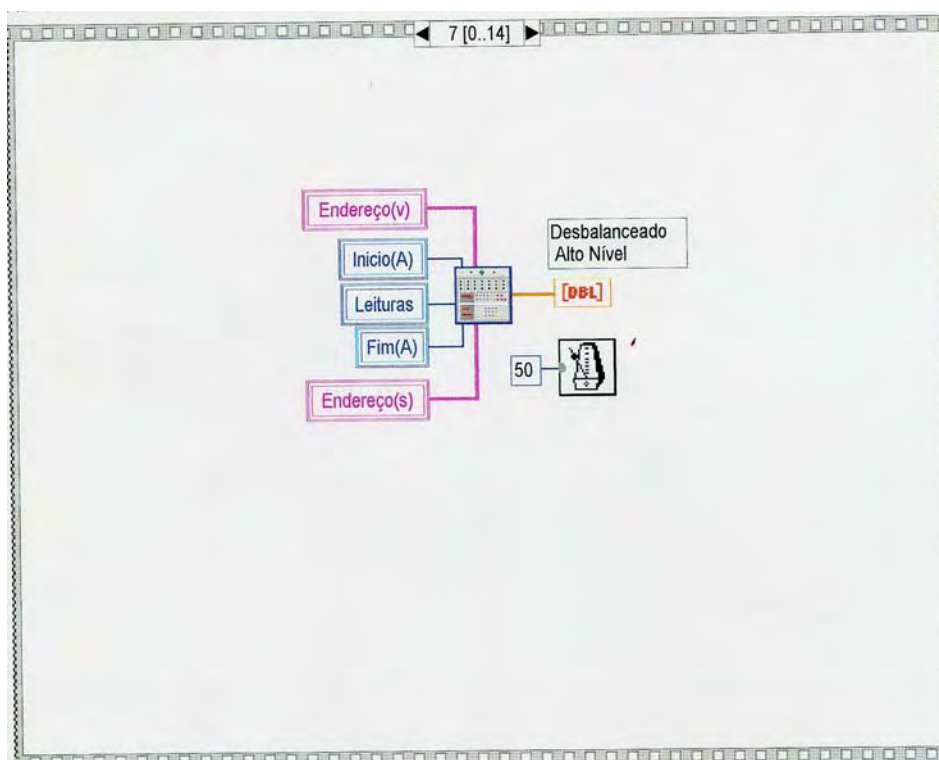


Figura 6.47 - Implementação do SBA alto nível desbalanceado.

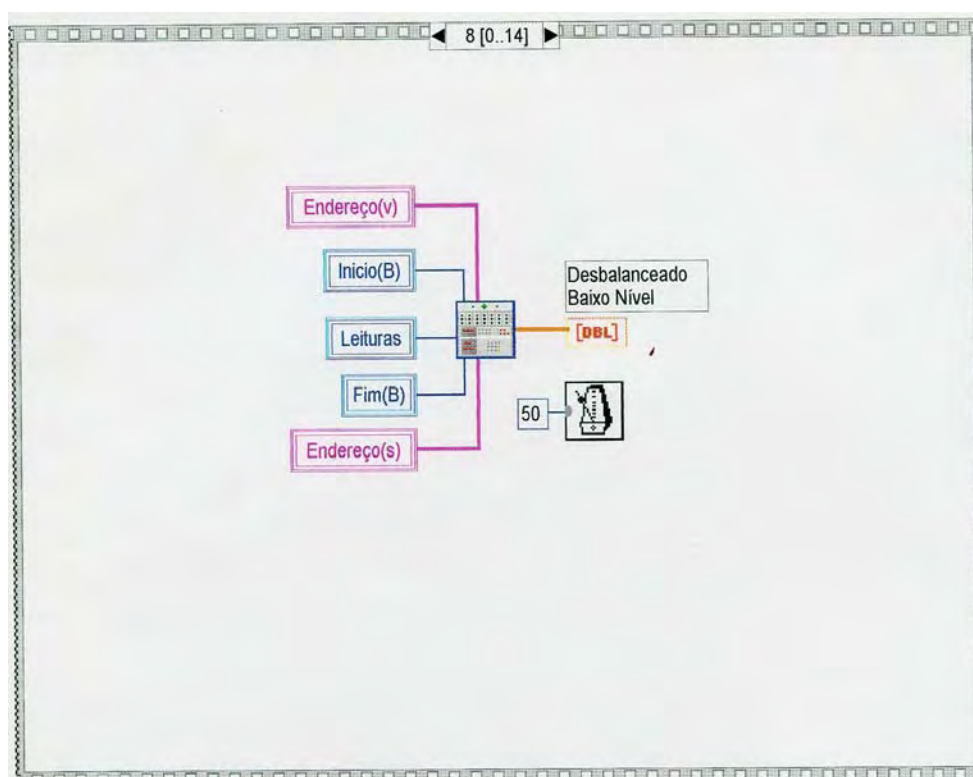


Figura 6.48 - Implementação do SBA baixo nível desbalanceado.

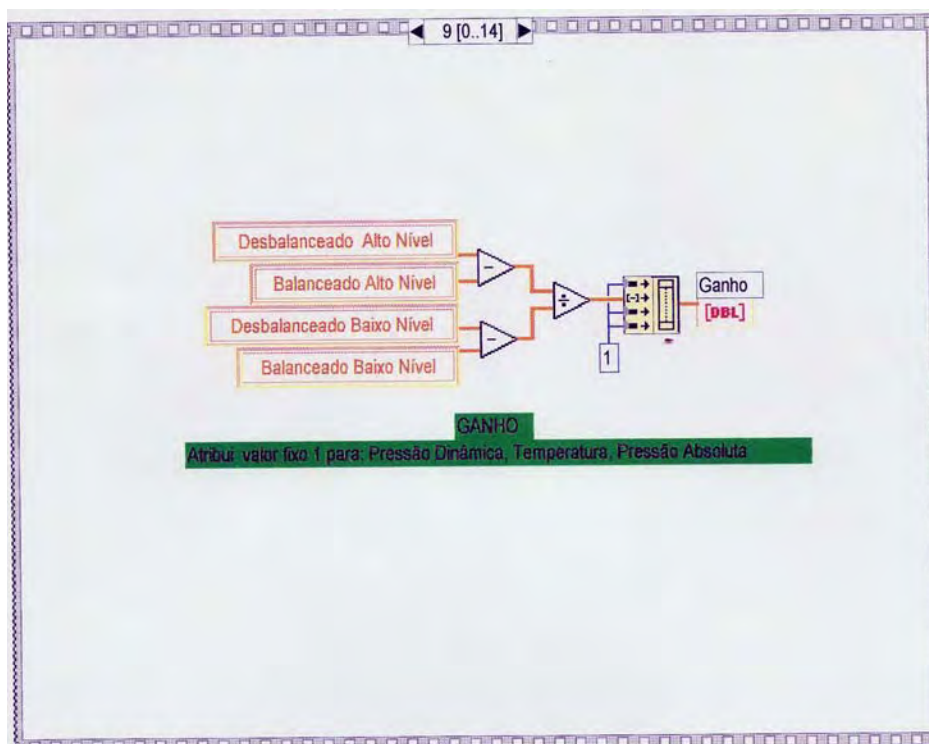


Figura 6.49 - Implementação do SBA aquisição, ganho e zero inicial.

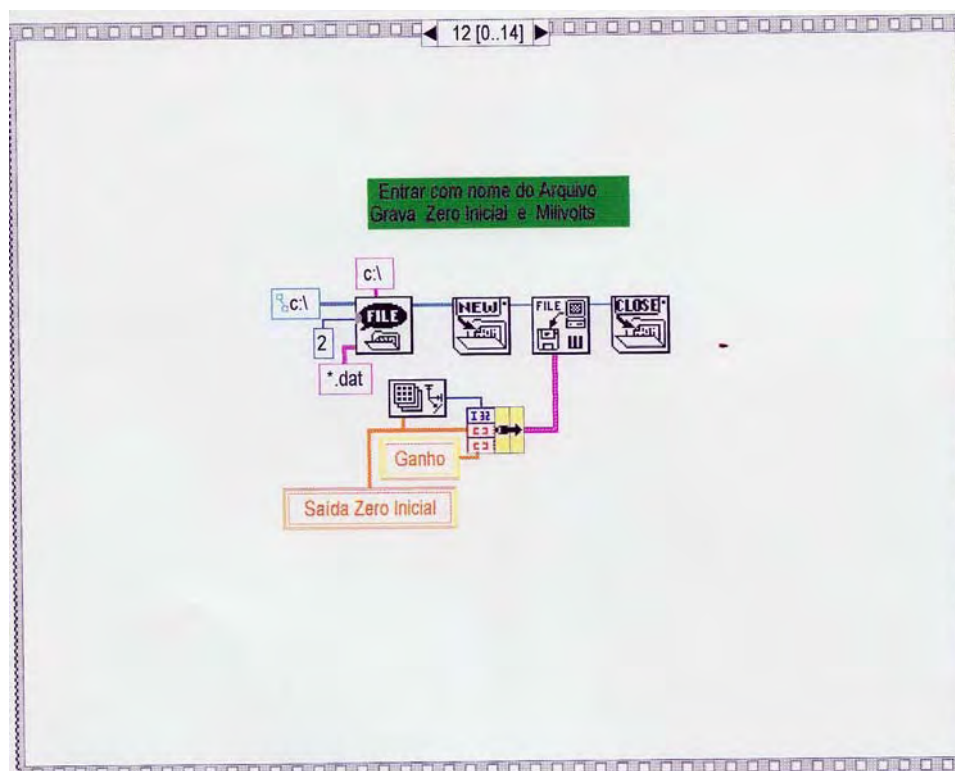


Figura 6.50 - Implementação do SBA de unidade para gravação de dados.

O objetivo da calibração é relacionar os sinais eletrônicos fornecidos pelos sensores com os fenômenos físicos registrados computacionalmente como dados, por um software de aquisição de dados. A calibração da balança externa do TA-2, por exemplo, tem por objetivo relacionar os sinais fornecidos pelas células de carga às forças aplicadas através dos carregamentos, ou do próprio vento durante o pré-ensaio com o modelo aerodinâmico.

As atuações das forças surgem de deformações elásticas que provocam desalinhamento do carregamento e, com isso, há indução de forças e momentos em outras componentes, a partir dos quais primeiramente são obtidas as constantes de calibração no laboratório de força, e em seguida é feita a obtenção dos dados com os carregamentos e cargas em alfa ou beta, para posteriormente aplicar um modelo de iteração (NOGUEIRA, 1980), para obter a matriz de calibração.

A calibração da balança do TA-2 é feita com aquisição de dados brutos conhecidos na comunidade de especialistas do TA-2 como dados de baixo nível. Os ensaios aerodinâmicos com os modelos instalados no mastro sob a balança, são feitos com aquisições de dados processados, conhecidos como dados de alto nível, ou seja, os sinais são medidos com ganho e a aquisição de dados via software registra o efeito do filtro, de condicionadores de sinais entre outros dispositivos interligados.

O efeito desse sistema de aquisição em baixo e alto nível foi avaliado mediante comparações entre sinais obtidos com software da HP e com o software do PC contendo a “vi” de cálculo de ganho, para que através dos resultados destas fosse validado o novo software de aquisição de dados. O cálculo de ganho é um dos itens fundamentais no processo de correção do sistema de medidas com o software de redução de dados com modelo aerodinâmico. Para a aceitação do novo software de aquisição de dados foi estabelecido um caso de teste de cálculo de ganho.

6.4.3.4.1 Caso de teste de cálculo de ganho

Um dos casos de teste trata do cálculo do ganho com as aquisições de dados de cada uma das componentes aerodinâmicas de forças e momentos, no qual computacionalmente é demonstrada a conformidade do cálculo de ganho. Para este caso de teste são feitas aquisições de dados dos sinais em baixo nível e em alto nível com os dispositivos eletrônicos instalados para as plataformas PC e HP.

O ganho de cada componente é obtido através de cálculos feitos via software, durante a aquisição dos dados. A comparação dos dados obtidos das células de carga da balança, entre as aquisições com as plataformas PC e HP, visam validar os resultados deste caso de teste de cálculo de ganho, com o software de aquisição de dados da terceira iteração da fase de construção do processo unificado.

Para obtenção dos resultados fizeram-se uso dos dispositivos estabelecidos para as plataformas PC e HP; no entanto, um dos dispositivos, por disponibilizar mais facilidades de manuseio de condicionamento de sinais, foi comum para ambas as plataformas: trata-se do condicionador de sinal da *Honeywell Accudata*. Dois destes dispositivos foram instalados um interligado à plataforma PC e outro, à HP. O operador manipula ambos simultaneamente e a aquisição de dados é feita através do software existente denominado “Cálculo do Ganho” ou “NOVOGANHO” para HP e do subproduto SBA para PC, com objetivo de que forneçam resultados compatíveis.

Para este caso de teste, a placa SCXI 1121 da *National Instruments* estava instalada, porém com ganho zero, pois o objetivo do caso de teste é a aceitação do software com vistas ao reuso da “vi” de ganho. O cálculo efetuado pelo novo software de aquisição e o da plataforma HP fez uso de uma das facilidades com o dispositivo da *Honeywell*, no qual o painel de comando dos ganhos pode ser facilmente manuseado. A placa SCXI desempenharia a mesma função, porém dificultava o processo devido ao estabelecimento do ganho se dar através de “*jumpeamento*”, na qual a placa deveria

ser aberta e manuseada a cada alteração de ganho. A figura 6.51 mostra o diagrama das medidas de alto nível e baixo nível.

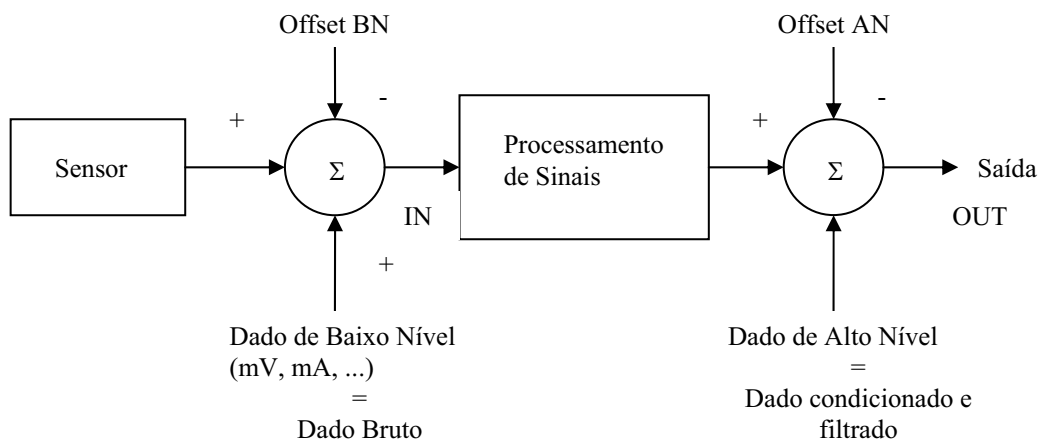


Figura 6.51 - Diagrama das medidas de alto e baixo nível.

O ganho é calculado com os dados de *offset* de alto e de baixo nível de cada canal, conforme a equação 6.1 a seguir. Os resultados desta aquisição de dados são apresentados nas tabelas de 6.3 a 6.9 a seguir. Nestes resultados pode ser observados que os cálculo de ganho tanto com a plataforma HP quanto com a plataforma PC são similares. Este fato comprova a qualidade dos dados, os quais também validam o subproduto SBA e a possibilidade de reuso das funções implementadas.

$$Ganho = \frac{Saída}{Entrada} = \frac{altonivel - offsetAN}{baixonivel - offsetBN} \quad (6.1)$$

Tabela 6.3 - Dados de *offset* de baixo nível com HP e PC

Software	R1 (V)	R2 (V)	R3 (V)	R4 (V)	R5 (V)	R6 (V)
HP	-0,00003	0,00031	-0,00006	-0,00007	-0,00006	0,00025
PC	-0,00002	0,00030	-0,00009	-0,00006	-0,00005	0,00026

Tabela 6.4 - Dados de calibração baixo nível com HP

R1 (V)	R2 (V)	R3 (V)	R4 (V)	R5 (V)	R6 (V)
-0,01023	-0,01027	-0,01019	-0,01021	-0,01021	-0,01023
-0,00510	-0,00514	-0,00509	-0,00510	-0,00509	-0,00502
0,00513	0,00511	0,00510	0,00513	0,00513	0,00512
0,01026	0,01025	0,01020	0,01025	0,01025	0,01022

Tabela 6.5 - Dados de calibração baixo nível com PC

R1 (V)	R2 (V)	R3 (V)	R4 (V)	R5 (V)	R6 (V)
-0,01025	-0,01007	-0,01023	-0,01022	-0,01023	-0,00997
-0,00513	-0,00494	-0,00519	-0,00512	-0,00513	-0,00480
0,00511	0,00534	0,00502	0,00509	0,00508	0,00526
0,01024	0,01048	0,01013	0,01024	0,01023	0,01060

Tabela 6.6 - Dados de *offset* de alto nível com HP e PC

Software	R1 (V)	R2 (V)	R3 (V)	R4 (V)	R5 (V)	R6 (V)
HP	-0,02587	0,07969	-0,05399	-0,01950	-0,02520	0,03153
PC	-0,02572	0,07356	-0,05866	-0,01713	-0,02432	0,03144

Tabela 6.7 - Dados de calibração alto nível com HP

R1 (V)	R2 (V)	R3 (V)	R4 (V)	R5 (V)	R6 (V)
-4,09413	-3,39808	-8,34261	-7,65091	-7,67788	-4,05577
-2,06692	-1,69589	-4,26323	-3,81540	-3,81690	-2,01908
1,98204	1,70490	3,89486	3,85128	3,89664	2,04578
4,00941	3,40725	7,97457	7,69000	7,75820	4,08426

Tabela 6.8 - Dados de calibração alto nível com PC

R1 (V)	R2 (V)	R3 (V)	R4 (V)	R5 (V)	R6 (V)
-4,09464	-3,39889	-8,33982	-7,65147	-7,67572	-4,05787
-2,06731	-1,69654	-4,26020	-3,81597	-3,81437	-2,02029
1,98189	1,70452	3,89827	3,85122	3,84898	2,04653
4,00933	3,40690	7,97863	7,69027	7,76070	4,08575

Tabela 6.9 - Cálculo do ganho HP e PC

Software	R1 (V)	R2 (V)	R3 (V)	R4 (V)	R5 (V)	R6 (V)
Nominal	395,00	331,00	800,00	750,00	755,00	400,00
HP	395,67	331,96	800,71	749,67	754,59	399,80
PC	395,47	330,93	800,32	750,40	754,93	400,08
# HP	+0,67	+0,96	+0,71	-0,33	-0,41	-0,20
# PC	+0,47	-0,07	+0,32	+0,40	-0,07	+0,08

Os dados apresentados nas tabelas anteriores são aqueles adquiridos com as plataformas HP e PC, usando o subproduto SBA na aplicação do caso de teste de cálculo de ganho do PC, o qual possui funções em uma “vi” com características de usabilidade, com a possibilidade de reuso na iteração C4, pelo subproduto software de aquisição de dados e controle do túnel de vento (SCB).

Usando este mesmo subproduto, para a aplicação de outro caso de teste na calibração dos *pitots*, com objetivo de validação dos dados dos ensaios aerodinâmicos, surgiu a urgência de desenvolvimento de outro subproduto de software, o STD denominado software de transferência de dados da HP para o PC, implementado de forma rápida devido aos equipamentos HP virem apresentando sinais de falhas.

Com a implementação do STD, todos os dados obtidos com a plataforma HP foram recuperados, transferidos e armazenados no PC, possibilitando a validação do

SBA, por apresentar uma similaridade de valores obtidos com aplicação dos cálculos de calibração, usando os dados de ambas as plataformas em um caso de teste. Os resultados com aplicação do caso de teste de calibração de *pitots* foram comparados, comprovando assim que o subproduto SBA apresenta dados corretos.

6.4.3.5 Implementação e testes do subproduto de software de transferência de dados da HP para o PC - STD.

O subproduto de software de transferência de dados da HP para o PC foi implementado em linguagem de programação C, em regime de urgência. A implementação deste subproduto colaborou na recuperação de todos os dados adquiridos pela plataforma HP, e armazenagem em arquivos na plataforma PC.

Estes dados foram posteriormente usados para validação dos resultados de calibração usando os subprodutos SBA e SCB deste trabalho, no cálculo de distribuição dos dados usando o aplicativo ARENA, e no cálculo dos coeficientes aerodinâmicos usando o subproduto de software de redução de dados do túnel de vento (SRD).

Nesta implementação se fez uso de técnicas estruturadas e da ferramenta C++ Builder 4, e também de uma parceria com especialistas da EMBRAER para o desenvolvimento deste subproduto. Parte desta implementação pode ser observada no Apêndice D, deste trabalho. A aplicação do caso de teste de calibração dos *pitots* usando o subproduto SBA, assim como os demais casos de testes aplicados, demonstra a validação da correta transferência dos dados usando o STD.

O caso de teste do processo de calibração do *pitot* é apresentado de forma completa, ou seja, são apresentados os dados obtidos e armazenados pelo SBA e o processamento destes para obtenção dos coeficientes de calibração, com comparações dos resultados das aplicações dos métodos de regressões usando os dados adquiridos

por ambas as plataformas, com auxílio do software de transferência de dados da HP para o PC (STD).

Exclusivamente para o processo de calibração do *pitot*, considera-se que a velocidade do vento é medida por um tubo de *pitot* instalado na Seção de Ensaios e outro *pitot* próximo ao modelo aerodinâmico, com os quais se estima a velocidade aerodinâmica através da medida da pressão dinâmica no local do ensaio. Alguns erros podem ser cometidos ao se obter essa medida, que podem ocorrer na indicação do instrumento quando este não está calibrado ou instalado corretamente, ou baseados no posicionamento das tomadas de pressão, na compressibilidade do fluido, na diferença entre a densidade local e a utilizada para computar a velocidade aerodinâmica.

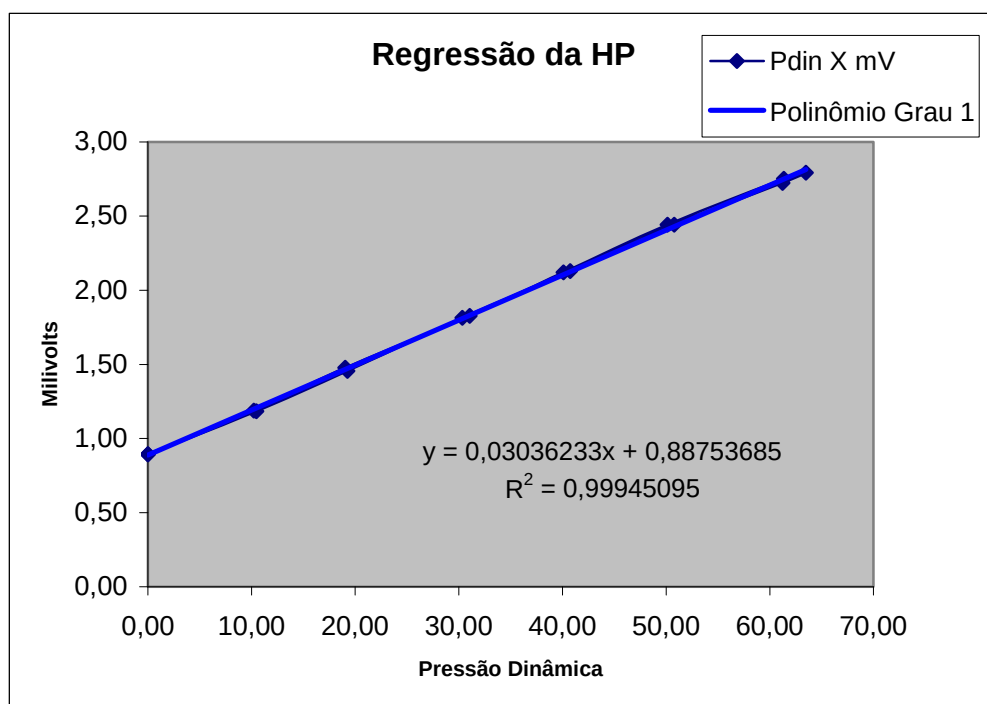
Para solucionar alguns desses erros, primeiramente é calibrado o tubo de *pitot* instalado no local do ensaio, em seguida este *pitot* é alinhado em relação ao vento ou adotam-se fatores de correção do vento em relação ao modelo aerodinâmico. Para comprovação dos requisitos de calibração do *pitot*, além dos dados de aquisição apresentados nas tabelas pelos valores em milivolts, se executou o processamento destes para apresentação dos resultados do processo de calibração do *pitot*, para obtenção dos coeficientes da pressão dinâmica.

6.4.3.5.1 Caso de teste de calibração de *pitot*

O caso de teste de calibração do *pitot* utiliza funções de regressão, para processamento dos dados da pressão dinâmica em *milivolts* adquiridos com o SBA, e que através do STD são transferidos da plataforma HP para o PC. Estes cálculos de constantes ou coeficientes da calibração são uma parte do software de redução de dados do túnel de vento (SRD). Os resultados do caso de teste de calibração do *pitot* com a plataforma HP são apresentados na tabela 6.10 e figura 6.52.

Tabela 6.10 - Resultado da aplicação do caso de teste de calibração do *pitot* com HP.

Pressão Dinâmica	Milivolts (mV)	Resultado do Teste (mV)
0,00	0,8953071	0,8875368
10,23	1,1892414	1,1981435
19,05	1,4774741	1,4659393
31,06	1,8259247	1,8305908
40,10	2,1208484	2,1050663
50,11	2,4404654	2,4089933
61,35	2,7529688	2,7502659
63,50	2,7937263	2,8155449
61,24	2,7244823	2,7469260
50,78	2,4420671	2,4293360
40,75	2,1284617	2,1248018
30,35	1,8138217	1,8090336
19,25	1,4554307	1,4720117
10,47	1,1842961	1,2054305
0,00	0,8926424	0,8875368

Figura 6.52 - Resultado da calibração do *pitot* com HP.

A aplicação deste caso de teste é para validar os dados transferidos da plataforma HP para o PC. Neste teste foram usados os dados do processo de calibração do *pitot*, de um tubo de *pitot* instalado próximo ao modelo aerodinâmico no TA-2, e as

amostras de um dos canais do *scanner* interligado ao multímetro, para medir a pressão dinâmica de outro *pitot* instalado na seção de ensaio do TA-2, e calcular as constantes de calibração. Os dados transferidos da HP para o PC foram inspecionados e considerados dados válidos ao serem comparados com os armazenados pela plataforma HP.

O software STD transfere e armazena os dados gerando o arquivo em formato de texto (.txt ou .dat) e após o processamento destes é obtido o gráfico, apresentando as constantes de calibração com correlação de 0,999, para um desvio padrão menor que um (0,67%). Estas constantes de calibração são necessárias na etapa de redução dos dados de ensaios aerodinâmicos. A aplicação do caso de teste com o processo de calibração do *pitot*, com a plataforma PC utiliza a placa SCXI 1121, a qual via SBA adquire amostras por um canal do sensor de pressão dinâmica, conforme apresentado na tabela 6.11 e figura 6.53.

Tabela 6.11 - Resultado da aplicação do caso de teste de calibração do *pitot* com PC.

Pressão Dinâmica	Milivolts (mV)	Resultado do Teste
0,00	0,8953071	0,9015013
10,23	1,2189205	1,2204342
19,05	1,5167078	1,4954085
31,06	1,8685558	1,8698351
40,10	2,1682261	2,1516682
50,11	2,4787895	2,4637423
61,35	2,8215980	2,8141632
63,50	2,8687457	2,8811921
61,24	2,7876170	2,8107338
50,78	2,4809245	2,4846304
40,75	2,1750535	2,1719328
30,35	1,8571191	1,8477000
19,25	1,4961125	1,5016438
10,47	1,2148308	1,2279165
0,00	0,8954955	0,9015013

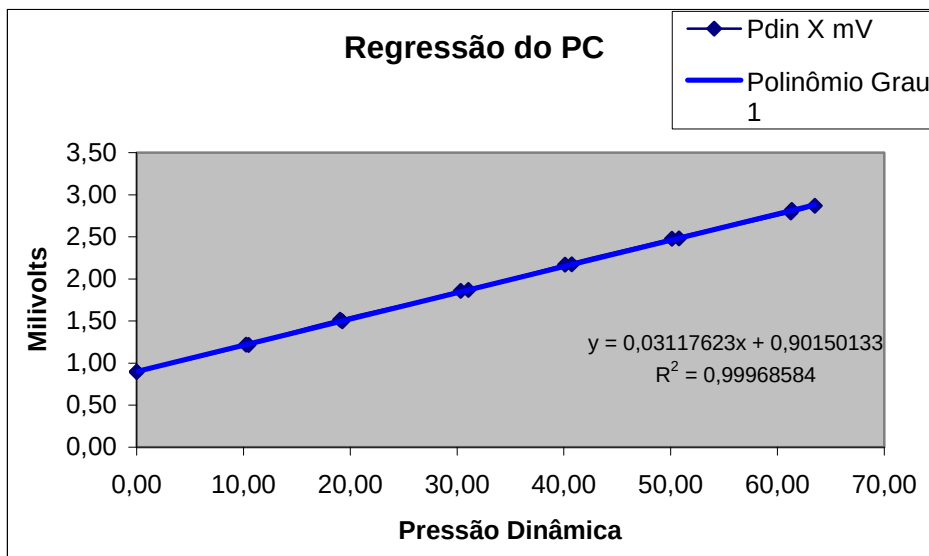


Figura 6.53 - Resultado da calibração do *pitot* com PC.

Segundo as especificações do fabricante dos *pitots*, os dados de calibração obtidos através de uma regressão linear devem apresentar um coeficiente de correlação com valor próximo a um, caso contrário este instrumento de medição não pode ser usado para o sistema de medidas. Com esta prática se comparou o resultado da calibração do *pitot* obtido com a plataforma HP com o obtido com a plataforma PC, apresentando a correlação entre estes resultados através da figura 6.54.

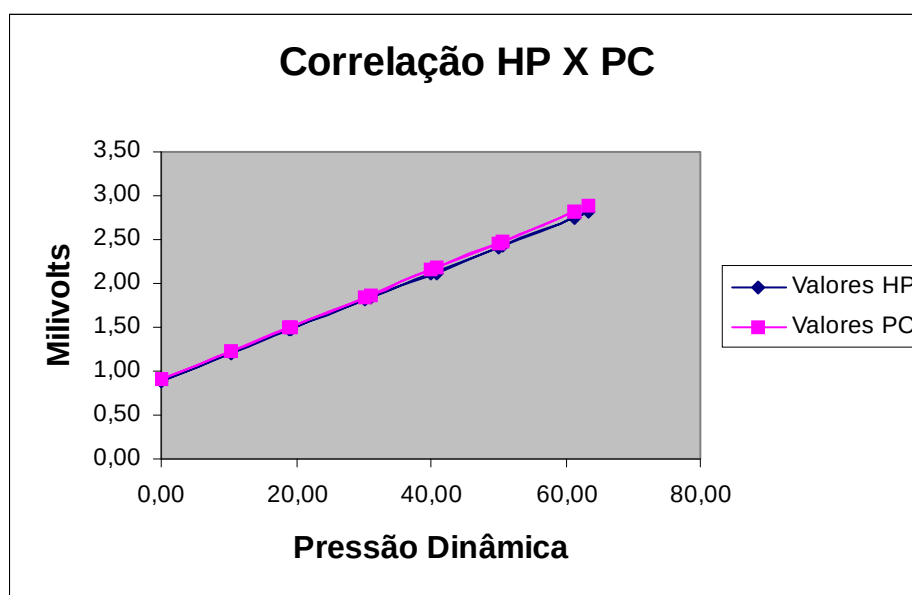


Figura 6.54 - Correlação entre a calibração do *pitot* com plataforma HP X PC.

Este caso de teste serviu para validar a parte do subproduto SBA ou unidade de software usada para calibração do pitot, buscando a compatibilidade de resultados com aqueles obtidos com o software da HP. Em um ensaio em túnel atualmente o desvio padrão estabelecido está numa ordem menor, cerca de 0,5%. Chegou-se ao cumprimento deste requisito com a melhoria do processo de calibração, e demais processos de engenharia, para obtenção dos coeficientes ou constantes de calibração de pressão dinâmica.

6.4.3.6 Implementação e testes do subproduto de software de aquisição de dados e controle do túnel de vento – SCB (C4)

O subproduto de software SCB foi desenvolvido na quarta iteração da fase de construção, e atualmente está em fase de beta teste, para aceitação final do sistema de software integrado, englobando a operação de todos os subsistemas do sistema de medidas do TA-2. Este software será considerado um produto final, após esta aceitação formal pelos especialistas em ensaios em túnel e seus operadores. Na construção deste subproduto é aplicado o reuso de partes dos subprodutos das construções anteriores, dentre eles a unidade de software para cálculo de ganho, conforme mostrada nas figuras 6.55 e 6.56.

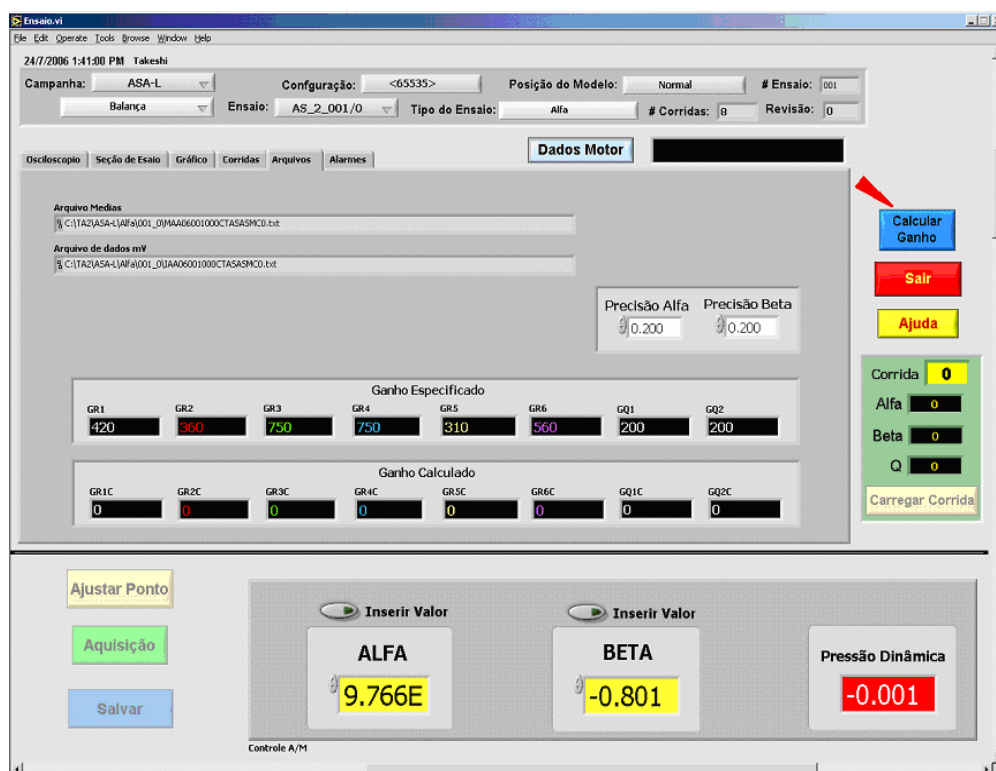


Figura 6.55 - Painel de acionamento do cálculo de ganho.

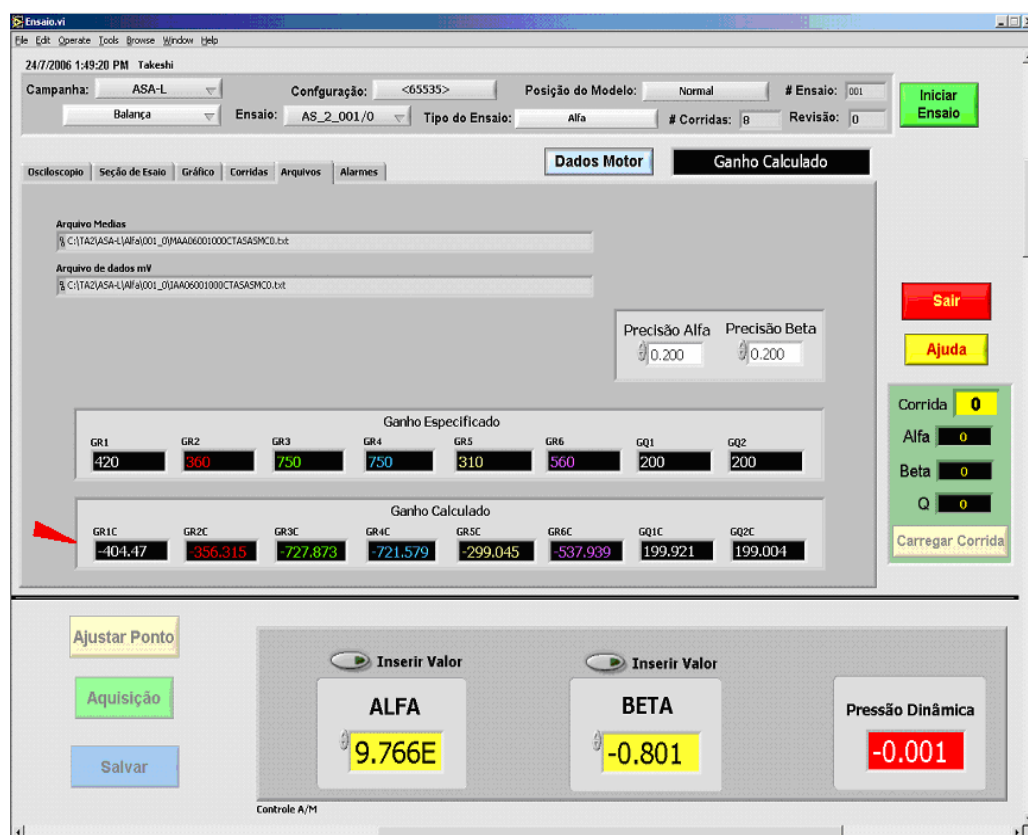


Figura 6.56 - Painel de cálculo de ganho do SCB.

O ganho especificado é aquele nominal do subproduto SBA e estabelecido por hardware e o calculado é implementado por software, para cálculo de ganho de forças e momentos, e de pressão dinâmica. Assim se comprova o reuso no desenvolvimento do subproduto de software de aquisição de dados e controle do túnel de vento (SCB), conforme apresentada na “vi” de cálculo de ganho.

Além do reuso de funções ou “vi” diversas, outras unidades ou componentes de software foram desenvolvidos, adaptados e implementados no SCB. Para cada “vi” são aplicados casos de teste para validação dos dados armazenados por este subproduto.

São implementadas via SCB as funções de monitoramento de dados do subsistema motor e hélice, o qual complementa o sistema de medidas do TA-2. O SCB faz parte da última iteração da fase de construção (C4), complementado quando a hélice foi substituída por outra confeccionada em fibra de vidro. Detalhes de algumas “vi” implementadas são apresentadas nas figuras de 6.57 a 6.61.

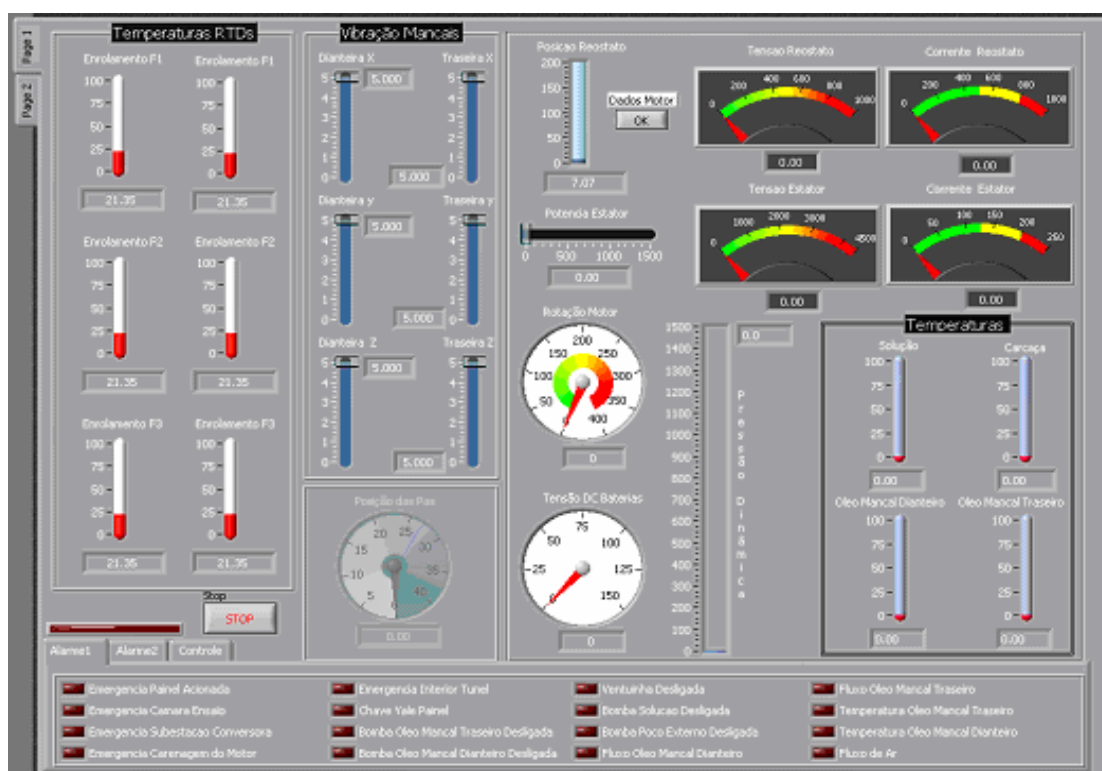


Figura 6.57 - Monitoramento dos dados do subsistema motor e hélice.

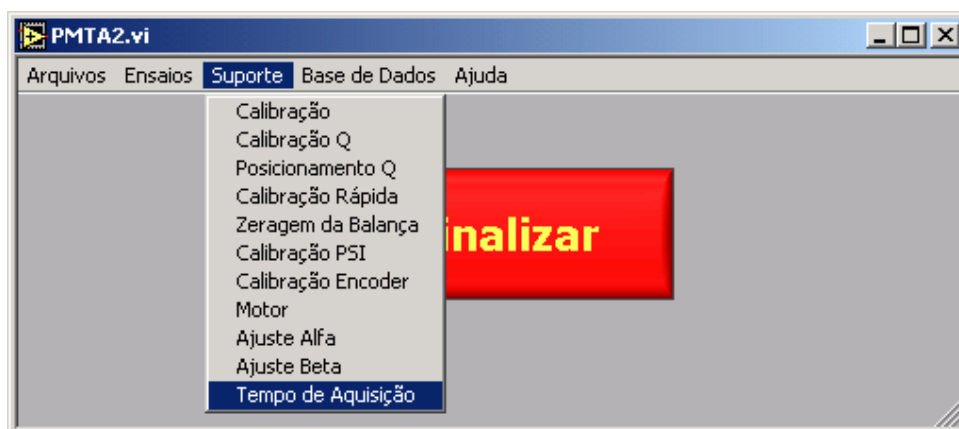


Figura 6.58 - Painel de tipos de aquisições e calibrações.

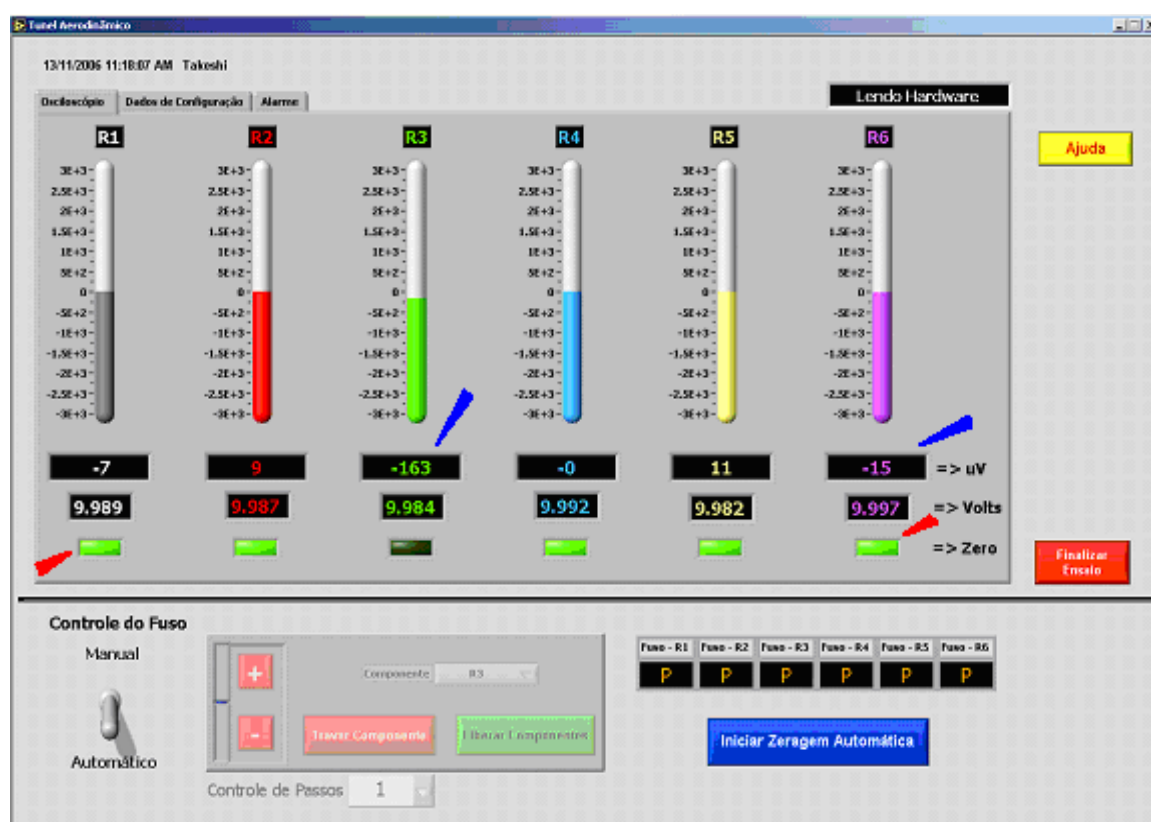


Figura 6.59 - Painel de zeragem automática da balança

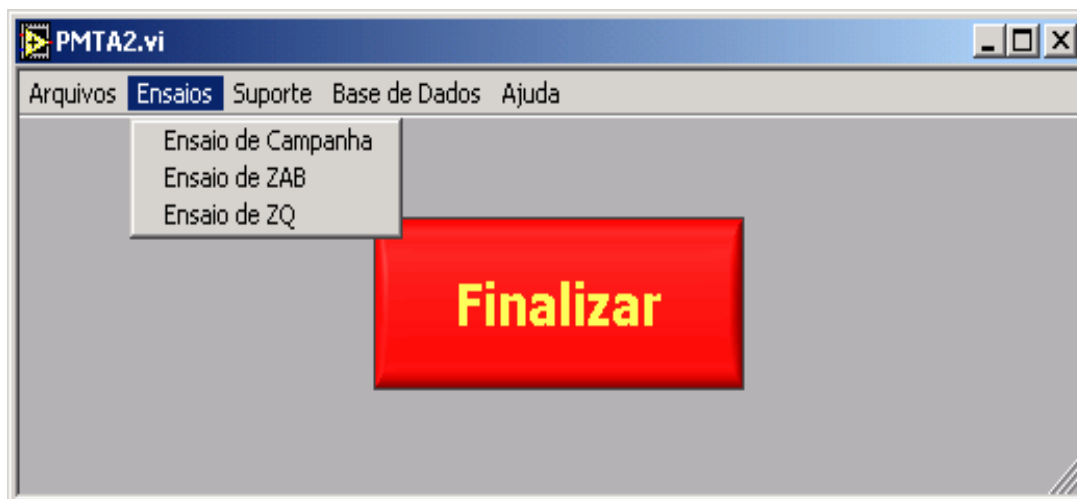


Figura 6.60 - Paine de obtenção de dados de ensaios, ZAB e ZQ

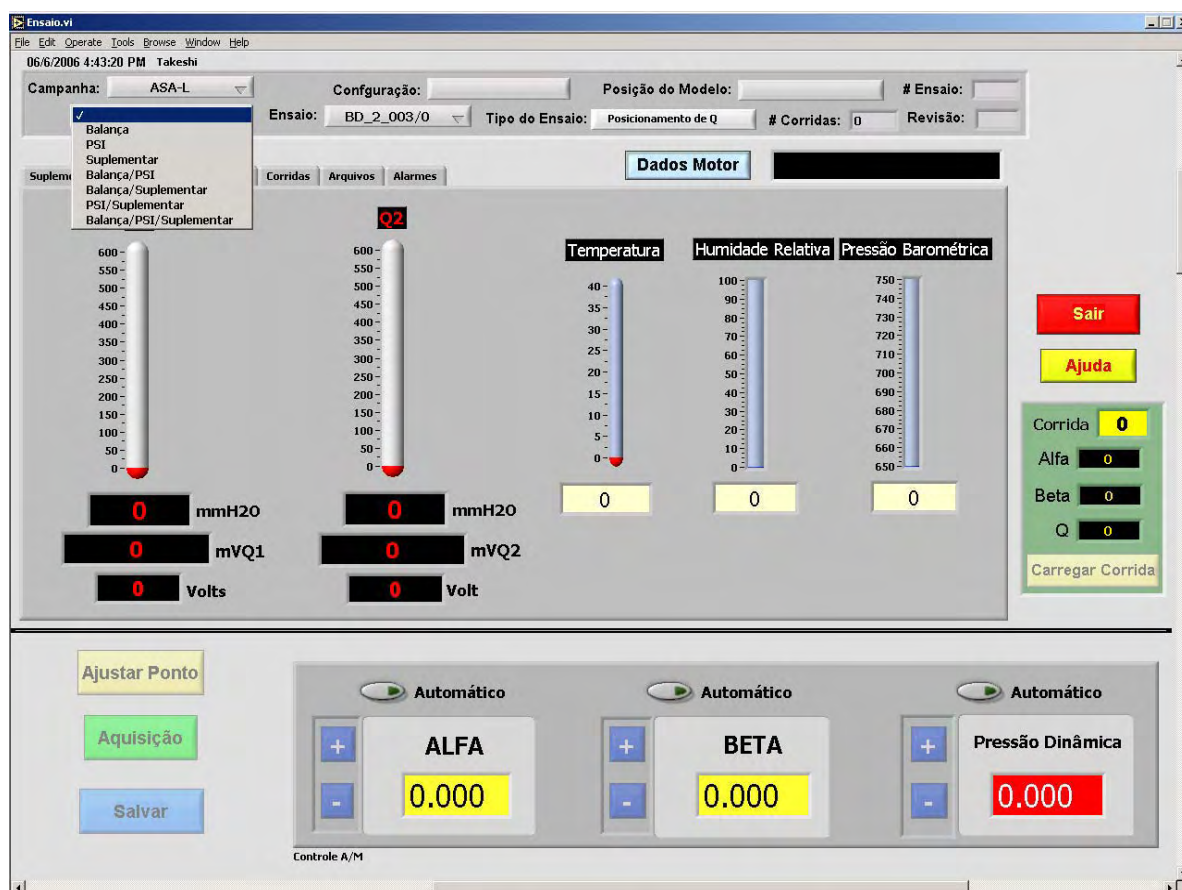


Figura 6.61 - Paine de controle de tipo de ensaio.

Na primeira iteração da fase de transição se validariam as unidades de software referentes ao monitoramento e controle do motor com uma equipe formada pelos especialistas do TA-2 e alguns elementos externos ao túnel, porém internos ao CTA. Nesta primeira iteração os testes de aceitação ainda envolviam algum risco com

respeito ao entendimento das funcionalidades do sistema de software pelos operadores do túnel.

Uma vez dirimidos os riscos se deu início a segunda iteração da fase de transição em que este produto final está em execução dos testes de aceitação. Para aceitação do sistema de software nesta segunda fase de transição, consideram-se principalmente os processos de engenharia abrangendo: modelos aerodinâmicos em escala na seção de testes; monitoramento e controle do motor e hélice; aquisição de cargas da balança externa; movimento do modelo em posições relativas; e medidas das interações aerodinâmicas resultantes.

Os casos de testes aplicados para validação do SCB são praticamente os mesmos aplicados para os demais subprodutos deste trabalho. Aqueles requisitos referentes ao subsistema motor e hélice são comprovados de forma integrada, ou seja, englobando o sistema de medidas como um todo, incluindo o software com aplicativo ARENA para cálculo de distribuição de dados (SAV) e o software de redução de dados (SRD).

6.4.3.7 Implementação e testes do software com aplicativo ARENA para cálculo de distribuição de dados – SAV

A pesquisa na pré-redução dos dados em milivolts das forças e momentos foi feita por simulação usando o aplicativo ARENA da Rockwell, buscando obter as distribuições estatísticas para os dados, para os quais, em princípio, deve ser alcançada a distribuição normal, conforme preconizado por Gasper e Muhlstein (1998).

O processo de aquisição de dados é normalmente executado com a velocidade do vento constante, e por meio de painéis virtuais adequados, implementados via novo software de aquisição de dados e monitorados pelo operador. Para reduzir a quantidade de dados com problemas na fase que precede a redução de dados foi criado este novo processo, no qual o critério de eficiência adotado para os dados é a distribuição

normal, com alterações na taxa de amostragem para observações dos efeitos e influências nos dados, de forma mais apurada.

Primeiramente foram calculadas as distribuições para uma duração de história de tempo de 10 segundos, e posteriormente para 6 segundos. O objetivo principal é observar o quanto o tempo de amostra influencia nos dados adquiridos do sensor de túnel de vento, e observar o nível de ruído ou desvio padrão do sinal do sensor, variando com histórias de tempos diferentes, observando as variações de distribuições em cada força e momento aerodinâmico. As figuras 6.62 e 6.63 apresentam as distribuições das forças e momentos de R1 (força de arrasto) obtidas com o aplicativo ARENA. Os demais resultados de distribuição para histórias de tempo de 10 e 6 segundos de R2 a R6 constam do Apêndice D deste trabalho.

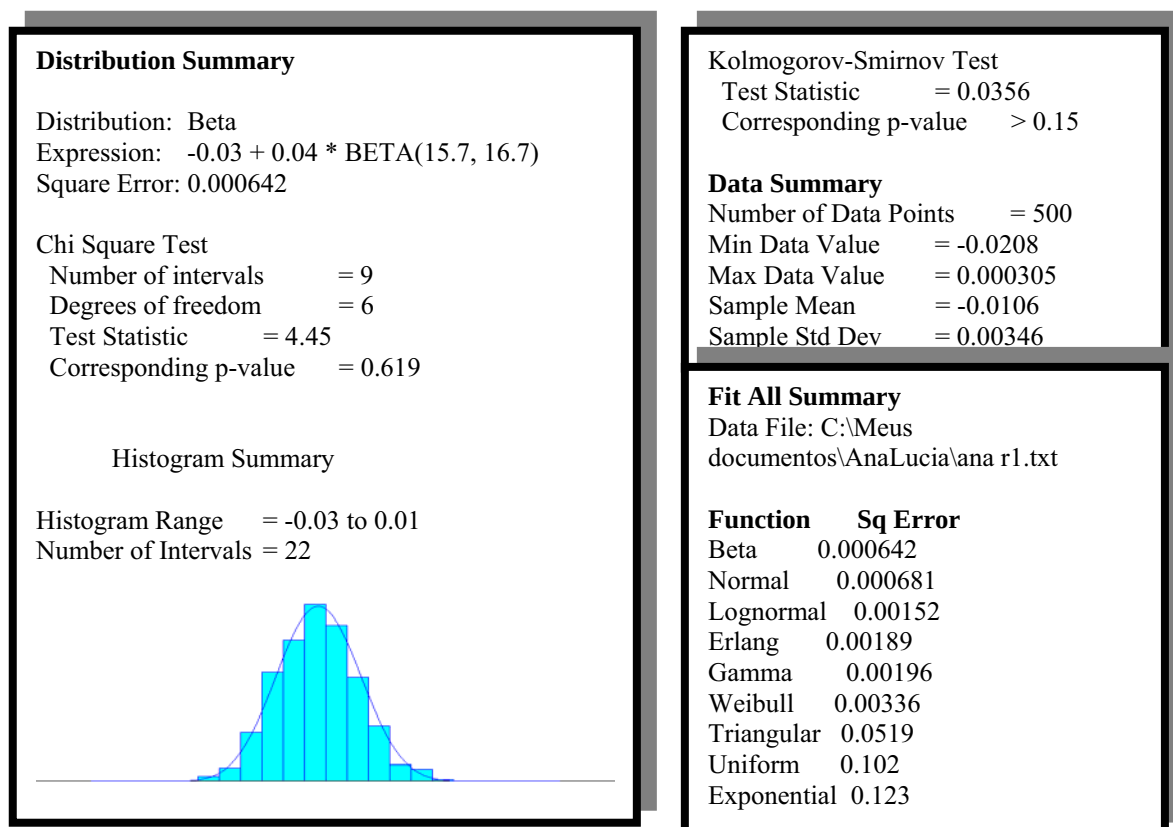


Figura 6.62 - Distribuição de R1 para história de tempo de 10 segundos.

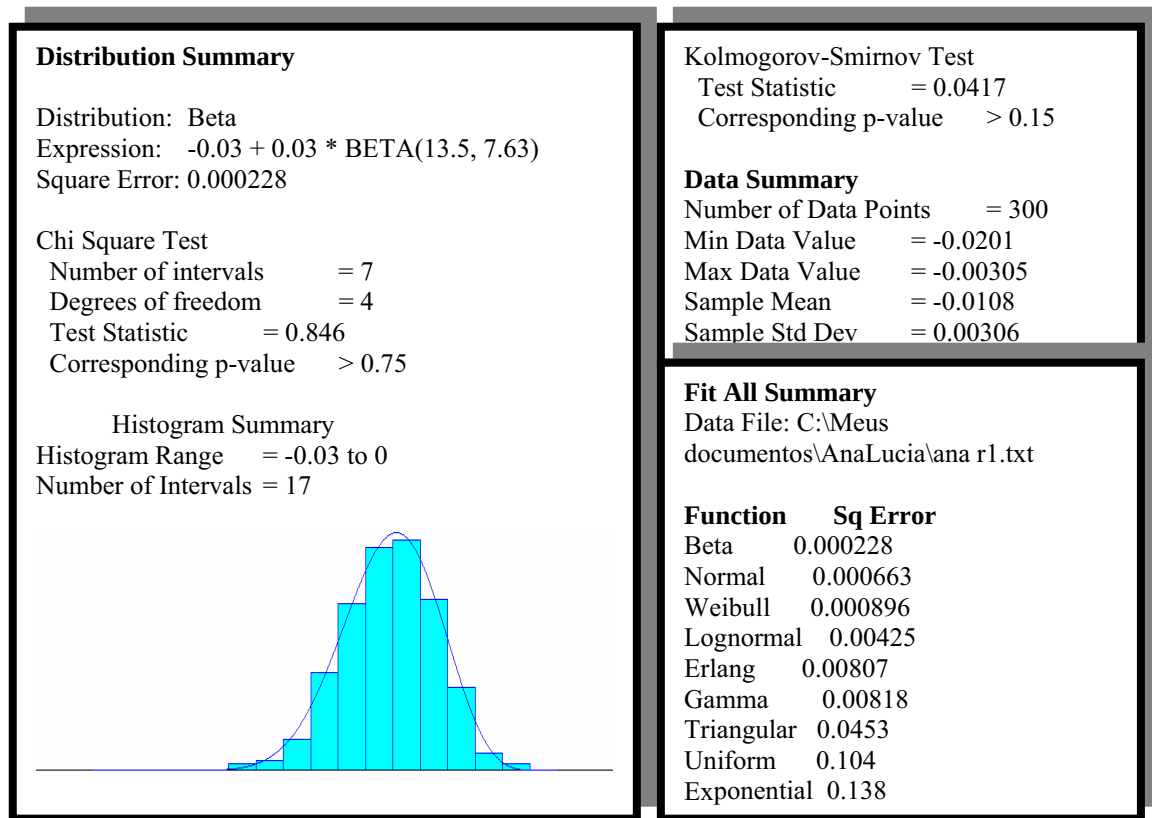


Figura 6.63 - Distribuição de R1 para história de tempo de 6 segundos.

Os dados analisados na pré-redução recebem a influência causada pelo vento no modelo aerodinâmico, capturada pelos sensores de células de carga instaladas na balança externa abaixo da seção de ensaios. Para melhoria do processo de obtenção de resultados aerodinâmicos, é recomendável que essas influências sejam conhecidas antes de processar a redução de dados, as quais podem implicar na mudança de amostragem dos dados, ou até mesmo em nova aquisição de dados, facilitando a decisão de limitação ou liberação dos dados devido ao processo de análise de pré-redução.

Uma indicação de outra distribuição que não a normal pode ocorrer nos dados de forças e momentos. No entanto, o desvio padrão apresentado deve permanecer na ordem almejada pelos especialistas em ensaios em túnel, que atualmente é de 0,5%. Assim, os testes estatísticos foram aplicados como um processo de pré-redução de dados, para determinar se uma amostra de dados de força ou de momento é originada

de uma população que segue uma distribuição normal (GASPERS, MUHLSTEIN, 1998).

O último dos subprodutos construído e testado, utilizado para os ensaios em túnel, integra todos os subsistemas, mas este ainda é considerado em desenvolvimento até a validação dos resultados após processamento usando o software de redução de dados do túnel de vento (SRD), no qual são aplicados os testes de integração, para avaliação funcional de todos os subsistemas do novo sistema de medidas. Os resultados destes testes e daqueles para liberação dos subprodutos ou componentes, são submetidos para avaliação dos clientes. Este trabalho é de responsabilidade do integrador.

O papel do integrador é incorporar os elementos dos subprodutos construídos ao sistema de medidas do TA-2, e criar os casos de testes necessários para comprovação dos resultados, quando comparados aos obtidos com o sistema de software da plataforma HP. A figura 6.64 mostra o papel do integrador com suas atividades e artefatos correlatos.

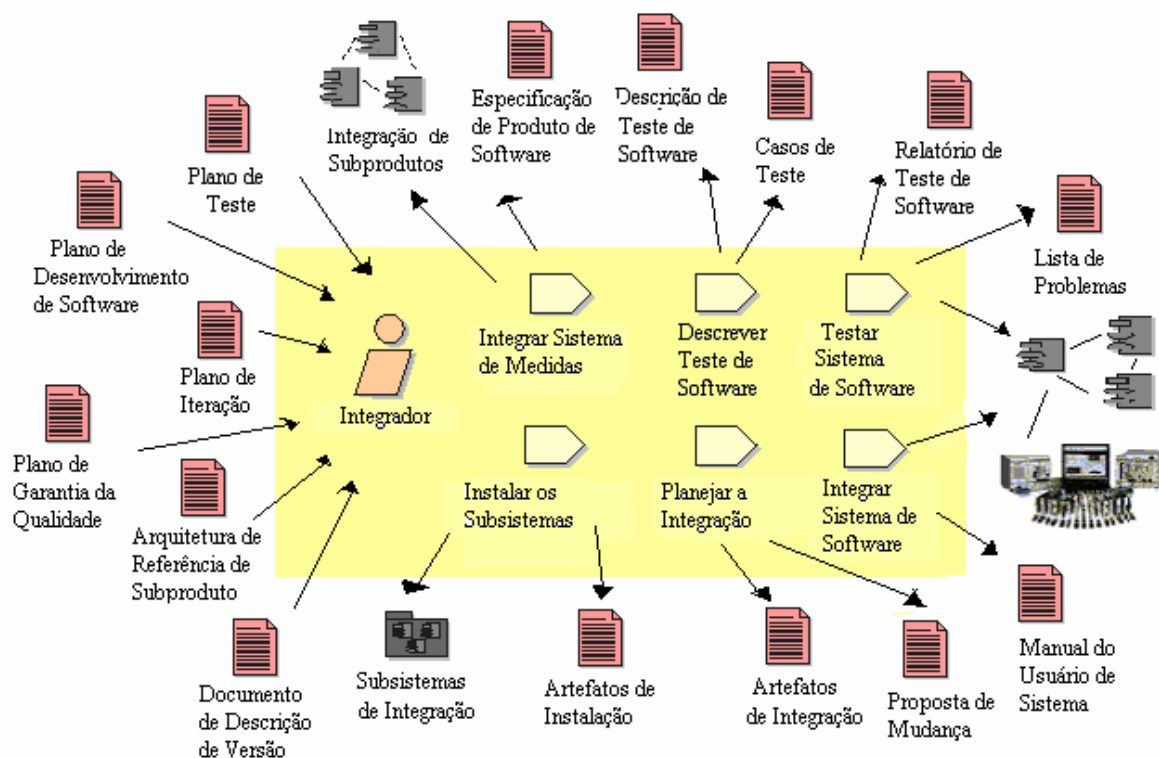


Figura 6.64 - Papel do integrador no TA-2

O papel do integrador é combinar os componentes de software liberados pelo revisor de software e implementador, proceder a interação com o hardware, a descrição dos testes de integração e testar o sistema de software de forma integrada. Também é responsável por planejar a integração com as instalações dos subsistemas, gerar o manual do usuário de sistema e descrever a especificação do produto de software.

Na modernização do TA-2, a integração de alguns subsistemas, foi feita com os subprodutos da construção C2 e C3 e o novo software de redução de dados. O subsistema de motor e hélice foi integrado na construção C4, sendo considerado o produto final de aquisição de dados, atualmente em fase de testes finais de integração com o software de redução de dados, para aprovação dos resultados de ensaios pelos clientes.

6.4.3.8 Implementação e testes do subproduto de software de redução de dados do túnel de vento – SRD

A reengenharia neste trabalho também está associada ao re-projeto dos processos de engenharia de software do túnel de vento, visando melhorar a eficiência destes processos. Esta reengenharia, devido às falhas na plataforma de hardware HP, e a falta de apoio nas manutenções do software, não pôde ser acomodada por meio da manutenção normal dos programas em BASIC. Por isso, neste trabalho se executou a reestrutura do projeto, a tradução para a linguagem C e outras atividades. O sistema de medidas antigo com seus softwares associados é a especificação para o novo sistema de software de redução de dados.

As práticas da METAPLI aplicadas no processo de reengenharia estão voltadas para a conversão dos programas em BASIC para a linguagem C, com atualização da plataforma de hardware, a análise dos métodos de engenharia extraídos deste processo

de redução de dados, a análise e modificação da estrutura do programa para facilitar o entendimento, o agrupamento de forma apropriada de partes que se relacionam nos programas, com alguma transformação de arquitetura de software e a reengenharia dos dados para refletir as reestruturações feitas nos novos softwares, mantendo atualizada a documentação do sistema de software.

A aplicação via METAPLI do processo de reengenharia do software de redução dos dados teve início com análise de domínio, para obter conhecimento sobre o funcionamento do sistema de medidas do TA-2 antigo, que serve de base para a determinação dos requisitos de engenharia de software. A implementação deste subproduto foi feita usando a linguagem de programação C, levando-se em conta os processos de engenharia aplicáveis, as funções e os métodos de forma similar aos softwares existentes em linguagem BASIC, a alteração dos processos de engenharia de software, das estruturas de dados e de projeto.

No processo de reengenharia do software de redução de dados foram implementadas as melhorias na estrutura do programa e dos dados, para diminuir a complexidade da estrutura lógica, simplificando e tornando-o compreensível, abolindo os comandos “*go to*” e reescrevendo os códigos em termos condicionais simples tais como: “*while*”, “*if-then-else*”, entre outros. Também se agruparam os dados, de forma a proporcionar funções de construção e de acesso facilmente recuperáveis e de interpretação rápida. Praticamente todas as denominações dos dados foram alteradas, para facilitar a compreensão e a identificação das entidades lógicas.

Os comentários e as atualizações dos programas ou unidades de software são documentados como parte do processo de reestruturação dos dados. Os dados que são modificáveis de um programa de ensaio para outro, constam em depósitos de dados específicos (*reduc.h*, *dred.h*), possibilitando efetuar as mudanças de valores referentes ao novo modelo aerodinâmico, refletindo as condições do espécime e dos ensaios

solicitadas pelos clientes. Os detalhes de parte do código e dos dados reestruturados nos depósitos de dados do SRD podem ser observados nas figuras de 6.65 e 6.66.

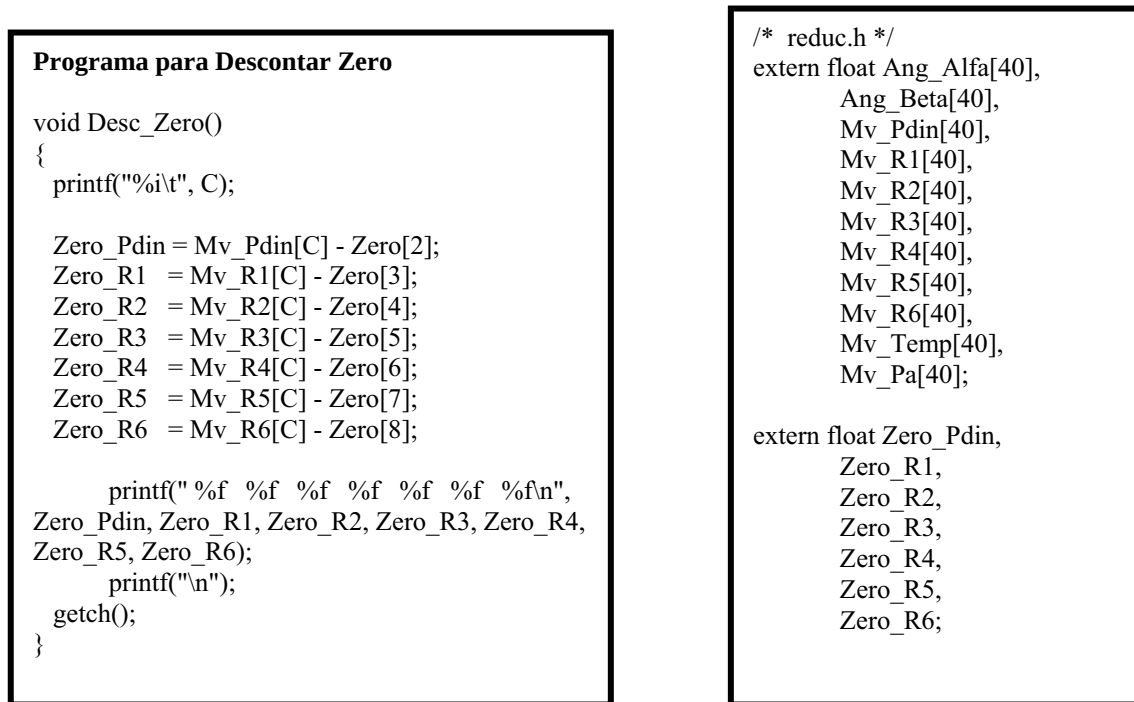


Figura 6.65 - Reestrutura de dados e código que desconta zero.

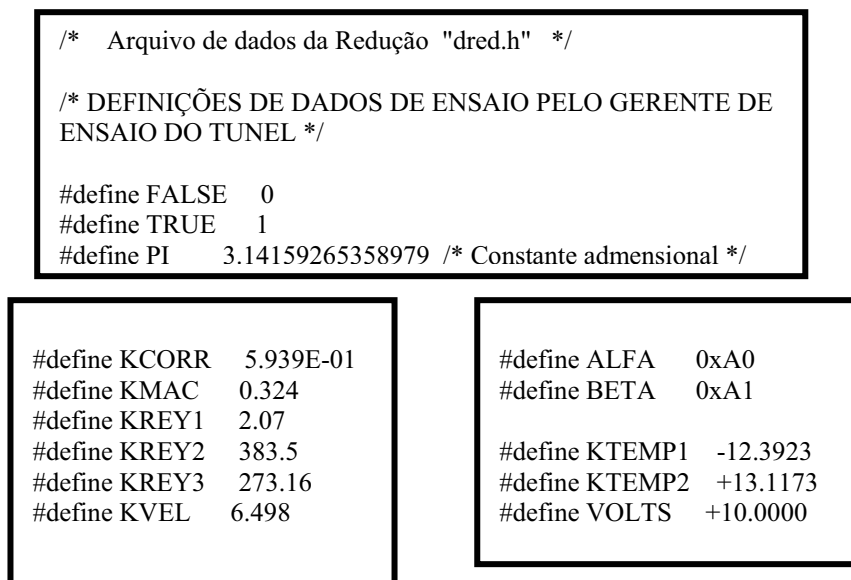


Figura 6.66 - Reestruturação de dados modificáveis em campanhas de ensaios.

O processo de geração de documentos é iterativo e as informações obtidas durante o ciclo de vida de desenvolvimento com reengenharia são utilizadas para as reestruturações de projeto e dados. A edição do código compreende a documentação do código definida no plano da qualidade, bem como outras atividades referentes à mídia contendo as relações de componentes identificados, conforme estabelecido neste mesmo plano.

Os programas foram reorganizados e as partes relacionadas, por exemplo, ao cálculo de ZAB, ZQ, forças e momentos, entre outros, foram coletadas e consideradas um único módulo funcional ou programa, os quais juntamente com outros módulos de apoio ao processo, e de abstração de dados, compõem o novo software de redução de dados do TA-2. Para validação do SRD, foram preparados e aplicados vários casos de teste, sendo que neste trabalho, é apresentado aquele referente ao processo de obtenção da matriz de calibração.

O caso de teste aplicado abrange os dados da calibração transferidos da HP e aqueles obtidos com o SBA e SCB, os quais são os mais importantes para garantir resultados aerodinâmicos confiáveis. Em vista de calibração, o método adotado é aplicado para medidas de múltiplas componentes de forças e momentos (NOGUEIRA, 1980), pelas quais são averiguadas e avaliadas as influências aerodinâmicas globais, nos dados processados com uso do SRD.

O novo software de redução de dados abrange os métodos implementados para obtenção da matriz de calibração. Dentre esses métodos estão as regressões lineares, quadráticas e cúbicas dos dados de aquisições para obter os coeficientes ou constantes de calibração, e o método de medidas de múltiplas componentes com interações, para determinar a matriz de calibração.

Para aplicação deste caso de teste para obtenção de matriz de calibração, diversos outros casos de teste foram aplicados separadamente com a validação dos resultados.

Dentre eles podemos destacar o caso de teste para cálculo das forças e momentos, e o caso de teste para calibração de células de carga, no ambiente de ensaio.

6.4.3.8.1 Caso de teste para obtenção de matriz de calibração

O caso de teste para obtenção da matriz de calibração usa os dados dos carregamentos com as respectivas cargas, em que são comparados os resultados obtidos com o PC e HP. Neste trabalho são demonstrados os resultados com os carregamentos da força de arrasto (F_1) da célula de carga de 300 libras de série 91481, nos quais, devido ao grande volume de dados, os carregamentos com carga zero foram desprezados.

O objetivo deste caso de teste é demonstrar que a matriz de calibração obtida com os softwares de redução de dados da HP e PC são equivalentes. São então apresentados os resultados intermediários obtidos, que comprovam os resultados de calibração da força F_1 . Observa-se que os resultados apresentados, usando ambas as plataformas demonstram um desvio padrão similar. Deve ser mencionado que, se colocados todos os carregamentos, inclusive aqueles com zero o desvio padrão é melhorado.

Para cada combinação de graus 1 ou 2 existem dois carregamentos combinados, com cargas mínimas, máximas ou zero já calculadas computacionalmente, por outro módulo do software de redução de dados, para o ângulo ALFA. Para os carregamentos em ângulo alfa, são efetuados 73 carregamentos sem repetição, para os quais os dados foram armazenados e posteriormente processados por outro módulo do software de redução de dados. As combinações de grau 1 são mostradas na tabela 6.12.

Tabela 6.12 - Combinações de grau 1

FORÇA/MOMENTO	CARGA 1	CARGA 2
Arrasto	Máximo	Zero
Sustentação	Máximo	Mínimo
Força Lateral	Máximo	Mínimo
Arfagem	Máximo	Mínimo
Guinada	Máximo	Mínimo
Rolamento	Máximo	Mínimo

Nas combinações de grau 1 ou com duas cargas por componente são obtidos 12 carregamentos combinados:

$$C_6^1 = 6 * 2$$

Nas combinações de grau 2 com duas cargas por componente são obtidos 60 carregamentos combinados:

$$C_6^1 * C_5^1 = 30 * 2$$

Além das combinações de graus 1 e 2, há um carregamento nulo, compondo o correspondente aos 73 carregamentos, com as cargas estabelecidas na lista de carregamentos para calibração em ângulo alfa.

$$Carregamentos = 12 + 60 + 1 = 73$$

Para a calibração em ângulo Beta são feitos 219 carregamentos combinados, que abrangem tanto os dados das componentes com carregamento acima de zero, quanto aqueles dados das componentes descarregadas ou com carregamento zero.

Como mencionado, a obtenção da matriz de calibração envolve diversos outros casos de teste já aplicados, com a obtenção dos dados e resultados em milivolts, lidos de arquivos armazenados em forma de planilhas, visando a aceitação das funcionalidades do software de redução de dados, integrado ao sistema de medidas. A tabela 6.13 apresenta o resultado de um destes casos de teste já aplicados para o cálculo de cargas F1 com software da HP e do PC, para ângulo ALFA, ou seja, com 73 carregamentos, porém apresentados aqui sem os carregamentos com zero.

Tabela 6.13 - Cálculo de cargas de F1 com software da HP.

Número do Carregamento F1	Carga Carretada (Kgf)	Carga Calculada (Kgf)	Diferença entre Carga Calculada e Carregada
8	50	49,9779	-0,0221
9	100	99,9856	-0,0144
18	50	49,9395	-0,0605
19	100	99,9496	-0,0504
26	50	49,8923	-0,1077
27	100	100,0411	0,0411
28	100	100,0063	0,0063
29	100	100,0105	0,0105
30	100	100,0277	0,0277
31	100	100,0133	0,0133
32	100	100,0095	0,0095
33	50	50,0517	0,0517
34	50	50,0572	0,0572
35	50	50,1181	0,1181
36	50	50,0587	0,0587
37	50	50,0581	0,0581
52	50	49,956	-0,044
53	100	100,0085	0,0085
60	50	49,8898	-0,1102
61	100	99,9749	-0,0251
70	50	50,003	0,003
71	100	99,9742	-0,0258

O desvio padrão obtido das cargas calculadas pelo software de aquisição de dados com HP com as cargas da tabela 6.13 e com todas as cargas, contando com aquelas com zero é apresentada na tabela 6.14.

Tabela 6.14 - Valores dos desvios padrão de cargas parcial e total de F1 com HP

Desvio Padrão Carga Parcial HP	Desvio Padrão Carga Total HP
0,055006	0,036991

Tabela 6.15 - Cálculo de cargas de F1 com software da PC.

Número do Carregamento F1	Carga Carretada (Kgf)	Carga Calculada (Kgf)	Diferença entre Carga Calculada e Carregada
8	50	49,9286	-0,0714
9	100	99,9561	-0,0439
18	50	49,952	-0,048
19	100	100,0124	0,0124
26	50	49,9691	-0,0309
27	100	100,0259	0,0259
28	100	99,9644	-0,0356
29	100	100,0069	0,0069
30	100	100,0483	0,0483
31	100	100,0077	0,0077
32	100	99,9741	-0,0259
33	50	50,0469	0,0469
34	50	50,0493	0,0493
35	50	50,1063	0,1063
36	50	50,0507	0,0507
37	50	50,0659	0,0659
52	50	49,9863	-0,0137
53	100	100,0172	0,0172
60	50	49,9409	-0,0591
61	100	100,0181	0,0181
70	50	49,9035	-0,0965
71	100	99,9684	-0,0316

O desvio padrão obtido das cargas calculadas pelo software de aquisição de dados com PC com as cargas da tabela 6.15 e com todas as cargas, contando com aquelas com zero é apresentado na tabela 6.16.

Tabela 6.16 - Valores dos desvios padrão de cargas parcial e total de F1 com PC

Desvio Padrão Carga Parcial PC	Desvio Padrão Carga Total PC
0,050211	0,03322

As preocupações das análises com a aplicação dos casos de teste deste trabalho são com a similaridade de resultados entre ambas plataformas, processados pelo SRD para a validação dos processos de reengenharia para obtenção dos dados e resultados. Conforme estabelecido pelos especialistas em ensaios em túnel, estes casos de teste foram aplicados em separado, sem as condições otimizadas do túnel de vento e do modelo aerodinâmico.

O método implementado computacionalmente para obtenção da matriz de calibração usando o SRD, é usado na determinação dos coeficientes aerodinâmicos, com o uso destes dados armazenados das três forças e três momentos. Este método de calibração do sistema de medidas de múltiplas componentes com interação (NOGUEIRA, 1980) usa a combinação de carregamentos, descarregamentos ou carregamento zero sem repetição e conta com a equação 6.2:

$$F_k = T_k + \sum_{i=1}^6 A_k^i * R_i + \sum_{i=1}^6 \sum_{j=1}^6 B_k^{ij} * R_i * R_j \quad (6.2)$$

Na equação 6.2, implementada computacionalmente no SRD, os termos lineares são causados por uma não perfeita isolamento entre as componentes de forças e momentos, assimetria dos mecanismos de ensaios, posicionamento de sensores, entre outros, que influenciam no resultado do ensaio do modelo aerodinâmico. Os termos quadráticos são devidos a deformações causadas pelas cargas na geometria e os termos

de ordem superior a 2, se aparecerem deverão ser analisados quanto aos problemas de projeto, falha de fabricação do modelo aerodinâmico ou outros itens do projeto real.

A equação 6.2 fornece 162 coeficientes a determinar, variando o k de 1 a 6, para obter seis grupos de 27 combinações. O método em forma de matriz pode ser modelado conforme mostrado a seguir:

$$\begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_{27} \\ b_1 & b_2 & b_{27} \\ \vdots & \vdots & \vdots \\ f_1 & f_2 & f_{27} \end{bmatrix} * \begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ \vdots \\ R_1^2 \\ R_1 R_2 \\ \vdots \\ R_1 R_6 \\ R_2^2 \\ \vdots \\ R_6^2 \end{bmatrix}$$

As componentes aerodinâmicas correspondem às três forças e três momentos, que constituem as grandezas distintas denominadas F1, F2, F3, F4, F5 e F6, e os sinais de saída denominados R1, R2, R3, R4, R5 e R6, que produzem 27 combinações dois a dois, carregando-se duas componentes e medindo a influência em todas as outras na calibração. A matriz de calibração final é calculada através das 27 combinações obtidas com o modelo de interação (NOGUEIRA, 1980), no qual o resultado deste cálculo representa os coeficientes de calibração das forças e momentos, esses coeficientes esses obtidos ao quadrado, e em 15 grupos com combinação das seis componentes, dois a dois.

$$C_6^1 = 6 \quad R_1^2, R_2^2, R_3^2, R_4^2, R_5^2, R_6^2 = 6 \quad C_6^2 = \frac{6 * 5 * 4!}{2!4!} = 15$$

$$\text{Combinações} = 15 + 6 + 6 = 27$$

A matriz de calibração apresentada neste trabalho foi obtida com o novo software de redução de dados em linguagem C, e usando os dados de aquisição obtidos do NACA 0012 com os softwares SBA da plataforma PC, no sistema de medidas integrado. Com este resultado do caso de teste para obtenção da matriz de calibração, os novos softwares do sistema de medidas do TA-2 foram aprovados pelos clientes, com base nestes resultados. A matriz de calibração obtida é apresentada na tabela 6.17.

Tabela 6.17- Resultados dos testes de obtenção da matriz de calibração

1	R1	+2,0292E+02	-9,4167E-01	+7,7101E-01	-8,9793E-01	-4,2310E-01	-1,4914E-01
2	R2	+3,1318E-01	-2,7455E+02	-3,5191E-01	-7,3249E-01	-2,7426E+00	-2,4502E+00
3	R3	+2,7236E-01	+2,6857E-01	+6,2692E+02	-8,5739E+00	-6,6761E-01	+1,9485E-01
4	R4	+7,6865E-02	-1,0542E-01	+2,4094E-01	+1,4988E+02	+1,1951E-01	+3,9691E-02
5	R5	-4,5932E-02	+3,3798E-02	-7,4666E-02	-9,2139E-02	-7,5451E+01	+1,7001E-02
6	R6	-2,3165E-02	-4,8331E+01	-9,4808E-02	+3,4277E-012	-2,5972E-02	-6,8546E+01
7	R1.R1	+1,4007E+00	-2,1832E-02	+3,5752E-03	+1,6562E-01	+8,1778E-01	+1,6960E-02
8	R1.R2	+2,7391E-02	-3,8442E-01	-4,0256E-02	-3,4837E-01	-2,2785E-02	-1,6902E-01
9	R1.R3	+1,6384E+00	+8,9691E-02	-3,4449E-02	-3,5946E-01	+2,0237E+00	+1,0646E-01
10	R1.R4	+2,0437E-01	-1,1037E-01	-7,3540E-01	-4,9078E-02	+2,0794E-01	-1,1073E+00
11	R1.R5	+1,1968E-01	-3,2120E-02	+1,6228E-01	+1,3096E-01	-3,9819E-01	-6,20407E-02
12	R1.R6	-6,4699E-02	-1,1253E-02	-2,2598E-01	+1,7699E+00	-1,8204E-02	-9,3574E-01
13	R2.R2	-2,3998E-03	+4,0505E-01	+4,7166E-01	-3,7658E-01	+2,9057E-02	-3,0963E-02
14	R2.R3	-6,2905E-02	-7,9181E-01	+4,1035E-01	+2,5064E+00	-1,7574E-01	+3,9574E-01
15	R2.R4	-1,6502E-01	-1,3657E-02	+3,5884E-01	+1,5472E-01	-4,1090E-02	-7,3521E-02
16	R2.R5	-1,7390E-02	+1,5719E-02	-4,5612E-02	+4,6130E-03	-3,2570E-03	-5,4416E-01
17	R2.R6	+2,9027E-01	-3,1012E-02	+1,3563E-01	+1,1290E-02	-1,9501E+00	-3,3649E-01
18	R3.R3	+1,2163E-01	+4,7432E-01	+7,1502E-01	-1,5655E-01	+5,2972E-01	+1,4043E-01
19	R3.R4	+2,1351E-01	-1,7668E-01	+1,8394E+00	+2,1386E+00	+5,5406E-02	+8,7819E-02
20	R3.R5	-6,4079E-02	+4,3983E-02	+3,8948E-02	-6,3628E-02	-1,1348E+00	+6,8095E-02
21	R3.R6	-2,2759E-01	-6,8079E-02	+3,2525E-01	+3,0468E-01	-2,0721E-01	-1,8672E-01
22	R4.R4	+1,3462E-01	+4,5435E-01	+3,7369E-01	-4,2612E-01	-2,2462E-02	+1,1773E-01
23	R4.R5	+5,8425E-02	-5,5715E-02	+2,7817E-01	-1,4248E-01	-2,0320E-01	+1,4911E-03
24	R4.R6	+1,5197E-01	-2,1405E-01	,+9,2989E-02	+1,1722E-01	+5,3127E-01	+1,5294E-01
25	R5.R5	+6,7803E-02	+9,5712E-02	+8,4002E-02	-4,1514E-02	+5,5874E-02	+4,1934E-01
26	R5.R6	+2,6873E-03	-1,2119E-01	-1,2959E-02	+3,0681E-02	-3,1816E-02	-7,0949E-02
27	R6.R6	+1,0319E-01	+8,1741E-02	-5,8498E-02	-1,2454E-01	-3,2001E-01	-1,4218E-01

Finalizando o trabalho de reengenharia do software de redução de dados, são apresentados os resultados com SRD de um dos ensaios com o modelo aerodinâmico NACA 0012. Estes complementam a integração do sistema de software, as medidas sendo obtidas com o SBA e a redução, com o SRD, e são mostradas na tabela 6.18.

Tabela 6.18 - Resultados de ensaios com o modelo aerodinâmico NACA0012

ALFA	CD	CY	CL	CM	CR	CN	Q	VEL	MACH	RE*10 ⁶
-2,0	-0,0020	0,009	-0,044	-0,005	0,0362	0,0196	358,5	80,6	0,232	2,380
-1,0	-0,0020	0,010	-0,044	-0,005	0,0375	0,0198	363,3	81,2	0,234	2,393
0,0	-0,0019	0,010	-0,043	-0,005	0,0369	0,0199	360,2	80,9	0,233	2,381
1,0	-0,0018	0,011	-0,043	-0,005	0,0366	0,0204	362,0	81,1	0,233	2,386
2,0	-0,0017	0,012	-0,042	-0,005	0,0382	0,0208	361,9	81,1	0,233	2,385

Com os resultados apresentados podemos afirmar que a metodologia deste trabalho abrange o desenvolvimento dos softwares do novo sistema de medidas do túnel de vento, e pode ser estendida para desenvolver outros sistemas de software. Através dela muitas práticas que implicam em mudanças técnicas e gerenciais, auxiliam na avaliação e assimilação de novos conhecimentos pelo grupo envolvido, de modo a evitar a dependência de conhecimentos gerados por fontes cujo acesso seria limitado.

As práticas da METAPLI são implantadas como uma combinação entre as ações dos processos unificado e tradicional, as quais sustentam a nova forma de trabalho para desenvolvimento de software do TA-2 e envolvem:

- Um modelo gerencial com conjunto de papéis e responsabilidades definidos.
- Mecanismos de qualidade associados aos processos e aos produtos e subprodutos de software.
- Ênfase em técnicas de reuso e de reengenharia.
- Aumento da produtividade.

- Soluções rápidas suportadas pela reutilização de “vi” e bibliotecas de ambiente de programação visual comercial.
- Soluções com desenvolvimento de subprodutos adicionais para agilizar a operacionalidade do TA-2.

Todos esses esforços juntamente com aplicação de reuso e de reengenharia, métodos gráficos usados para a modelagem do sistema de software, enfoques de linguagens estruturadas e de ambientes de programação visuais comerciais; proveram aos envolvidos no projeto de modernização do TA-2, oportunidades adicionais de aplicação de métodos, técnicas, processos e práticas de engenharia de software.

Os softwares do sistema de medida do TA-2 contemplam a aquisição automática dos dados, geram os arquivos de saída com dados organizados e apresentam a maioria dos dados na forma gráfica ou numérica ao operador ou usuário. As medidas e o controle dos subsistemas foram implementados e testados no TA-2, com comparações dos dados obtidos das plataformas PC e HP, e com apresentação de resultados similares usando o sistema de medidas modernizado, montado e testado no TA-2.

A capacidade dos integrantes do túnel de vento, no que tange ao desenvolvimento de software, aumentou gradativamente e foi de fundamental importância para a continuidade dos ensaios no TA-2. O atendimento aos requisitos especificados pelos especialistas em ensaios em túnel foi satisfeito, e significou superar as deficiências de desempenho funcional que afetavam a eficácia operacional do sistema de medidas, ao qual pertence os softwares de aquisição e de redução de dados.

CAPÍTULO 7 DISCUSSÕES DOS RESULTADOS

As metodologias apresentadas na literatura compreendem ações voltadas para diversos tipos de softwares. No entanto, as metodologias ágeis são apontadas para projetos considerados pequenos, que a modernização do sistema de software do túnel de vento do CTA não se enquadra. Entretanto o enfoque de um desenvolvimento voltado para as pessoas, contribuiria com ações importantes em um sistema de grande porte como a modernização do Túnel de Vento Aerodinâmico nº 2 - TA-2, do CTA.

As metodologias tradicionais são empregadas em sistemas de softwares aeroespaciais com sucesso. No entanto, os requisitos destes sistemas em termos operacionais são definidos desde o início de um projeto, que geraria restrições e limitações para manter operacional o túnel de vento durante a atualização de software. Portanto, sua implantação na íntegra não seria adequada para o sistema de software do TA-2.

Neste trabalho se criou uma nova metodologia de projeto, a METAPLI, considerando as ações das metodologias tradicionais e ágeis, no que tange ao desenvolvimento dos softwares de aquisição e redução de dados, com enfoque em processos, documentos e pessoas. Esta nova metodologia de projeto se mostrou vantajosa em relação às metodologias tradicionais e ágeis porque com suas ações ajustadas ao grupo de projeto, possibilitou reduzir significativamente a resistência às mudanças e o esforço dos envolvidos, com aprendizado e uso dos processos, práticas, técnicas e tecnologias.

O trabalho de implantação da metodologia de projeto permitiu a mudança cultural do grupo de projeto em relação à engenharia de software, de forma que este grupo cumprisse com sucesso todas as etapas previstas nesta metodologia que são:

- Levantamento das informações necessárias do estado atual do sistema de medidas, para iniciar o projeto de modernização do TA-2.

- Planejamento das práticas, processos, atividades, métodos e pessoas para execução dos trabalhos de renovação do sistema de medidas.
- Desenvolvimento do sistema de software cumprindo as atividades de requisitos, arquitetura de hardware e de software, implementação e teste dos subprodutos, deixando-os operacionais para uso nos ensaios aerodinâmicos requeridos.
- Entregas e aceitações freqüentes destes subprodutos de software para aplicação operacional nos ensaios aerodinâmicos.
- Testes e Validações dos Resultados por comparação com resultados experimentais com uso da plataforma HP.
- Realinhamento dos Dados e Processos adotando práticas para melhoria da qualidade dos processos e dos dados a cada implementação e teste com os subprodutos.
- Lições aprendidas diversas descritas e documentadas conforme atividades estabelecidas via METAPLI e norma de desenvolvimento de software.
- Reuso do conhecimento no desenvolvimento de cada subproduto.
- Demonstrações aos clientes por intermédio de relatórios de ensaios, com descrições e apresentações de resultados, com uso de instrumentações, equipamentos, subsistemas e subprodutos de software modernizados.

Com a inovadora metodologia de projeto aplicada na modernização do TA-2 para o desenvolvimento dos softwares de aquisição e redução de dados com técnicas de reuso e reengenharia, atingiu-se as seguintes perspectivas:

- Permitir a modelagem e construção dos softwares de aquisição e de redução de dados.
- Fornecer subprodutos executáveis, permitindo a operacionalidade do TA-2.
- Permitir a aplicação de ações adequadas dos processos incremental e iterativo com o processo em cascata.
- Facilitar a documentação do sistema de medidas e dos softwares com uso de norma.
- Permitir a adequação da linguagem de modelagem, tanto para modelagem gerencial quanto para a modelagem técnica, incluindo um conjunto específico de requisitos necessários para representar o novo sistema de medidas.

Com o uso das práticas descritas e implantadas via METAPLI houve melhoria da qualidade no projeto nos seguintes elementos:

- Qualificação da mão de obra.
- Treinamento dos operadores do TA-2.
- Subprodutos de software e do sistema de medidas final.
- Nível tecnológico dos produtos empregados e obtidos.
- Capacidade dos envolvidos para a produção e alteração de software.
- Flexibilidade do processo de desenvolvimento de software.
- Disponibilidade das informações sobre o desenvolvimento do sistema de medidas.
- Reuso de Tecnologia viável.
- Medição e entendimento do impacto de reuso de software nos produtos e processos de desenvolvimento.

- Entendimento do que é e como implantar reuso em cada fase do ciclo de vida.
- Coleta e acúmulo de conhecimento para evolução de tecnologias no futuro.

A inovadora abordagem gerencial e técnica contribuíram para geração da documentação requerida pela norma DOD-STD-2167A, e outros documentos complementares do projeto do sistema de software. Com a implantação da METAPLI, os subprodutos de software gerados durante o desenvolvimento, foram de fundamental importância para manter a operacionalidade do TA-2, provendo dados e resultados representativos dos ensaios em túnel de vento.

Portanto, foi implantado com sucesso o projeto do novo sistema de software. Os investimentos e os esforços nas melhorias de processos de engenharia, de software e gerenciais foram evidenciados. Também foram minimizados os erros funcionais que afetavam os dados e resultados aerodinâmicos dos subsistemas. As mudanças culturais foram conseguidas com esforços do grupo de projeto, de modo a agilizar a implementação dos processos de engenharia e obter os ganhos almejados, atingindo as perspectivas deste trabalho.

CAPÍTULO 8 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foram analisadas diversas fontes de informações sobre metodologias tradicionais e ágeis, processos de desenvolvimento de software, sistema de medidas, ensaios em túnel de vento, reuso e reengenharia de software. Os processos, métodos, práticas, técnicas e capacitações aplicáveis ao empreendimento, são estabelecidos de forma clara e são praticadas e compreendidas pelos envolvidos na modernização do TA-2.

A METAPLI foi criada com ações adequadas das metodologias ágeis e tradicionais, e implantada com sucesso via modelagem do desenvolvimento dos softwares de aquisição e redução de dados com aplicação de reuso e reengenharia. Os projetistas dos softwares atenderam as funcionalidades necessárias, provendo a automação das medidas e de controles dos subsistemas que compõem o sistema de medidas modernizado do TA-2, mantendo a operacionalidade do túnel durante todo o projeto. Com aplicação das ações da METAPLI houve aumento do conhecimento, da produtividade, da qualidade e da confiabilidade dos dados e resultados de ensaios aerodinâmicos, representando a engenharia de software de um novo sistema de medidas em túnel de vento.

Através desta metodologia de projeto foram implantados os novos softwares de aquisição e redução de dados, considerados os elementos mais importantes do sistema de medidas, que juntamente com os demais subprodutos de softwares e a documentação representam os resultados obtidos com este trabalho. A implantação da METAPLI facilitou a gerência e a engenharia do projeto do novo sistema computacional complexo. Também via esta metodologia de projeto se aplicou normas de desenvolvimento para documentação destes softwares, e assim foram transferidos os conhecimentos para o novo projeto. Durante o ciclo de vida deste projeto de modernização do sistema de medidas, todo conhecimento foi disponibilizado e divulgado aos envolvidos podendo ser reutilizado em trabalhos futuros.

A implantação da METAPLI possibilitou representar de forma consistente o projeto de desenvolvimento do sistema de software do TA-2, e foi utilizada em todo o ciclo de vida do desenvolvimento. O emprego das práticas prescritas na metodologia possibilitou a disponibilidade de informações e recursos de forma organizada, a verificação e avaliação da eficácia dos processos aplicados na atualização e revitalização do sistema de medidas de um túnel de vento. Assim, conclui-se que foram conseguidos com sucesso os aspectos principais, que estimularam o interesse deste trabalho, tais como:

- Sanar as necessidades de recursos computacionais e eletrônicos para substituir os anteriores que apresentavam falhas.
- Conscientizar a equipe de que as atualizações requeriam arquiteturas particulares como sistemas operacionais, hardwares e softwares mais adequados ao novo projeto do sistema de medidas.
- Implementar novas técnicas, ferramentas, arquitetura, ambiente de programação, e principalmente implantar os processos de engenharia de software, que melhoraram a velocidade de execução das atividades e operações do novo sistema de medidas.
- Usar ferramentas computacionais adequadas e dispositivos eletrônicos ou de instrumentação mais modernos, que permitiram progressos na integração dos diversos subsistemas ao sistema de medidas final.
- Permitir através da METAPLI a aplicação de práticas de engenharia de software com aplicação de reuso e de reengenharia, cuja abordagem possibilita avaliar os aspectos como desempenho, programabilidade, confiabilidade e futuras evoluções do novo sistema de medidas.

Em trabalhos futuros, a METAPLI pode ser utilizada em projetos voltados para outras áreas, as quais tem sistemas de software específicos. As etapas desta metodologia em nível de macro-ações que são: informação, planejamento, desenvolvimento, entregas e aceitações freqüentes, testes e validações de resultados, realinhamento dos dados e processos, conhecimento acumulado, conhecimento de

interesse, e demonstrações e qualificações dos clientes, provêm a abrangência necessária, para o emprego em outros empreendimentos.

A partir das práticas de cada macro-ação da METAPLI, os projetistas de novos sistemas de software podem adequar os processos, técnicas, métodos e novas tecnologias para o projeto específico. Mesmo as futuras evoluções do sistema de software do TA-2 abrangendo outro projeto de modernização do sistema de medidas, podem fazer uso destas macro-ações. Por exemplo, os projetistas podem fazer uso do diagrama da tartaruga nos novos processos, a abordagem de reuso do projeto anterior, outras normas de desenvolvimento com adequação da documentação, adequação das responsabilidades, artefatos e atividades específicas do grupo de projeto para uso de ferramentas CASE com abordagem de tempo real e orientação a objeto, entre outras adequações. Na proposta de um novo trabalho estaria um re-planejamento das atividades de engenharia de software e conseqüentemente a re-documentação, de forma a possibilitar o uso de tecnologias como ferramentas CASE, que contemplem a geração de códigos em *LabView*, para possibilitar o enfoque de orientação a objetos e UML, desde o início de um novo projeto.

Novos estudos referentes aos métodos de engenharia aplicáveis em túnel de vento são propostas de futuras modernizações, possibilitando trabalhos na área de sistema de software de tempo real. Também possibilita estudos na disciplina de análise de sinais, voltados a métodos matemáticos a serem implantados computacionalmente, para estudo de história de tempo de amostras de dados, medidos de ensaios no TA-2.

Propõe-se um trabalho de pesquisa em visualização dos ensaios em túnel, voltado para uso de imagens para levantamento completo do escoamento. A pesquisa em visualização de imagens permitiria aumentar significativamente a quantidade de informações sobre os ensaios, e proveria a visualização de detalhes do escoamento nos modelos aerodinâmicos submetidos aos ensaios. Este trabalho envolveria projeto de sistema de software complexo, com tecnologias avançadas.

REFERÊNCIAS

AGILE MANIFESTO, BECK, K.; BEEDLE, M.; BENNEKUM; A., COCKBOURN, A. CUNNINGHAM, W.; FOWLER, M.; GRENNING, J.; HIGHSMITH, J.; HUNT, A.; JEFFRIES, R.; KERN, J.; MARICK, B.; MARTIN, R. C.; MELLOR, S.; SCHWABER, K., SUTHERLAND, J., THOMAS, D. **Manifesto for Agile Software Development**, February 2001. Disponível em: <<http://agilemanifesto.org/>>, Acesso em: 10 Out. 2005.

AIAA-R-092-1-2002. American Institute of Aeronautics and Astronautics **Recommended Practice for Wind Tunnel Testing**. Part 1, Draft Review Copy (R-092-1-2002). AIAA Standards Series, 2002. 15p.

AIAA-R-092-2-2002. American Institute of Aeronautics and Astronautics **Recommended Practice for Wind Tunnel Testing**. Part 2, Draft Review Copy (R-092-2-2002). AIAA Standards Series, 2002. 45p.

AIAA-R-091-2002. American Institute of Aeronautics and Astronautics. **Recommended Practice for Calibration and Use of Internal Strain-Gage Balances with Application to Wind Tunnel Testing**. Draft Review Copy. AIAA Standards Series, Published by AIAA, 2002, 63p.

AIAA-S-071-1995. Standard, **Assessment of Wind Tunnel Data Uncertainty**, American Institute of Aeronautics and Astronautics, Washington, USA, 1995. 84p.

AMERICAN FOUNDRY SOCIETY (AFS). **Case Study: Process Definitions - Process Measuring How Does a Small Foundry Get Their ROI Off ISO/TS 16949:2002**, Schaumburg, USA, 2005, paper 05-176, Disponível em: <<http://www.moderncasting.com/archive/WebOnly/0306/WebOnly0306.pdf>> Acesso em: 12 Nov. 2006.

ANGIER, R. C., KELLEY, K. L., **Reuse Tools**, In: Aerospace Software Engineering: *a collection of concepts*. Washington DC. American Institute of Aeronautics and Astronautics, p. 441-452, 1991.

AQAP-150, NATO Quality Assurance Requirements for Software Development, **AQAP-150**, Edition 2, North Atlantic Treaty Organization, September, 1997.

AQAP-159, Guidance for the Use of AQAP-150, **AQAP-159**, North Atlantic Treaty Organization, September, 1997.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR ISO 15100**: Sistemas da qualidade – Aeroespacial – Modelo para garantia da qualidade em projeto, desenvolvimento, produção, instalação e serviços associados. Associação Brasileira de Normas Técnicas, Rio de Janeiro, Segunda Edição, 2004. 25p.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR/ISO 9000 Parte 3** Normas de gestão da qualidade e garantia da qualidade – diretrizes para aplicação da norma ISO 9001 ao desenvolvimento, fornecimento e manutenção de software. Rio de Janeiro; ABNT, 1991.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS & INTERNATIONAL ORGANIZATION FOR STANDARTIZATION. **NBR/ISO 9001:1994** Sistemas de gestão da qualidade - Requisitos. Associação Brasileira de Normas Técnicas. Rio de Janeiro, 1994.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS & INTERNATIONAL ORGANIZATION FOR STANDARTIZATION. **NBR/ISO 9001:2000** Sistemas de gestão da qualidade - Requisitos. Associação Brasileira de Normas Técnicas. Rio de Janeiro, 2000. 21p.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS & INTERNATIONAL ORGANIZATION FOR STANDARDIZATION & INTERNATIONAL ELECTROTECHNICAL COMMISSION. **NBR ISO/IEC 12207: 2000**. Tecnologia da Informação: Processos do Ciclo de Vida do Software. Rio de Janeiro; ABNT, 1998. 75p.

BARLOW, J. B., RAE, W. H., POPE, A. **Low-Speed Wind Tunnel Testing**, Third Edition, John Wiley & Sons, Inc., 1999.

BECK, K., FOWLER, M. **Planning Extreme Programming**. Boston, MA. Addison-Wesley, 2001. 160p.

BELLOQUIM, A. **Capability Maturity Model com Metodologia**. Infochoose nº 53. Ano III. Outubro 2002. Disponível em: <<http://www.choose.com.br/infochoose/artigos/>>, Acesso em: 10 Out. 2005.

BOEHM, B. W. **Spiral Model for Software Development and Enhancement**, **IEEE Computer**, v.21. nº 5, pp 61-72, 1988.

BROOKS, F., **No Silver Bullet: Essence and Accidents of Software Engineering**, Proceeding IFIP, IEEE CS Press, pp. 1069 – 1076; reprinted in IEEE Computer, Apr, 1987, pp. 10-19, citado por: Michel dos Santos em Metodologias Ágeis eXtreme Programming e Scrum para o Desenvolvimento de Software.

CAHILL, D. M.; **Application of uncertainty methodology for Wind Tunnel facilities at AEDC**, AIAA-1996-2216, AIAA, Advanced Measurement and Ground Testing Technology Conference, 19th, New Orleans, LA, June 17-20, 1996. 9p.

CAHILL, D. M.; **Implementation of New Wind Tunnel Data Reduction Techniques at the AEDC**, AIAA-2001-903, Aerospace Sciences Meeting and Exhibit, 39th, Reno, NV, Jan. 8-11, 2001

CMMI-a, Capability Maturity Model Integration (CMMI). **CMMISM for Software Engineering, version 1.1(CMMI-SW, V1.1)**., Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, Staged Representation, Improving processes for better products (CMU/SEI-2002-TR-029, PA 15213-3890), Aug. 2002. 638p. Disponível em: <<http://www.sei.cmu.edu/pubs/documents/93.reports/pdf/02tr029.pdf>>, Acesso em: 10 Out. 2005.

CMMI-b, Capability Maturity Model Integration (CMMI). **CMMISM for Software Engineering version 1.1 (CMMI-SW, V1.1)**. Pittsburgh, PA. Software Engineering Institute, Carnegie Mellon University. Continuous Representation, Improving processes for better products (CMU/SEI-2002-TR-028/ESC-TR-2002-028), Aug. 2002. 645p. Disponível em: <<http://www.sei.cmu.edu/pubs/documents/93.reports/pdf/02tr028.pdf>>, Acesso em: 10 Out. 2005.

DeMARCO, T. **Controle de Projetos de Software, Rio de Janeiro**, Editora Campus, 1989.

DEPARTMENT OF DEFENSE. **DOD-STD-7935A**, DOD Automated Information Systems (AIS) Documentation Standards, Oct. 31, 1988.

DEPARTMENT OF DEFENSE. **DOD-STD-2167A**, Defense System Software Development. Washington, D.C. U.S. Government Printing Office, 1988.

EMBREE, P. M.; KIMBLE, B. C **Language Algorithms for Digital Signal Processing**. Prentice-Hall, 1991, 496p.

FERNANDES, A. A., KUGLER, J. L. C. **Gerência de Projeto de Sistemas: uma abordagem prática**, Rio de Janeiro, Livros técnicos e Científicos, 1990.

GANE, C. P., SARSON, T. **Structured Systems Analysis: Tools and Techniques**, Prentice Hall, Englewood Cliffs, New Jersey, 1979.

GASPERS, P. A. Jr.; MUHLSTEIN, L. Jr. **Real-Time Minimization of Sample Time for Static Wind Tunnel Data to Maximize Productivity**, AIAA-98-2890, Advanced Measurement and Ground Testing Technology Conference, 20th, Albuquerque, NM, June 15-18, 1998.

GRAY, E. M., THAYER, R. H. **Requirements**. In: *Aerospace Software Engineering – A Collection of Concepts*. American Institute of Aeronautics and Astronautics, Inc. Washington, DC, 1991, p. 89-121.

HATLEY, D. J., PIRBHAY, I. A., **Estratégias para Especificação de Sistema em Tempo Real**, tradução Anna Terzi Giova, Makron Books do Brasil, São Paulo, 1991. 455p.

HEDIN, G.; BENDIX, L., MAGNUSSON, B., **Introducing Software Engineering by Means of Extreme Programming**, International Conference on Software Engineering, Proceedings of the 25th International Conference on Software Engineering, IEEE Computer Society, Washington, DC, USA, 2003. Disponível em: <<http://www.viktoria.se/fal/kurser/winograd-2004/p586-hedin.pdf>>, Acesso em: 10 Out. 2005.

IAE, WEB Site do Instituto de Aeronáutica e Espaço, 2005. Disponível em: <<http://www.iae.cta.br/laboratorios.htm>>, Acesso em: 10 Out. 2005.

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, **IEEE STD 830 Guide to Software Requirements Specifications**, IEEE, New York, 1984.

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, **IEEE STD 983 Guide for Software Quality Assurance Planning**, IEEE, New York, 1986.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION & INTERNATIONAL ELECTROTECHNICAL COMMISSION. **ISO/IEC 15504** Software Process Improvement and Capability determination Model (SPICE). Disponível em: <<http://www.iso.ch/>>, 2001, Acesso em : 06 Nov. 2002.

JACOBSON, I.; BOOCH, G.; RUMBAUCH, J., **The Unified Software Development Process**. Reading Addison-Wesley, 1999. 463p.

JAMSA, K., **C Library Bibliotecas**, tradução Eduardo Márcio de Barros Franco, São Paulo, McGraw-Hill, 1988, 274p.

KRUCHTEN, P., **From Waterfall to Iterative Life Cycle – A tough transition for project managers**, White Paper, Rational Software, 2000. Disponível em: <<http://www.rational.com/produts/%20rup/proinfo/whitepapers/>>, Acesso em: 12 Maio 2000.

KUNTZMAN, A.; KRUCHTEN, P. **The Rational Unified Process – An Enabler for Higher Process Maturity**, White Paper, Rational Software, 2003. Disponível em: <ftp://ftp.software.ibm.com/software/rational/web/whitepapers/2003/rup_tp178.pdf>, Acesso em: 20 Maio 2004.

LEMES, M. J. R. **Uma Abordagem para Concepção de Planos de Garantia da Qualidade de Software**. 1997. 115 f. Dissertação (Mestrado em Ciências). Instituto Tecnológico da Aeronáutica – ITA, São José dos Campos, 1997.

LEVINSON, L. H., POTAPCZUK, M. G., MELLOR, P. A., **Software Development Processes Applied to Computational Icing Simulation**, AIAA-99-0248, Aerospace Sciences Meeting and Exhibit, 37th, Reno, NV, Jan. 11-14, 1999.

MACMANUS, J. I., **Software Quality Assurance CASE Tolls**. In: Hadbook of Software Quality Assurance. 2^a edition, New York: Van Nostrand Reinhold, 1992. p. 225-284.

MADDEN, M. M., Measuring Reuse in a Simulation Framework, American Institute of Aeronautics and Astronautics, **Journal of Aerospace Computing**, Information and Communication, vol. 1, USA, August 2004.

MAIBOR, D.S, ANDERSON, C., DORFMAN, M. **The DOD Life Cicle Model**. In: Aerospace Software Engineering: *a collection of concepts*. Washington DC. American Institute of Aeronautics and Astronautics, 1991, p. 33-50.

MARTIN, R. C., **RUP ®/XP Guidelines: Pair Programming**, Rational Software White Paper, 2000. Disponível em: <<http://www.angelfire.com/dc/marcodorantes/xppair.pdf>>, Acesso em: 20 Maio 2004.

MILITARY STANDARD, **MIL-STD 1521B**, Technical Reviews and Audits for Systems, Equipments and Computer Software, 1995.

MILITARY STANDARD, **MIL-STD 2168**, Defence System Quality Program, Washington, D.C: U.S. Government Printing Office, 1988.

MILITARY STANDARD, **MIL-STD 498**, U.S. Department of Defence, Software Development and Documentation, Washington, DC: U.S. Government Printing Office, Dec. 1994. 67p.

MINISTÉRIO DA CIÊNCIA E TECNOLOGIA. Secretaria de Política de Informática. **Qualidade e Produtividade no Setor de Software Brasileiro**. Resultados de pesquisa bienal de 2001: Editora Brazilian Software, Brasília. nº 4, 2002. 260p.

NARAYANAN, V. K. Managing technology and innovation for competitive advantage. [S.l.]: Prentice-Hall, 2001.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION (NASA). Software Engineering Laboratory's (SEL), **SEL-81-305 Recommended Approach to Software Development, Revision 3**. June 1992. Disponível em: <<http://sel.gsfc.nasa.gov/website/documents/online-doc.htm>>, Acesso em 12 Maio 2000.

NATIONAL INSTRUMENTS, WEB Site da National Instruments, 1998, Disponível em: <<http://www.natinst.com>>, Acesso em: 17 Junho 1998.

NOGUEIRA, S. L., **Calibração de Sistemas de Medidas de Múltiplas Componentes com Interações**. 1980. 126 f. Dissertação (Mestrado em Engenharia Aeronáutica), Instituto Tecnológico de Aeronáutica - ITA, São José dos Campos, 1980.

NONAKA, I., TAKEUCHI, H. **Criação de conhecimento na empresa: como as empresas japonesas geram a dinâmica da inovação**. 4ª edição, Rio de Janeiro: Campus, 1997.

PALMA, D. P., PASTORELLI, A. L. S., TARANTI, C. G. R. **Estudo sobre os Aspectos de Desenvolvimento e de Certificação de Software de Produtos Aeroespaciais de Emprego Militar**. 2005. Instituto Tecnológico da Aeronáutica - ITA, São José dos Campos, 2005.

PASTORELLI, A. L. S. **Um Esquema para Seleção de Modelos de Processo de Desenvolvimento de Software**. 1998. 163 f., Dissertação (Mestrado em Ciências), Instituto Tecnológico de Aeronáutica, São José dos Campos, 1998.

PAULK, M. C.; CURTIS, B.; CHRISSIS, M. B.; WEBER, C.V. **Capability Maturity Model for software, version 1.1**, Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, Technical Report (CMU/SEI-93-TR-024, ADA 263403) Feb. 1993. 82p. Disponível em: <<http://www.sei.cmu.edu/pubs/documents/93.reports/pdf/93tr024.pdf>>, Acesso em: 12 Maio 2000.

PAYNE, F. M. **Low Speed Wind Tunnel Testing Facility Requirements: A customer's Perspective**. 37th Aerospace Sciences Meeting and Exhibit, January 11-14, 1999, Reno, NY, 18p.

POLLICE, G. **Using the Rational Unified Process for Small Projects: Expanding Upon eXtreme Programming**, White Paper, 2001, Disponível em: <<http://www.rational.com/produts/%20rup/proinfo/whitepapers/>>. Acesso em: 20 Maio 2004.

PRESSMAN, R. S. **Software Engineering, A Practitioner's Approach**. third edition, McGraw-Hill, 1992, 793p.

PROBASCO, L., The Ten Essentials of Development Process, White Paper, Rational Software, 2000, Disponível em: <<ftp://ftp.software.ibm.com/software/rational/web/whitepapers/2003/TP177.pdf>>, Acesso em: 20 Maio 2004.

RATIONAL CORPORATION, **Rational Unified Process – best practices for software development teams**: white paper. 2002, Disponível em: <<http://www.rational.com/produts/%20rup/proinfo/whitepapers/>>. Acesso em: 20 Maio 2004.

RATIONAL CORPORATION, **A Rational Development Process**: white paper, Rational Software, 2000. Disponível em: <<http://www.rational.com/produts/%20rup/proinfo/whitepapers/>>. Acesso em: 20 Maio 2004.

RATIONAL SOFTWARE, **Reaching CMM Levels 2 and 3 with the Rational Unified Process**, white paper, Rational Software, 2000. Disponível em: <<http://www.rational.com/produts/%20rup/proinfo/whitepapers/>>, Acesso em: 20 Maio 2004.

REIS, M. L. C. C. **Expressão da Incerteza da Medição Associada a um Ensaio Aeronáutico em Túnel de Vento Subsônico**. 2000. 103 f. Tese (Doutorado em Engenharia Mecânica), Universidade Estadual de Campinas, Outubro, 2000.

ROBINS, A.; PHILBRICK, D.; BOWEN, P.; KLABUNDE, R. **Commercial Visual Programming Environments: One Step closer to Real Simulation Reuse**, AIAA-1998-4528, Collection of Technical Papers (A98-37288 10-01), AIAA Modeling and Simulation Technologies Conference and Exhibit, Boston, MA, Aug. 10-12, 1998.

ROETZHEIN, W. H. **Developing Software to Government Standards**. New Jersey: Prentice Hall, 1991.478p.

RADIO TECHNICAL COMMISSION FOR AERONAUTICS, **RTCA/DO-178B, Software Considerations in Airborne Systems and Equipment Certification**, Washington, RTCA, December 1992.

SHIVANANDA, T. P. ZABRENSKY, E. F., MCKEEL S. A., PAPAY, M. L. **Comparison of engineering and CFD predictions and wind tunnel data for a launch vehicle configuration**, AIAA-1997-2251, AIAA Applied Aerodynamics Conference, 15 th, Atlanta, GA, june 23-25, 1997, Technical Papers, PT 1 (A97-31695 08-02), 1997.

SHNEIDERMAN, B. **Designing the user Interface: Strategies for Effective Human-Computer Interaction**, 3nd edition, 1998, Addison-Wesley Publishers, Reading, MA, 1998, 672p.

SOARES, M. S. **Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software**, Disponível em: <<http://controlchaos.com/about/xp.php/>>, Acesso em: 20 Maio 2005.

SOARES, M. S. **Metodologias ágeis Extreme Programming e Scrum para o Desenvolvimento de Software**. Disponível em: <<http://controlchaos.com/about/xp.php/>>, Acesso em: 20 Maio 2004.

SOMMERVILLE, I. **Engenharia de Software**, 6^a edição, tradução de André Maurício de Andrade Ribeiro, Editora Addison-Wesley, 2003, 592p.

TAKAHASHI, T., LIESENBER, G. H. K. E., XAVIER, D. T. **Programação Orientada a Objetos**, VII Escola de Computação, São Paulo, 1990, 335p.

WILLIAMS, L. G., **Software Reuse: Definition and Overview**, In: Aerospace Software Engineering: *a collection of concepts*. Washington DC. American Institute of Aeronautics and Astronautics. p. 395-413, 1991.

YOURDON, E. **Análise Estruturada Moderna**, Editora Campus, Rio de Janeiro, 1992, 836p.

ZARRELLA, P. **CASE Tool Integration and Standardization**, Technical Report CMU/SEI-90-TR-14, Software Engineering Institute, Carnegie Mellon University, Pittsburg, PA, December, 1990. Disponível em: <<http://www.sei.cmu.edu/pubs/documents/90.reports/pdf/90tr014.pdf>>, Acesso em: 12 Maio 2000.

APÊNDICE A - PRINCIPAIS NORMAS DE SOFTWARE.

Na tabela A1 a seguir são apresentadas as principais normas utilizadas para projeto de sistemas de software. Algumas dessas normas ou padrões já se encontram descontinuadas ou substituídas, no entanto, muitos projetos nacionais ainda vem sendo desenvolvidos segundo os requisitos apresentados por elas.

Tabela A1 - Principais normas de software

Aplicação	Normas de Software
Processo / Ciclo de Vida	DOD-STD-2167A
	MIL-STD-498
	SPICE ou ISO 15504
	ISO/IEC 12207
Qualidade de Software como Produto	DOD-STD-2168
	ISO 9126 ou NBR13596
	ISO 12119 (p/ software COTS)
	ISO 14598 (em conjunto com a ISO 9126)
Certificação de Sistemas de Software	RTCA/DO-178B
Gerenciamento da Configuração	MIL-STD-973
	MIL-STD-1521B
Sistema da Qualidade de Software	ISO 9000-3
	SAE AS 9106
	AQAP 150
	AQAP 159
	AQAP 160

As diferenças entre os documentos das normas DOD-STD-7935A (1998), DOD-STD-2167A (1988) e MIL-STD-498 (1994) podem ser observadas na tabela A2 a seguir:

Tabela A2 - Diferenças entre os documentos das normas de software.

DOD-STD-7935A	DOD-STD-2167A	MIL-STD-498
Functional Description (FD)	Software Development Plan	Software Development Plan
Implementation Procedures	—	Software Installation Plan
Maintenance Manual	Computer Resources Integrated Support Document	Software Transition Plan
Functional Description	System/Segment Design Document	Operational Concept Description
System/Subsystem Spec Functional Description (FD)	System/Segment Specification	System/Subsystem Specification
System/Subsystem Spec	System/Segment Design Document	System/Subsystem Design Description
Software Unit Specification	Software Requirements Specification	Software Requirements Specification
Software Unit Specification	Interface Requirements Specification	Interface Requirements Specification
Software Unit Specification Maintenance Manual	Software Design Document	Software Design Description
Software Unit Specification	Interface Design Document	Interface Design Description
Database Specification	--	Database Design Description
Test Plan	Software Test Plan	Software Test Plan
Test Plan	Software Test Description	Software Test Description
Test Analysis Report	Software Test Report	Software Test Report

DOD-STD-7935A	DOD-STD-2167A	MIL-STD-498
Manual Maintenance	Computer Resources Integrated Support Document Software Product Specification	Software Product Specification
--	Version Description Document	Software Version Description
End User Manual	Software User's manual	Software User Manual
Computer Operation Manual	--	Software Center Operator Manual
Users Manual	--	Software Input/Output Manual
--	Computer System Operator's Manual	Computer Operation Manual
--	Software Programmer's Manual	Computer Programming Manual
--	Firmware Support Manual	Firmware Support Manual

As normas apresentadas na tabela A2 foram evoluídas e podem ser observadas pelos acréscimos ou ajustes de documentações de uma para outra. Existe atualmente outra evolução destas normas que é a NBR ISO/IEC 12207 (1998), a qual durante o desenvolvimento deste trabalho se observou a imaturidade dos processos de engenharia de software utilizados no TA-2, para adoção de tal norma para desenvolvimento dos subprodutos.

APÊNDICE B – MATERIAIS DO SISTEMA DE MEDIDAS DO TA-2

B.1 CARACTERÍSTICAS FÍSICAS DO TA-2

O Túnel de Vento nº 2, mais conhecido como TA-2, é um túnel de vento subsônico, do tipo contínuo, de circuito fechado, funciona com pressão e temperatura ambientes. Possui uma seção de ensaios fechada e de baixa turbulência. A estrutura do TA-2 é construída em concreto armado, com uma Seção de Ensaio em chapa de aço, que possui uma área de $6,3 \text{ m}^2$, em forma geométrica retangular. As velocidades típicas na Seção de ensaios são:

- Máxima sem bloqueio: 500km/h (139 m/s).
- Mínima: 5 km/h (1,4 m/s).

As características típicas da Seção de Ensaios são:

- Seção transversal de 3,20 m por 2,10 m.
- Máximo número de Mach: 0,25.
- Potência do Motor de 1600 HP.
- Entradas e Saídas de sinais: na faixa de 0 a 5V.
- Nível de ruído: Baixo, inferior a 20 mV.
- Alimentação elétrica: $\pm 12\text{V Gnd DC}$ ou 220V AC.
- Perturbação elétrica: em degrau via chave on/off.

A velocidade do vento deve ser controlada na execução do ensaio do modelo aerodinâmico. O ensaio no túnel é feito com uso de um modelo aerodinâmico em escala (1:10 e 1:8) instalado em um ou dois suportes. Estes suportes são montados sob uma balança externa de 06 componentes com sensores instalados para medir forças e momentos, existindo também outros sensores para medir temperatura, pressão e outros fenômenos físicos. No suporte estão instalados atuadores para mudança de ângulos em pitch e yaw. Os dados e resultados dos testes são obtidos e armazenados através dos softwares de aquisição e redução de dados

Materiais usados no sistema de medidas do TA-2 com a plataforma HP são os seguintes:

- 1- Computador HP9000/300;
- 2- Multímetro HP3956A;
- 3- *Scanner* HP3947A;
- 4- Impressora HP2932A;
- 5- Condicionador de sinal Honeywell da Accodata;
- 6- Filtro Analógico feito nas dependências da Boeing para o TA-2;
- 7- Cabo de comunicação HP-IB.

Os softwares em linguagem BASIC da plataforma HP estudados neste trabalho são apresentados a seguir:

1. Software de Redução de Dados NACA0012/ALFA/SCANIVALVE,
“NACA001299/REDNACA12SC”, de março de 1999;
2. Software de Aquisição de Dados em ALFA/BIDIMENSIONAL,
“EMB17099BD/AQ170BA”, de abril de 1999;
3. Software de Aquisição de Dados em ALFA/BETA,
“EMB991701C/AQU1701MC”, de julho de 1999;
4. Software de Redução de Dados em ALFA/BETA,
“EMB9917001C/RED1701OS”, de junho de 1999;
5. Software de Redução de Dados em ALFA/BETA,
“EMB991701C/RED170FS2”, de agosto de 1999;
6. Software de Redução de Dados em ALFA/BETA,
“EMB991701C/RED1701MC”, de julho de 1999;

7. Software de Aquisição de Calibração da Balança T & C – TA2,
“PROGRAMASL/CALBALTA2/ACALBALTA2”, de março de 2000;
8. Software de Cálculo de Cargas para Calibração da Balança T & C, “CALCAR”,
julho de 1986;
9. Software de Cálculo de Cargas para Calibração da Balança T & C,
CALBALTA2/CARGATA2”, sem data;
10. Software de Calibração de Sesor de Pressão Dinâmica”,
CALSENSASL/CALSENSORQ”, sem data;
11. Software de Calibração de Pressão Dinâmica X Referência (pitot centro X
pitot AMX), ‘CALSENSASL/PRDINREF”, sem data
12. Software de Aquisição de Medida PSI.ESP.32HD, “PSIESP32HD/AQUPSI”,
de outubro de 2000;
13. Software de Zeragem da Balança TA-2, “CALBALTA2/ZERAGEM”, sem
data;
14. Software para Calcular a Matriz Alfa, “CALBALTA2/MATRIZALFA”, de
setembro de 1999;
15. Software para Calcular a Matriz Beta, “CALBALTA2/MATRIZBETA”, sem
data;
16. Software para Calcular a Matriz K (Coeficientes de Sensores),
“CALBALTA2/CALMATK”, sem data;

17. Software que Fornece Resultados de Carregamentos ,
“CALBALTA2/CALGRAU1”, de novembro de 1990;
18. Calibração da Balança Combinado – 12 carregamentos,
CALBATA2/CALBALRAP, de setembro de 1989;
19. Software de Cálculo de Ganho – Regressão Linear, Quadrática e Cúbica,
“AUXILIAR/NOVOGANHO”, sem data;
20. Software de Calibração da Balança Combinado 12 Carregamentos,
“AUXILIAR/TESTECAL”, sem data.

Como podem ser observados existem vários softwares para aquisição e redução de dados relacionados anteriormente, no entanto, como a cada manutenção estes softwares eram alterados devido aos requisitos do cliente, estes foram também estudados para confecção deste trabalho.

B.2 MATERIAIS PARA CALIBRAÇÕES E MEDIDAS

Para as calibrações e medidas de pressão, de temperatura e de velocidade de escoamento são utilizados os seguintes instrumentos:

- Transdutores de pressão: Conjunto de sensores de pressão diferencial com faixas de $\pm 0,60$ Pa (0,6 mbar) a 17236,88 Pa (2,5 psi), para medir pressões estática, dinâmica e total (estática + dinâmica), são sensores da marca Druck, Gold Statham e Ashcroft.
- Scanivalves: Conjunto de 08 scanivalves com 48 tomadas de pressão, modelo 48D3, com sensores PDCR23D, para medir pressões advindas do modelo aerodinâmico.

- Bancos de transdutores de pressão: 11 escaneadores de pressão diferencial com faixas de 2490,89 Pa (10 polegadas) de coluna de água a 17236,88 Pa (2,5 psi), com 32 canais cada escaneador.
- Padrões de transferência Druck para calibração de sistema em instalação final, faixas de 1723,69 Pa (0,25 psi), 6894,75 Pa (1 psi) e 17236,88 Pa (2,5 psi), com precisão de 0,01%.
- Coluna líquida: marca airflow de 4903,33 Pa (500 mmH₂O) e 58,84 Pa (6 mmH₂O).
- Balança de peso morto para calibração de pressão DANTEC, com faixas de 50000 Pa (500 mbar) e 10000 Pa (100 mbar), com incerteza de 200 ppm.
- Termômetros de resistência: marca PT-100 condicionados, 13 unidades para monitoração do motor e 3 unidades para monitoração do escoamento e ambiente de equipamentos.
- Tubo de pitot: fabricado pela Airflow Developments Limited, sem modelo.
- Anemômetro a fio quente: marca DANTEC, com pontas de prova a 1, 2 ou 3 eixos.

B.3 MATERIAIS E PROCESSOS DE CALIBRAÇÃO PARA PLATAFORMA PC

Um processo para a calibração de um sensor de pressão deve ser usado, para verificar os valores das constantes de cada um dos sensores dos diversos ensaios. Essas constantes são comparadas com as calibrações dos fabricantes e devem ser armazenadas, caso sejam equivalentes, ou simplesmente usadas as do fabricante para os ensaios em túnel, que são inseridas nas respectivas variáveis do software de aquisição de dados.

Um outro processo de calibração só que com carregamentos de cargas é feito para as células de carga, no laboratório de força, também para obtenção das constantes de calibração de cada célula de carga, que serão instaladas na balança correspondentes às forças e aos momentos. Neste caso, o processo é repetido na cadeia de medidas, para 73 carregamentos em alfa (subida ou descida) e 219 em beta (esquerda ou direita), com as taxas de aquisições de dados em *milivolts*, a qual se determinou experimentalmente para o modelo aerodinâmico específico.

Para o processo de calibração das células de carga implementado no laboratório de força na primeira iteração da construção de software, foram usados alguns instrumentos, dispositivos eletrônicos e *drivers*, os quais foram definidos na primeira iteração da elaboração, para a criação do subproduto de software de aquisição de dados para calibração de células de carga. Estes recursos para as calibrações e medidas de forças e momentos do laboratório de força do TA-2, planejados para a primeira construção, são os seguintes:

- Células de carga: marcas BLH, HBM, com faixas de 20 kgf a 500 kgf.
- Máquina de calibração de células de carga para tração e compressão, para carga máxima de 500 kgf e conjunto de pesos padrão de aço inox.
- Multímetro digital marca HP, modelo HP3958A.
- Placa de aquisição de dados da *National Instruments* de interface IEEE 488 - AT-GPIB/TNT (*Plug and Play*), com configuração automática de endereço de entrada e saída, nível de interrupção e canais *Direct Memory Access* – DMA.
- *Driver* de Software NI-488.2, para *Windows 3.1/9x/Me/NT*.
- Cabo da *National Instruments* GPIB com 2 metros.
- Computador pessoal.
- Aplicativo *LabView* versão 3.1, para *Windows 3.1/ 95*.

A provisão de materiais para o novo sistema de medidas constam das tabelas B.1 a B.5, correspondentes a segunda, terceira e quarta iteração da fase de construção, com ações do processo unificado via METAPLI.

Tabela B.1 - Provisão de dispositivos da 2ª e 3ª iteração de construção.

Descrição Módulo/Placa	Aplicação	Característica
AT-MIO-64F-5 Placa de aquisição analógica	Aquisição de dados diversos	64 entradas e saídas analógicas de 12 bits, de 500 kS/s de amostragem por canal
PC-DIO-96 Placa de aquisição digital	Controles diversos programados	
SCXI-1001 Gabinete de CPU	Acoplar os módulos ou placas DAQ	12 slots
SCXI-1121 Condicionador de Sinais	Condicionar sinais de diversos sensores	4 canais diferenciais, range de $\pm 5V$ com ganho selecionado através de <i>jumper</i> , filtro tipo 3 pólos RC, frequência de corte de 4Hz ou 10 KHz, selecionado através de <i>jumper</i> , erro de offset de $\pm 6\mu V \pm 3$ mV/ganho.
SCXI-1181 Módulo de Alimentação	Criar o circuito otimizado no sistema SCXI	Controla sistema em um slot SCXI
SCXI-1160 Módulo de relês	Ativação dos dispositivos para controlar os sinais de baixo nível para ângulos, vibrador	De 0 a 5 V
AT-GPIB/TNT Placa de aquisição de dados	Controle e medição de instrumentos GPIB	padrão IEEE 488.2

Tabela B.2 - Provisão de dispositivos da 4ª iteração de construção.

Descrição Módulo/Placa	Aplicação	Característica
PXI-1045 Gabinete Plataforma compactPCI (PXI)	Uso misto para entrada e saída de dados com controlador interno ao gabinete	18 slots
SCXI-1001 Gabinete de CPU	Acoplar os módulos ou placas DAQ	12 slots
NI PXI 8186RT Microcomputador com processador Intel Pentium 4 de 2,2 Ghz com <i>LabView Real Time</i>	Gerencia todos os módulos acoplados através de barramento existente no gabinete	Controlador interno ao gabinete. Recursos adicionais de interfaces e sinais de trigger
NI PXI 8231 Controlador Ethernet GBIT	Para uso futuro	Para Windows
NI PXI 6071E Placa multifunção de aquisição de dados NI-DAQ	Controle dos módulos SCXI, para entrada analógica	Conversor A/D, conversão de 12 bits e 64 canais de entrada analógica, velocidade de varredura de 1,25 M S/s, de $\pm 10V$
NI PXI 6704 Conversor digital analógico	Reserva para implementações futuras em relação ao controle de velocidade do motor principal	16 canais de saída, para faixa de 0 a 10V

Tabela B.3 - Provisão de dispositivos da 4ª iteração de construção (continuação).

Descrição Módulo/Placa	Aplicação	Característica
NI PXI 4070 Voltímetro de 6½ dígitos	Capacidade de digitalizar em alta voltagem para leitura de sinais em Volts DC ou AC da vibração do mancal do motor e 6 componentes da balança em baixo nível usada com SCXI-1127	Adquirir gráficos usando funções do LabView para analisar estes gráficos no domínio do tempo e da frequência, de $\pm 10V$ e de $\pm 30 mV$
NI PXI 6602 Módulo Temporizador Contador e NI-DAQ	Para controle de zeragem do mastro, através de driver de controle para acionamento dos motores de passo e também do reostato líquido com controle através de driver apropriado	cartão com 8 canais configuráveis, anível TTL de 300 Hz
NI PXI 7358 Controlador de movimentos	Para o controle dos motores de passo para movimentação dos ângulos α e β do mastro	Controle de 8 eixos simultaneamente
NI PXI 6052E Placa multifunção	Aquisição DAQ de entrada analógica para leitura de pressão do módulo multiplexador (cartão 1) e de 6 componentes da balança e da pressão dinâmica (cartão 2).	Conversor de 16 bits e 16 canais de entrada analógica, para DC de $\pm 10V$ e de 0 a 5V, 30 mV.
SCXI-1102B Amplificador de Leitura Analógica	Monitoramento dos parâmetros do motor que gera o vento e a subestação do túnel	de 32 canais , para sensor de 0 a 10 V, e Banda de 200 Hz

Tabela B.3 - Provisão de dispositivos da 4ª iteração de construção (continuação).

Descrição Módulo/Placa	Aplicação	Característica
SCXI-1160 Módulo de relês	Ativação dos dispositivos para controlar os sinais de baixo nível para ângulos, vibrador, subestação, motor de passo e sentido de rotação 6 componentes	De 0 a 5 V
SCXI 1180 Pannel	Amplia os sinais de entrada e saída de dispositivos DAQ para o painel frontal do chassis SCXI	Ocupa um slot do sistema SCXI e é também usado para rotear os sinais de dispositivos DAQ separados.
SCXI 1127 Multiplexador de alta voltagem	Para uso associado ao PXI-4070, de sinais analógicos especiais de vibração do mancal e 6 componentes da balança em baixo nível	de 64 canais, do tipo VRMS e DC de ± 30 mV
SCXI 1581 Condicionador de sinais	Leituras dos parâmetros de temperatura de enrolamento do motor que gera o vento	de 32 canais, para sensor PT100 e temperatura RTD de 0 a 100°C
SCXI 1162HV Módulo de entrada digital de alta voltagem isolada	Indicação de anomalias do motor e da porta da seção de ensaio	de 32 canais e níveis de sinais de 24V DC e 5V TTL.

Tabela B.3 - Provisão de dispositivos da 4ª iteração de construção (continuação).

Descrição Módulo/Placa	Aplicação	Característica
SCXI 1163R Cartão com relês de estado sólido	Determinação do sentido de rotação e a velocidade do motor de passo que controla o movimento de subida e descida do reostato líquido	32 relês, corrente 100mA e tensão de 24V
SCXI 1353 Cabo Assembly Blindado	Usado com cabo SH1006868 e dois adaptadores de cabo SCXI de 50 pinos cada. O 1º adaptador é idêntico ao SCXI-1359 enquanto o 2º roteia os sinais digitais e analógicos ampliados para módulos digitais SCXI.	de 2m, para conectores de 100 pinos de placas séries E para chassis SCXI.
SCXI 1357 Cabo Assembly Blindado	Conectar os dispositivos NI PXI 4070 para um chassis SCXI	Conexão
SCXI 1300 Bloco de Terminal de Propósito Geral	Conectam sinais de entrada aos módulos SCXI-1100 e SCXI-1181	Possui sensor de temperatura a bordo
SCXI 1302 Bloco de Terminal	Para uso com SCXI 1180	50 ligações
SCXI 1331 Bloco de terminal	Para uso com SCXI 1127	de 64 canais com CJC
SCXI 1326 Bloco de terminal	Alta voltagem para módulos de séries SCXI-1162HV	48 terminais de ligação
SCXI 1324 Bloco de Terminal	Alta voltagem para SCXI-1160	48 terminais de ligação

Tabela B.3 - Provisão de dispositivos da 4ª iteração de construção (continuação).

Descrição Módulo/Placa	Aplicação	Característica
NI PXI 6509 Módulo de I/O de sinais digitais	Usado para entradas de ângulo do encoder	Para TTL
NI PXI 6528 Módulo de I/O de sinais digitais	Usado para as saídas de pressão multiplexadas e entradas de zeragem das 6 componentes	+ 12V, +24V
NI PXI 8423/4 Módulo de I/O serial	Módulo reserva para uso futuro	Possui 4 portas seriais RS-485, transferência de 460 Kb/s e isolamento de 2 kV
SCXI 1520 módulo condicionador de sinal	Módulo de reserva para condicionar entrada analógica, composto de 4 cartões	Do tipo DC para 30 mV
SCXI 1100B Módulo para leitura analógica amplificada	Para leitura de alto nível dos sensores de pressão estática, barométrica, temperatura, umidade e pressão dinâmica	De 0 a 10V e de 0 a 5 V
SCXI 1349 Cabo Assembly Blindado	Conectar os dispositivos NI PXI 6052E para um chassis SCXI	Conectores de 68 pinos
SCXI 1314 Bloco de Terminal	De montagem frontal para módulo SCXI-1520	Resistores de 350 Ω para cada canal Resistores de 120 Ω para strain gauge
SCB 100 Bloco conector de I/O blindado	Para saída analógica do módulo NI PXI-6509 e 6528	conexão

Tabela B.3 - Provisão de dispositivos da 4ª iteração de construção (continuação).

Descrição Módulo/Placa	Aplicação	Característica
SH100-100-F Cabo assembly	Para saída analógica, trabalha com dispositivos NI PXI 6528	Conexão de 2 m
SH68-68-EP Cabo I/O blindado	Para saída analógica, trabalha com dispositivos NI PXI-6052E	68 condutores com 2 conectores fêmea
SH68-C68-S	Para saída analógica, trabalha com dispositivos NI PXI 7358	conexão
SCB-68 Bloco de conector de I/O blindado	Para saída analógica do módulo NI PXI-6704 e 6602	conexão
SH68-68-D1 Cabo de I/O blindado	Para uso com NI PXI-6704 de dispositivos de saída analógico e NI PXI-6602 de dispositivos de I/O digitais	Similar ao SH-68-68-EP, para conexão de 2 m.

Em todos os ensaios em túnel, os dados da pressão dinâmica devem ser obtidos e armazenados durante o ensaio, assim como a umidade relativa e a temperatura, sendo que os valores em *milivolts* relacionados a cada grandeza física deverão ser obtidos a partir de testes experimentais para validar as taxas de aquisições de dados.

Além dessas provisões de dispositivos definidos na segunda iteração da fase de elaboração, outros mais são previstos, que são:

- Software *Driver NI-DAQ* para *Windows 95/98/NT*, com variedades de funções de entrada/saída para sinais analógicos e digitais, que trabalha com tipos diferentes de controladores para transferência de dados por *Direct Memory Access* -DMA e interrupções.
- Software NI-DAQ para *Windows 95/98/NT*.
- Pacotes de Software *Driver NI-488.2* para *Windows 95/98/NT*, permitindo a mudança de configuração de hardware com interface GPIB, para plataforma ISA.
- Driver de níveis de corrente e tensão para acionamento do mastro.
- UMI 7764: encoders de 20 MHz.

O sistema de aquisição de dados de ensaio foi especificado para adquirir, por exemplo, até 40 canais analógicos de parâmetros condicionados com conversão de 16bits, 333 kilosamples / segundo ou 6½ para medidas até 30mV, 48 entradas digitais isoladas para *encoders* dos ângulos do modelo, entradas/saídas a relé ou isoladas para comando de atuadores para tara de balança, sensoriamientos de seus fins de curso e enderçamento de escaneadores de pressão para até 352 canais.

Para execução de ensaios estão disponíveis também, alguns equipamentos adicionais como:

- Sistema de anemometria de fio quente com calibrador de velocidade;
- Equipamento para medida de velocidade de vento e turbulência em 3 eixos;
- Acelerômetros em faixas diversas de $\pm 0,1$ g a ± 50 g;
- Multímetro digital;
- Analisador de espectro de 2 canais; e
- Osciloscópio digital.

Estão ainda disponíveis para o ensaio em túnel de vento:

- Uma balança padrão de peso morto e padrão de transferência para rastreamento das medidas de pressão do sistema.
- Uma base para calibração de células de carga.
- Balanças digitais e conjunto de massa padrão para rastreamento das medidas de força do sistema.

O túnel de vento também possui equipamentos de visualização de escoamento como:

- Gerador de fumaça.
- Sistemas de visualização de escoamento e anemometria a laser PIV e LDA, além de câmeras de TV em miniaturas e gravador de vídeo SVHS.

B.4 FLUXOGRAMA DE ENSAIOS DO TA-2

Todos estes materiais e processos são empregados segundo a modelagem feita por fluxograma, que representa as atividades de um ensaio aerodinâmico, no qual cada condição ou declaração é seguida conforme necessidade planejada para o ensaio aerodinâmico específico. Este fluxograma possibilita o entendimento do grupo referente ao processo completo de um ensaio aerodinâmico, e é apresentado nas figuras B.1 e B.2. Nele estão definidos e identificados os fluxos planejados para os ensaios aerodinâmicos, que abrangem todas as atividades de um programa de ensaios do túnel de vento do CTA.

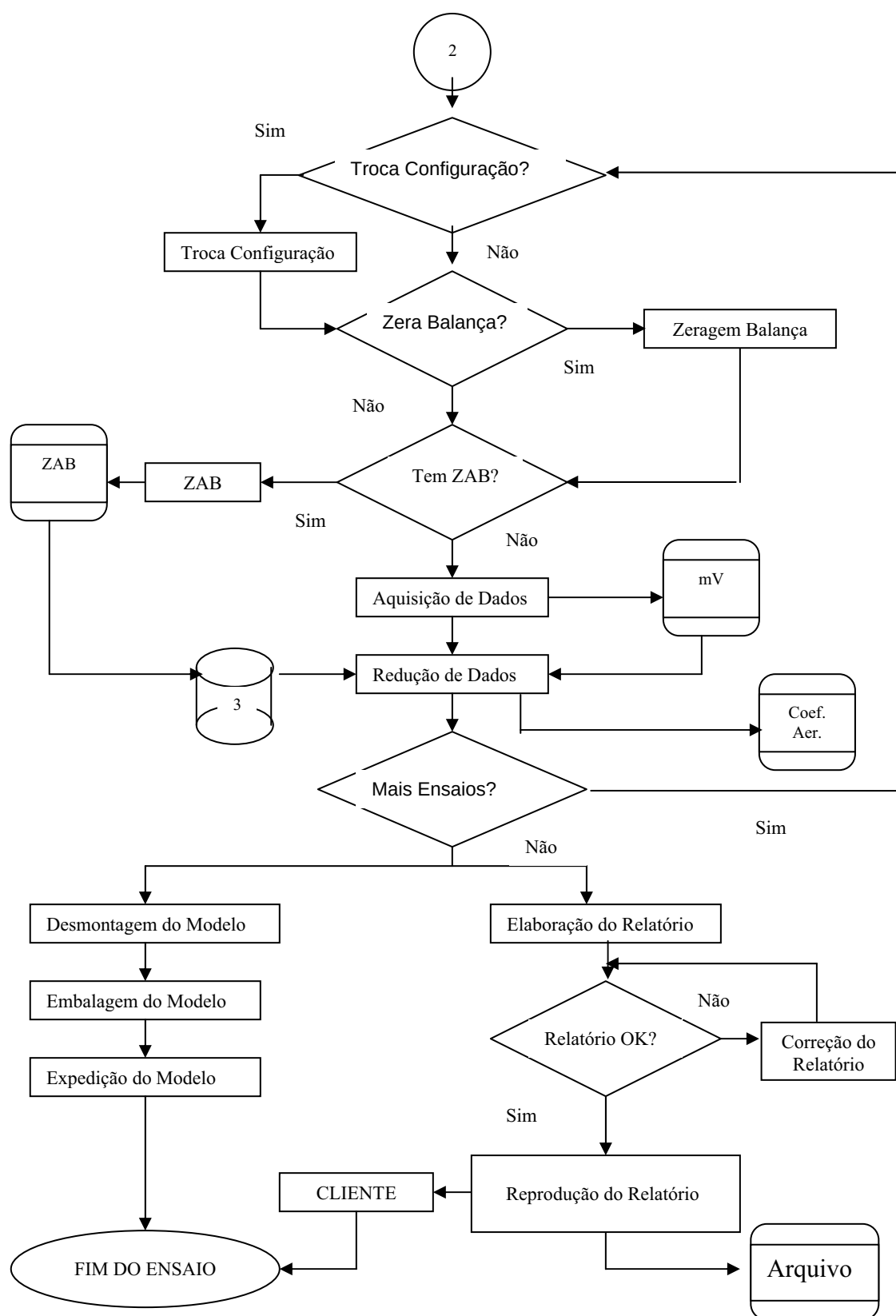


Figura B.2 - Continuação do fluxograma de ensaios aerodinâmico do TA-2

Estas atividades serviram de base para a seleção das práticas de cada ensaio em túnel, e auxiliou na decisão das atribuições das prioridades de modernização dos subsistemas envolvidos. Trata-se de atividades como calibrações e aquisições necessárias para a determinação dos coeficientes aerodinâmicos e também para a avaliação do projeto aerodinâmico em estudo.

APÊNDICE C - GLOSSÁRIO DE TERMOS

Abstração. Ato de criar ou ter de uma visão geral de um modelo que contenha características essenciais ou propriedades de partes ou de produtos de software, que atenda o cumprimento parcial ou completo de projetos de sistemas de softwares.

Ambiente de desenvolvimento. Uma instância ampla contendo configuração de hardware e software elaborada com o objetivo de instalar e executar o software durante o ciclo de desenvolvimento.

Ambiente de teste. Uma instância específica dentro do ambiente de desenvolvimento com uma configuração de hardware e software criada com a finalidade de realizar testes em condições conhecidas e controladas.

Análise e projeto. As ações do processo unificado implantada via METAPLI, para mostrar como os subprodutos do sistema de software serão realizados na implementação. Trata-se das atividades gerais implantadas durante o ciclo de desenvolvimento, as quais decisões gerenciais e técnicas são tomadas para atender aos requisitos funcionais e não funcionais, visando a qualidade de um subproduto do sistema.

Analista de Sistema. Papel responsável pela coordenação e obtenção dos requisitos de software e levantamento das funcionalidades do sistema de medidas e seus subsistemas.

Aprovação ou Aceitação. O ato ou instância de expressar uma opinião favorável ou conceder autorização oficial ou formal de uso de partes ou todo software desenvolvido.

Arquiteto de Instrumentação. Papel responsável pela seleção e controle de instrumentação aplicada aos ensaios, definição de interfaces de comunicação e protocolos, configuração dos dispositivos eletrônicos do ambiente de ensaio. É responsável por coordenar as atividades de arquitetura de hardware para um ensaio aerodinâmico.

Arquiteto de Software. Papel responsável por coordenar atividades de geração de arquitetura dos subprodutos e do produto final de software do sistema de medidas.

Arquitetura. Um projeto de alto nível que provê decisões feitas sobre os problemas que um componente irá resolver, podendo se tratar de componente de software ou de hardware, descrições de componentes, relações entre os componentes e descrição dinâmica de operação.

Artefato. Uma parte da informação que é produzida, modificada ou usada pelo processo, define uma área de responsabilidade e está sujeito a controle de versão. Um artefato pode ser um modelo, um elemento do modelo ou um documento.

Ator. Pessoa que assume, um dado momento, dentro do ambiente de desenvolvimento de software, uma função específica. Portanto ela pode assumir vários papéis simultâneos ou não durante o ciclo de desenvolvimento.

Baseline. Uma configuração aprovada e registrada de um ou mais itens de configuração (subprodutos), que serve como base para favorecer o desenvolvimento, que é mudado somente através de procedimentos de mudança.

Caso de desenvolvimento. O processo de engenharia de software usado pela organização realizadora, o qual é desenvolvido como uma configuração ou uma personalização para obtenção do produto, com processos adaptados às necessidades do projeto.

Caso de uso. Uma seqüência de ações de desempenhos de um sistema que produz um resultado observável de valores para um ator particular. Um caso de uso contém todos os fluxos principais, alternativos e exceções dos eventos relacionados com a produção de valores de resultados observáveis.

Caso de teste. A definição geralmente formal de um conjunto específico de teste com entradas, condições de execução e resultados esperados, identificados para avaliar um determinado aspecto de partes de um subproduto de software, o subproduto, subsistemas ou o sistema completo. É uma especificação do teste formada de maneira mais completa para testar um sistema de software.

Ciclo de desenvolvimento. Conjunto de fases do processo de desenvolvimento, no caso do processo unificado pelas fases de iniciação, elaboração, construção e transição e no processo tradicional as fases ou processos de requisitos, análise, projeto preliminar, projeto detalhado, implementação, teste e entrega.

Ciclo de testes. Um período de atividades que pode durar ou não uma fase dentro do ciclo de desenvolvimento, com objetivo de avaliar, testar e coletar dados que comprovem as funcionalidades e iterações de sub sistemas do projeto. A certificação ou aprovação dos sistemas de software tem como base os resultados apresentados nesta fase.

Código ou Programa. A implementação de dados particulares ou programa de computador particular na forma simbólica, como um código fonte, código objeto ou código de máquina.

Compilador. Programa que traduz declarações de código fonte de linguagem de alto nível, como BASIC, FORTRAN, C em código objeto.

Componente ou Subproduto. Uma parte auto contida, combinações de partes, sub montagens ou unidades, que desempenham uma função distinta de um sistema ou subsistema.

Controle de Mudança. O processo de registro, avaliação, aprovação ou desaprovação e coordenação de mudanças de itens de configuração, após o estabelecimento formal de suas identificações de configuração ou baselines.

Depósito de Dados. Um conjunto de dados, parte ou todo de qualquer conjunto de dados, consistindo de uma menor parcela de um arquivo, o qual é suficiente para aplicar ou conhecer o sistema de processamento de dados.

Desenvolvedor. É um agente coletivo responsável por criar um plano de construção de software que esteja dentro das condições estabelecidas pelo cliente. Uma organização ou grupo de pessoas responsável pelo desenvolvimento das funcionalidades necessárias para o projeto do sistema de software.

Desenvolvimento de Software. Processos ou atividades que abrangem a condução do projeto de um produto ou subproduto de software até um estado em que esteja pronto para entrar em fase operacional.

Diagrama. Uma representação gráfica parcial ou total de uma modelagem ou de uma coleção de elementos de modelo, geralmente processados como um gráfico de elementos de modelo conectados.

Diagrama de Contexto. Uma representação gráfica da visão de um conjunto de elementos de modelagem, relacionados e voltados para uma determinada finalidade como, por exemplo, a especificação de uma operação do túnel de vento.

Diagrama de Estado. Um diagrama que mostra uma máquina de estado. Consulte máquina de estado.

Dispositivo Eletrônico. Conjunto de dispositivos de engenharia (placas de aquisição, placas condicionadoras, entre outros), instalados ou interligados em uma determinada plataforma para a aquisição de parâmetros do sistema avaliado.

Documentação. Atividade de descrever e registrar informações produzidas pelos processos ou atividades do ciclo de vida de desenvolvimento de software.

Documento. Um documento é uma coleção de informações que são representadas em papel ou em uma mídia que representa um papel. Essa representação de papel inclui o conceito de páginas e tem uma seqüência de conteúdo explícita ou implícita. As informações estão em texto ou em imagens bidimensionais. São exemplos de representação de papel: documentos de processadores de texto, planilhas, agendas, páginas da Web e apresentações de slide.

Documento de Requisitos de Produto. Uma descrição de alto nível do produto (sistema), seu uso pretendido e o conjunto de recursos que ele oferece.

Elaboração. É a segunda fase do processo unificado em que a visão do produto e sua arquitetura são definidas.

Elaboração do Modelo. O processo de geração de um tipo de um repositório a partir de um modelo publicado. Inclui a geração de interfaces e implementações, que permite que os repositórios sejam instanciados e preenchidos de acordo e com base no modelo elaborado.

Engenharia Reversa. O método de extrair informação do projeto de software de um código fonte. Dentro de um contexto filosófico é o processo em que um produto tem suas partes criteriosamente analisadas e estudadas para entender as funcionalidades e interações, compondo um conjunto de conhecimentos que possa ser utilizada para obter o produto ou outro semelhante.

Engenheiro de Software. Papel responsável por coordenar e executar atividades referentes ao projeto de software. É responsável por propor a abrangência dos processos, atividades e modelagens gerenciais e técnicas para o projeto do sistema de software.

Entrega. O ato de formalidade marcando a autorização e liberação para uso do item de configuração obtido.

Especialista em Ferramentas e Ambiente. Papel responsável pelo suporte ao desenvolvimento na utilização e instalação dos hardwares, softwares, ambientes e ferramentas de apoio.

Especialista em Interface. Papel responsável pela coordenação e construção de interface de dados e prover a apresentação desses dados ao usuário, gerando os artefatos pertinentes.

Especificação. Uma descrição declarativa do que algo é ou faz.

Especificação de Requisitos de Software (SRS). Um conjunto de requisitos que definem completamente o comportamento externo do sistema a ser criado - às vezes é denominada especificação funcional.

Fase. O tempo entre dois marcos primários do projeto, durante o qual um conjunto bem definido de objetivos é atendido, artefatos são concluídos e decisões são tomadas sobre passar ou não para a próxima fase.

Ferramenta de Programação Visual. Uma ferramenta que fornece meios para especificar programas graficamente. Neste caso através de ícones ou representações gráficas colocados no editor de programação em que cada um representa uma função e ou ação do aplicativo em desenvolvimento.

Fluxo de Trabalho. Uma representação gráfica do agrupamento de atividades realizadas em colaboração com os parceiros, para alcançar algum resultado no desenvolvimento de sistema de software. As atividades geralmente são realizadas em paralelo ou iterativamente, com a saída de uma atividade servindo de entrada para outra atividade. Os fluxos de trabalho são usados para agrupar atividades a fim de fornecer um nível mais alto de abstração e aumentar a abrangência das atividades de trabalho.

Fluxograma. É uma representação diagramática resumida das entradas e saídas de um processo que passam pelas atividades específicas com as práticas e interações entre elas.

Garantia da Qualidade. Atividades para fornecer a garantia adequada de que os processos e subprodutos de software, no ciclo de vida do projeto, estejam em conformidade com seus requisitos especificados e sejam aderentes aos planos estabelecidos.

Gerência de Configuração. Atividades para aplicação de procedimentos gerenciais durante todo o ciclo de vida de software, destinados a identificar e definir partes dos subprodutos de software que compõem um sistema, estabelecendo suas linhas básicas (baseline) de composição, o controle das modificações e liberações das partes do software, o registro e apresentação da situação dos subprodutos e os pedidos de modificação. Estas atividades visam garantir que os subprodutos de software estejam completos, consistentes e corretos, que é feito o controle do armazenamento dos dados e dos subprodutos e que atende de forma correta a obtenção e distribuição de partes destes subprodutos, do subproduto e do produto final de software.

Gerenciamento de Requisitos. Uma abordagem sistemática para identificar, organizar e documentar os requisitos do sistema, além de firmar e atualizar acordos entre o cliente e a equipe do projeto sobre os requisitos variáveis do sistema.

Gerente de Projeto. O indivíduo ou grupo com responsabilidade total pelo projeto. O gerente de projeto assegura que as tarefas sejam programadas, alocadas e concluídas, de acordo com a programação do projeto, o orçamento e os requisitos de qualidade.

Gestão do Conhecimento. É um processo sistemático, articulado e intencional, apoiado na geração, codificação, disseminação e apropriação de conhecimentos, com o propósito de atingir a excelência organizacional.

Identificação de Configuração. O processo de designação de itens de configuração em um sistema e registro de suas características. A documentação aprovada que define um item de configuração.

Identificação de Riscos. Atividade ou processo necessário para determinar os riscos que podem afetar o projeto e documentar suas características.

Implantação. Uma disciplina no processo de engenharia de software cuja finalidade é assegurar uma transição bem-sucedida do sistema desenvolvido para seus usuários. Estão incluídos *artefatos* como material de treinamento e procedimentos de instalação. São ações procedidas na prática durante o ciclo do projeto de forma a ativar e empregar operacionalmente os processos técnicos e funcionalmente os processos gerenciais continuados.

Implementação. Uma disciplina no processo de engenharia de software cuja finalidade é implementar e realizar teste do desenvolvedor em componentes de software. Definição de como algo é construído ou computado. Por exemplo, uma classe é a implementação de um tipo; um método é a implementação de uma operação.

Implementador. Papel responsável pela codificação e implementação dos subprodutos de software e teste de unidades destes subprodutos. Responsável por gerar as funções para operações dos subsistemas e do sistema de medidas.

Instrumentação. Conjunto de dispositivos de engenharia (sensores, transdutores, mostradores, registradores, transmissores, entre outros), instalados ou interligados em uma determinada plataforma para a aquisição de parâmetros do sistema avaliado.

Integração. Atividade de engenharia em que componentes são combinados em um conjunto executável.

Integrador. Papel responsável pela combinação das unidades ou componentes de software com o hardware. Responsável pela integração e testes do produto no ambiente final de ensaio com todos os subsistemas completos e gerar a documentação para o usuário.

Item de Configuração. Um ou mais componentes de hardware ou de software tratado como uma unidade para propósitos de gerenciamento de configuração. Dados do ciclo de vida do sistema de software também são tratados como uma unidade para propósitos de configuração.

Iteração. Uma sequência distinta de atividades com um plano *baselines* e critério de avaliação, resultando em uma entrega (interna ou externa).

Interface. Uma coleção de operações usadas para especificar um serviço de uma classe ou de um componente. Um conjunto nomeado de operações que caracterizam o comportamento de um elemento.

Interface do Usuário. Constitui-se do hardware ou software (ou ambos) que permite que um usuário interaja com um computador. O termo "interface do usuário" geralmente se refere à apresentação visual e o software que a fundamenta, com os quais interage o usuário.

Interface Gráfica de Usuário (GUI). Um tipo de interface que permite a comunicação dos usuários com o programa através da manipulação de recursos gráficos, em vez da digitação de comandos. Normalmente, a GUI inclui uma combinação de elementos gráficos, dispositivos apontadores, barras de menu e outros menus, janelas sobrepostas e ícones.

Lições Aprendidas. Registro das causas e razões que motivam a modificação dos processos técnicos e gerenciais durante o ciclo de vida do projeto, de modo a compor o histórico deste projeto como parte de documentações ou em informações de interesse para o mesmo.

Linguagem Unificada de Modelagem (UML). Uma linguagem para visualizar, especificar, construir e documentar os artefatos de um sistema intensivo de software.

Manutenção. Atividades ou processos de modificação de software executada excepcionalmente para corrigir erros ou inserir novas funções para suprir novos requisitos.

Melhoria de Processo. É atividade que consiste no exame de problemas ocorridos, restrições e atividades com a identificação destas durante o processo em operação.

Mensagem. Uma especificação da comunicação de informações de uma instância para outra, com a perspectiva de que alguma atividade dela decorra. Uma mensagem pode especificar o aumento de um sinal ou a chamada de uma operação.

METAPLI. Uma Metodologia Especial Tradicional e Ágil de aplicação em Projeto Legado de software em Inovação, para de forma sistemática organizar as etapas a serem seguidas para o desenvolvimento de software, de modo a prover ações práticas para o cumprimento das atividades e geração da documentação de um projeto de software completo.

Método. Forma sistemática e regular de realizar algo; planos ou procedimentos detalhados e ordenados de forma lógica, seguidos de modo a cumprir uma tarefa ou atingir uma meta. A implementação de uma operação, do algoritmo ou do procedimento que gera os resultados de uma operação.

Metodologia de Projeto. Definição sistemática de processos, práticas, técnicas e métodos que auxiliam uma equipe no desenvolvimento e controle de um projeto.

Modelo em Cascata. Um modelo do processo de desenvolvimento de software no qual as atividades constituintes - fase de concepção, de requisitos, de design, de implementação, de teste, de instalação e de verificação - são executadas nessa ordem, provavelmente com pouca ou nenhuma iteração. No modelo de desenvolvimento em cascata, elas ocorrem apenas uma vez, em seqüência, com pouca ou nenhuma sobreposição.

Modernização. Modificação introduzida em um sistema, aperfeiçoando-o tecnologicamente, com objetivo de incrementar suas funções, alterando seu ciclo de vida.

Modificação. É toda e qualquer alteração em processo, documentos e em especificações originais decorrentes dos processos de melhoria, revitalização ou modernização do projeto do sistema de software ou alterações no uso de equipamentos, dispositivos eletrônicos, instrumentações, subproduto ou componente de software.

Norma. É um “documento estabelecido por consenso e aprovado por um organismo reconhecido que fornece para uso comum e repetido, regras, diretrizes ou características para atividades ou seus resultados, visando à obtenção de um grau ideal de ordenação de um dado contexto” (PMBOK, 2004).

Operacionalidade. Atributos que determinam a operação e procedimentos relacionados com a operação do sistema de software através do uso dos subprodutos, do produto final e seus documentos.

Padronização. É o uso mais eficiente possível dos processos de desenvolvimento e de gerenciamento, de modo a assegurar a aplicação de critérios, procedimentos técnicos e operacionais comuns e compatíveis.

Plano de Garantia da Qualidade. Documento contendo o processo necessário para identificar e aplicar os padrões de qualidade e as atividades de qualidade relevantes para o projeto de produtos, subproduto e documentos de um sistema de software e determinar como satisfazê-los para garantir que o projeto emprega todos os processos necessários para atender os requisitos.

Plano de Risco. Documento contendo o processo necessário para decidir como abordar, planejar e executar as atividades de gerenciamento de riscos do projeto de sistema de software.

Plano de Gerenciamento de Configuração. Documento contendo o processo e os procedimentos com orientações técnicas e gerenciais para acompanhamento, revisão, aprovação ou aceitação de mudanças propostas em um sistema de software, e os critérios de aprovação para autorizar as mudanças e liberação dos subprodutos, produtos e documentos de software.

Processo. Uma coleção de atividades desempenhada no ciclo de vida de software para produzir uma saída definível ou um produto.

Programação em Par. É o processo utilizado quando grupos de dois ou mais programadores compartilham um único computador para desenvolver todo o código, ou drivers, ou unidades de software, enquanto o outro revisa os códigos quanto à correta implementação e entendimento dos requisitos, ao mesmo tempo.

Programa de Ensaio. É um conjunto de ações e atividades sobre os serviços prestados durante o ciclo de vida do projeto aerodinâmico, que deve ser cumprido pelos envolvidos em relação ao ensaio em túnel de vento, visando a obtenção de resultados aerodinâmicos ou atendimento de uma necessidade do cliente.

Projeto de Sistema de Software. É um esforço temporário empreendido para criar um produto de software, serviço ou resultado exclusivo para ser usado em seu ambiente operacional.

Rake. Instrumento de medida similar a uma régua em que são instalados diversos *pitots* espaçados e alinhados, usado para tomadas de pressões aerodinâmicas em modelos bidimensionais.

Rastreabilidade. Atributos do software e de documentação que fornecem ligações dos requisitos às implementações com respeito ao desenvolvimento específico e ambiente operacional.

Reengenharia. Técnica de reestruturação ou modificação de software existente, ou desenvolvimento de um novo software, preservando-se inalterada a especificação ou o projeto de software.

Requisitos. Necessidades básicas do cliente explicitadas como um problema a ser resolvido. Um conjunto de atividades que permite identificar as necessidades do cliente de modo a obter uma definição clara das características do sistema de software em termos de funcionalidades, desempenho esperado, restrições do projeto, níveis de qualidade esperados, interfaces com outros elementos ou subsistemas. Processo de estudar as necessidades do usuário ou cliente para se chegar a uma definição dos sistemas, subsistemas, hardwares e softwares necessários.

Requisitos Suplementares. Requisitos adicionais resultantes de um processo de desenvolvimento de software, que pode não ser diretamente rastreável por requisitos.

Reuso. Técnica de utilização de uma função ou biblioteca de função em uma ou mais aplicação, ou em diferentes implementações de uma aplicação. Extensão em que um software pode ser utilizado em outras aplicações que são relacionadas com as funções que o mesmo executa.

Revisões de Software. Avaliação que visa fornecer visibilidade do processo de desenvolvimento. A revisão deve assegurar que em certos pontos de verificação o desenvolvimento do software segue de acordo com os planos e que o software final ou os subprodutos cumprirão todos os requisitos e que o projeto de software está documentado conforme planejado.

Revisor de Engenharia. Papel responsável por avaliar, revisar, arquivar e liberar os processos, métodos, artefatos, subprodutos e produto final do projeto de software.

Revitalização. É o trabalho executado com a finalidade de restaurar a capacidade operacional de um sistema de software, através de substituição ou modificação de equipamentos, dispositivos eletrônicos ou software, desde que não implique na degradação da capacidade operacional deste sistema.

Scanivalve. Instrumento de medida para tomadas de pressões contendo 48 canais.

Strain-gage. Instrumento de medida usado para medir forças e momentos aerodinâmicos como balanças internas aos modelos aerodinâmicos.

Testes: São processos de exercitar ou avaliar um sistema de software ou componentes por meios automáticos ou manuais para verificar que ele satisfaz os requisitos especificados ou para identificar diferenças entre os resultados esperados e os verdadeiros.

Transdutor. É um dispositivo que transforma um tipo de energia em outro tipo, utilizando para isso um sensor que recebe os dados e os transforma. Por exemplo, o sensor pode traduzir informação não elétrica velocidade, posição, temperatura em informação elétrica.

Testes de Aceitação. Teste forma conduzido para determinar se um sistema de software satisfaz ou não seus critérios de aceitação e para permitir ao cliente determinar se aceita ou não o sistema.

Testes de Sistema. Processo de testar um sistema integrado de hardware e software para verificar se satisfaz os requisitos especificados.

Testes de Unidade. Verificações dos componentes de um software através de teste funcional, desenvolvido a partir da especificação das funções previstas para o componente, ou teste estrutural, desenvolvido a partir da descrição da estrutura do componente.

Teste Funcional. Teste conduzido para demonstrar a operacionalidade das funções que foram especificadas.

Teste de Integração. Técnica sistemática para a construção da estrutura de um software, realizando-se ao mesmo tempo, testes para descobrir erros associados a interfaces. A partir dos módulos testados a nível de unidade, construir a estrutura do programa que foi determinada pelo projeto.

Usabilidade. É a extensão na qual um produto pode ser usado por usuários específicos para alcançar objetivos específicos com efetividade, eficiência e satisfação em um contexto de uso específico.

Validação. Atividades ou processo para determinar se os requisitos, os documentos, os subprodutos e o produto final atendem ao uso específico proposto.

Verificação. Atividades ou processo para determinar se os documentos, os subprodutos e o produto de uma determinada fase do ciclo de desenvolvimento atendem os requisitos estabelecidos em fases anteriores ou condições impostas a eles.

APÊNDICE D CÓDIGOS DOS SUBPRODUTOS DE SOFTWARE

Os subprodutos de software implementados e testados via METAPLI são constituídos de códigos para o tratamento dos dados, em sua maioria, antes de serem processados pelo software de redução de dados. Neste trabalho, limitou-se a mostrar partes destes códigos dos subprodutos implementados, as quais contribuíram para obtenção dos resultados dos experimentos do TA-2, provendo qualidade e a confiança no funcionamento para o projeto do novo sistema de medidas.

D.1 PARTE DO CÓDIGO DO SUBPRODUTO DE SOFTWARE COM APLICATIVO EXCEL PARA ORDENAÇÃO DE DADOS DA SCANIVALVE - SVE

As partes do código do SVE podem ser vistas nas figuras D.1 a D.3 a seguir.

<pre> 'Sub ARRUMAFAP() Range("G10:G42").Select Selection.Copy Windows(Nome).Activate Range("B5").Select Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, _ SkipBlanks:=False, Transpose:=False Range("B38").Select </pre>	<pre> 'CALHA Windows(MACROMV).Activate Range("D10:D14").Select Application.CutCopyMode = False Selection.Copy Windows(Nome).Activate Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, _ SkipBlanks:=False, Transpose:=False </pre>
---	--

Figura D.1 - Parte do SVE a) ordena dados flap b) ordena dados calha.

<pre> 'CORPO CENTRAL Windows(MACROMV).Activate Range("D15:D55").Select Application.CutCopyMode = False Selection.Copy Windows(Nome).Activate Range("B43").Select Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, _ SkipBlanks:=False, Transpose:=False </pre>	<pre> 'SPOILER Windows(MACROMV).Activate Range("F10:F23").Select Application.CutCopyMode = False Selection.Copy Windows(Nome).Activate Range("B84").Select Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, _ SkipBlanks:=False, Transpose:=False </pre>
---	---

Figura D.2: Parte do SVE a) ordena dados corpo central b) ordena dados spoiler.

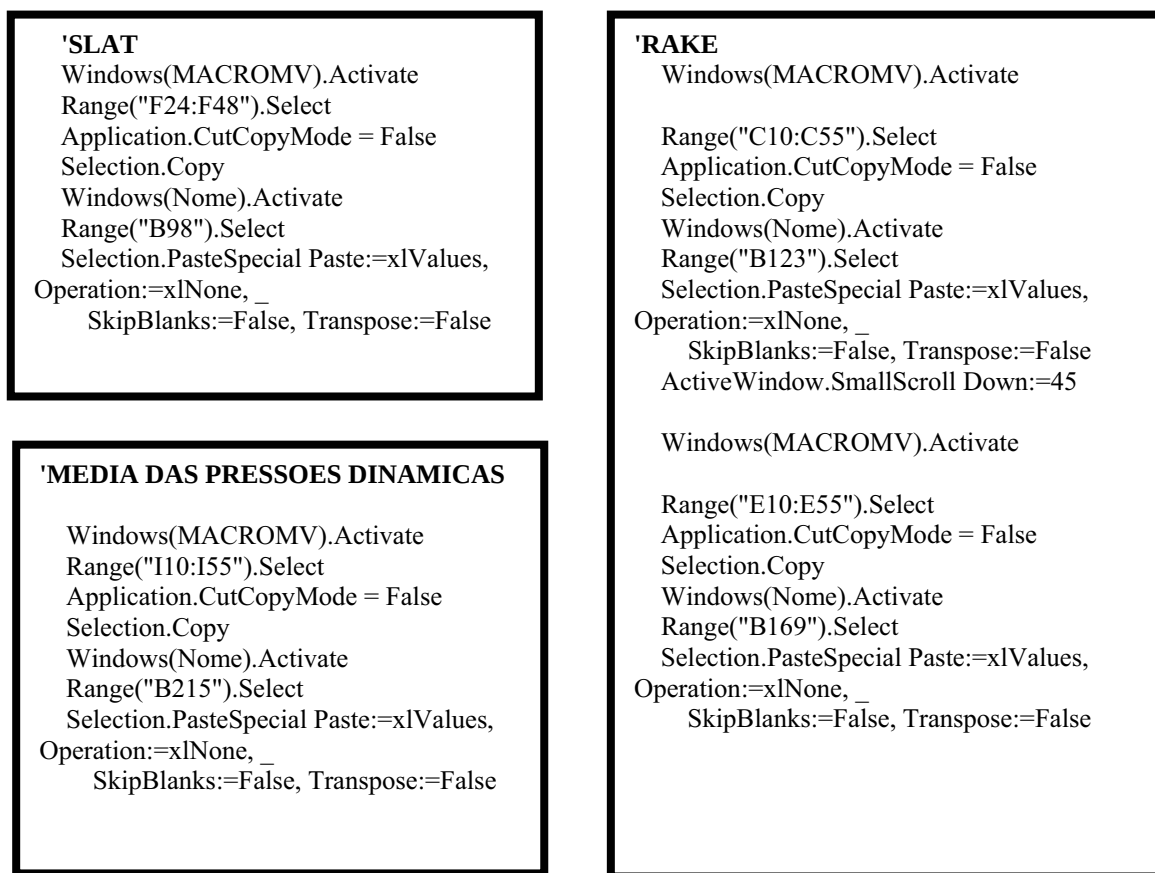


Figura D.3 - Parte do SVE a) ordena dados slat b) ordena dados rake de arrasto c) Calcula média das pressões dinâmicas.

D.2 PARTE DO CÓDIGO DO SUBPRODUTO DE SOFTWARE DE TRANSFERÊNCIA DE DADOS DA HP PARA O PC - STD

O subproduto STD é usado neste trabalho para processar a transferência dos dados da plataforma HP para o PC. Os dados e resultados obtidos com os softwares em BASIC da HP precisavam ser recuperados, dada a importância dos projetos aerodinâmicos ensaiados no TA-2. Para a transferência destes dados se gerou o subproduto STD, cujas partes do código podem ser vistas nas figuras D.4 a D.7 a seguir.

```

void inicializa ( )
/*
=====
    Inicialização da Placa HPIB
=====
*/
{
    error = IORESET (isc);
    error_handler (error, "IORESET");
    error = IOTIMEOUT (isc, 5.0);
    error_handler (error, "IOTIMEOUT");
    error = IOCLEAR (isc);
    error_handler (error, "IOCLEAR");
    error = IOEOI (isc, 0);
    error_handler (error, "IOEOI");
}

```

Figura D.4 - Parte do subproduto STD para inicialização da placa HP-IB.

```

void error_handler (int error, char *routine)
/*
=====
    Tratamento de erros
=====
*/
{
    char *estring;
    char ch;
    if (error != NOERR)
    {
        wopen (1,1,5,50,1,WHITE|_BROWN,WHITE|_BROWN);
        add_shadow( );
        wprintf("Erro na chamada de %s \n", routine);
        wprintf("    Error = %d    %s \n", error, errstr(error);
        Pause( );
    }
}

```

Figura D.5 - Parte do subproduto STD para tratamento de erros.


```

void readout2 ( )
/*
=====
Leitura de matrizes ZAB e mV (ENS)
=====
*/
{
    int x;
    int counter, error;
    float temp;
    x=0;
    message(3,0);
    counter=70;
    error=IOENTERS(hp_addr, filename, &counter);
    error_handler(error,"IOENTERS");
    message(4,0);
    if(strlen(filename) < 6)
        exit (0);

```



```

/*          */
/* numero do ensaio */
/*          */
message(6,0);
message(7,0);
counter=14;
error=IOENTERS(hp_addr, ensaio, &counter);
erro_handler(error,"IOENTER");
message(8,0);

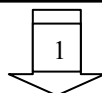
```



```

/*          */
/* recebe array ZAB */
/*          */
message(24,0);
counter=18;
zab_array=(float *) malloc(counter*sizeof(float));
errorIOENTERA(hp_addr,zab_array, &counter);
error_handler(error,"IOENTERA");

```



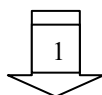
```

/*          */
/* recebe quantidade de pontos */
/*          */
message(16,0);
error=IOENTER(hp_addr, &temp);
error_handler(error,"IOENTER");
qty_ptos[x]=(int) temp;
if(qty_ptos[x]== 0)
{
    message(17,x);
    return;
}
else
    message(18,x);
message(19,0);
error=IOENTER(hp_addr, &temp);
error_handler(error,"IOENTER");
cc[x]=(int) temp;

```



Figura D.6 - Parte do STD para leitura da matriz ZAB e dados do ensaio.



```

/*                      */
/* recebe zero de sensores */
/*                      */
message(33,0);
counter=9;
zero_sens_array=(float *) malloc(counter*sizeof(float));
error=IOENTERA(hp-addr,zero_sens_array, &counter);
error_handler(error,"IOENTERA");

```



```

/*                      */
/* recebe array mV      */
/*                      */
message(23,0);
counter=qty_ptos[x] * qty_coefs[x];
mv_array=(float *) malloc(counter*sizeof(float));
error=IOENTERA(hp_addr,mv_array, &counter);
error_handler(error,"IOENTERA");
}

```

Figura D.7 - Continuação da parte do subproduto STD para leitura da matriz ZAB e dados do ensaio.

D.3 PARTE DO CÓDIGO DO SOFTWARE COM APLICATIVO ARENA PARA CÁLCULO DE DISTRIBUIÇÃO DE DADOS - SAV

Este processo de projeto do SAV foi definido gerencialmente como pré-redução dos dados. As distribuições estatísticas dos dados são de suma importância para validação dos mesmos. Neste processo pode-se identificar problemas com os dados e antecipar soluções de processo de engenharia. Os resultados das distribuições das forças e momentos de R2 a R6 usando o SAV podem ser vistas nas figuras D.8 a D.17 a seguir.

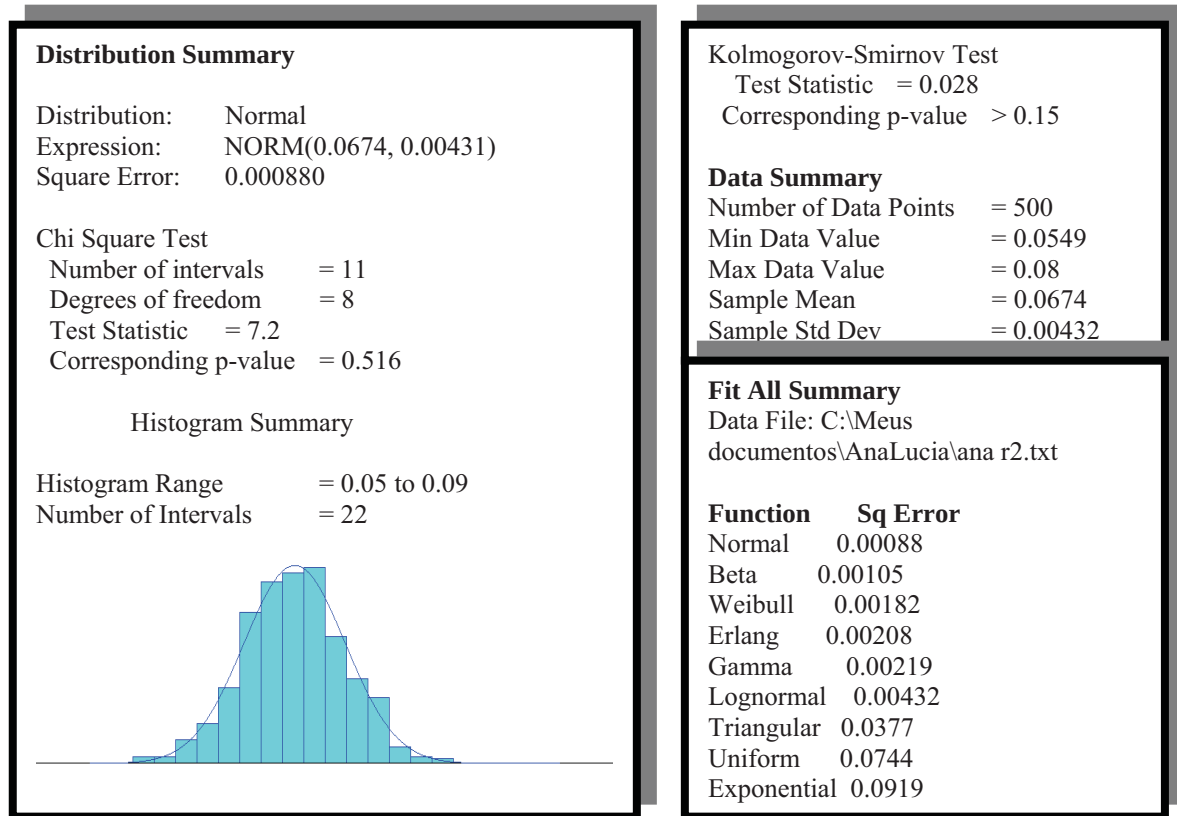


Figura D.8 - Distribuição de R2 para história de tempo de 10 segundos.

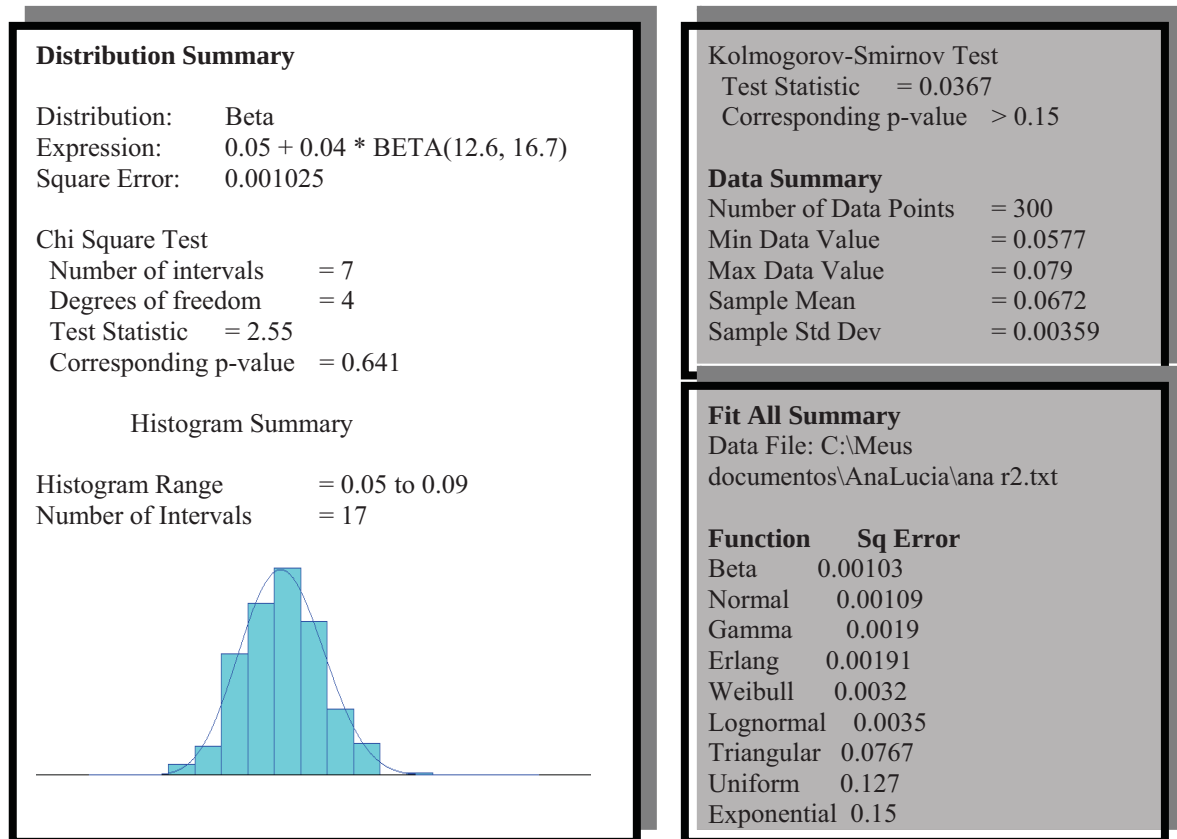


Figura D.9 - Distribuição de R2 para história de tempo de 6 segundos.

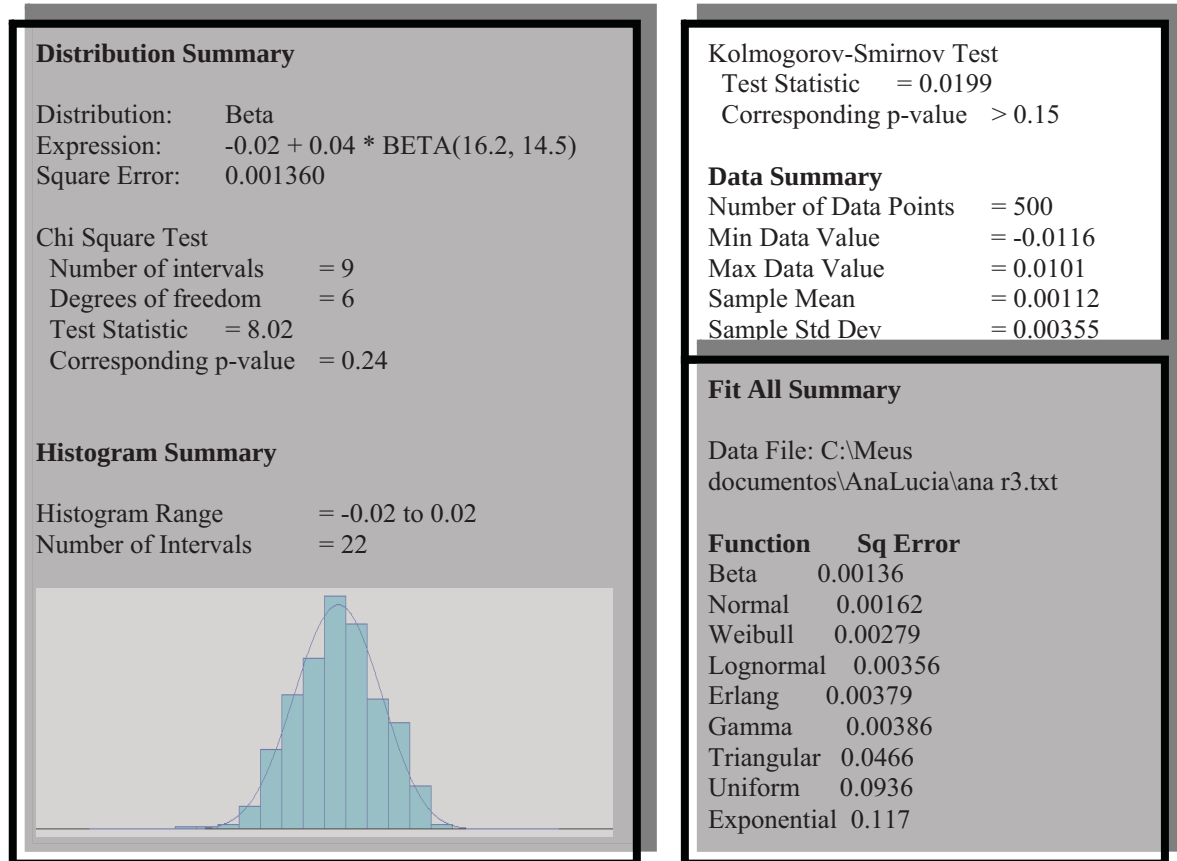


Figura D.10 - Distribuição de R3 para história de tempo de 10 segundos.

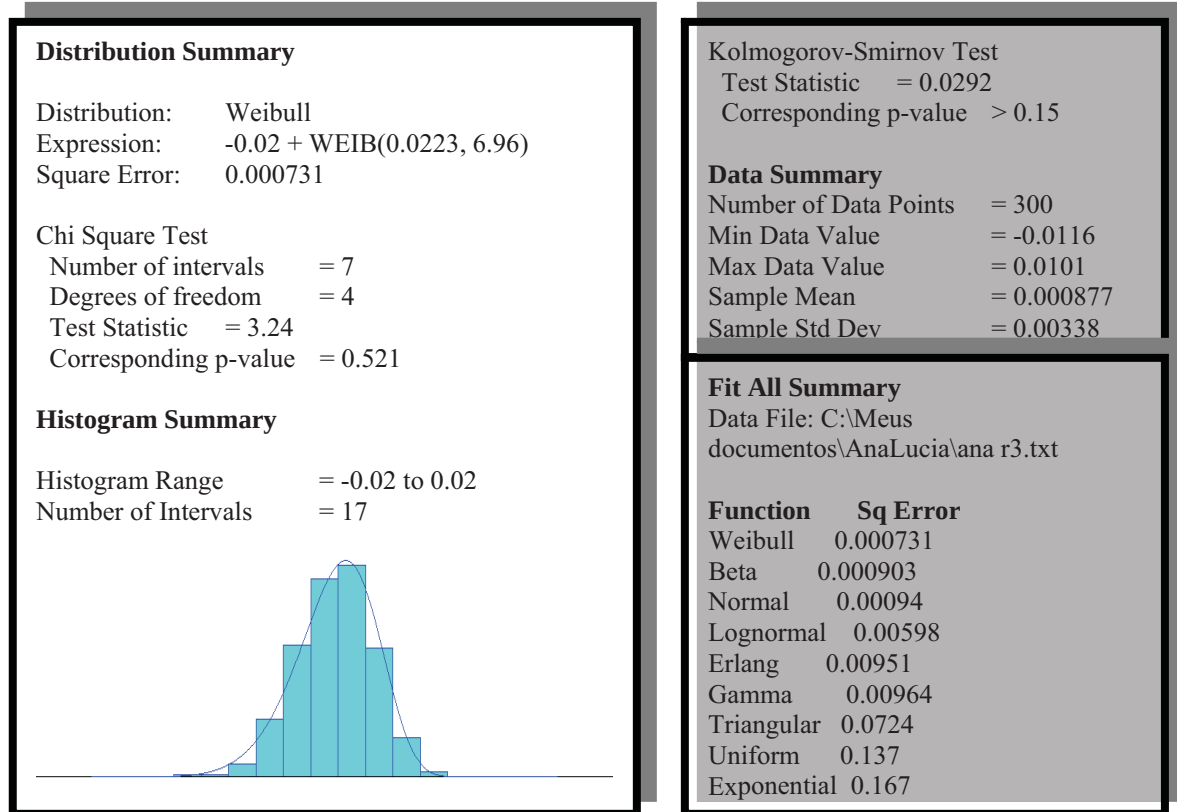


Figura D.11 - Distribuição de R3 para história de tempo de 6 segundos.

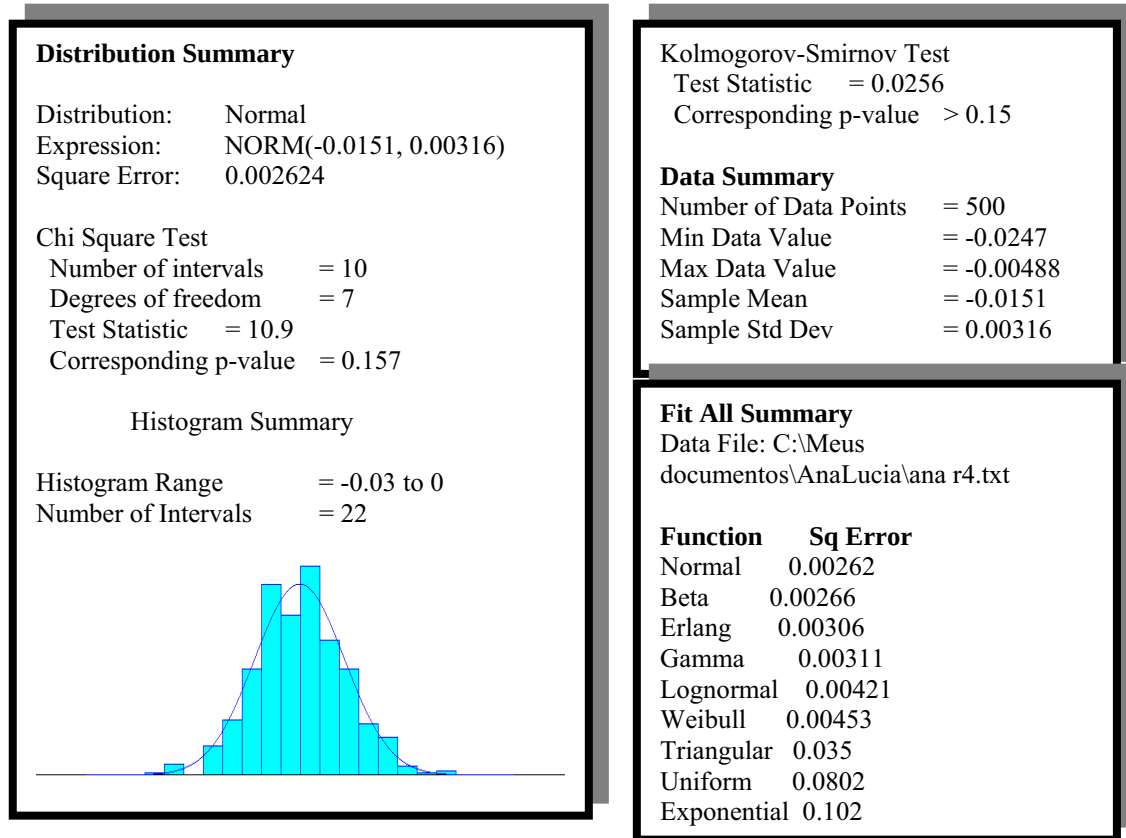


Figura D.12 - Distribuição de R4 para história de tempo de 10 segundos.

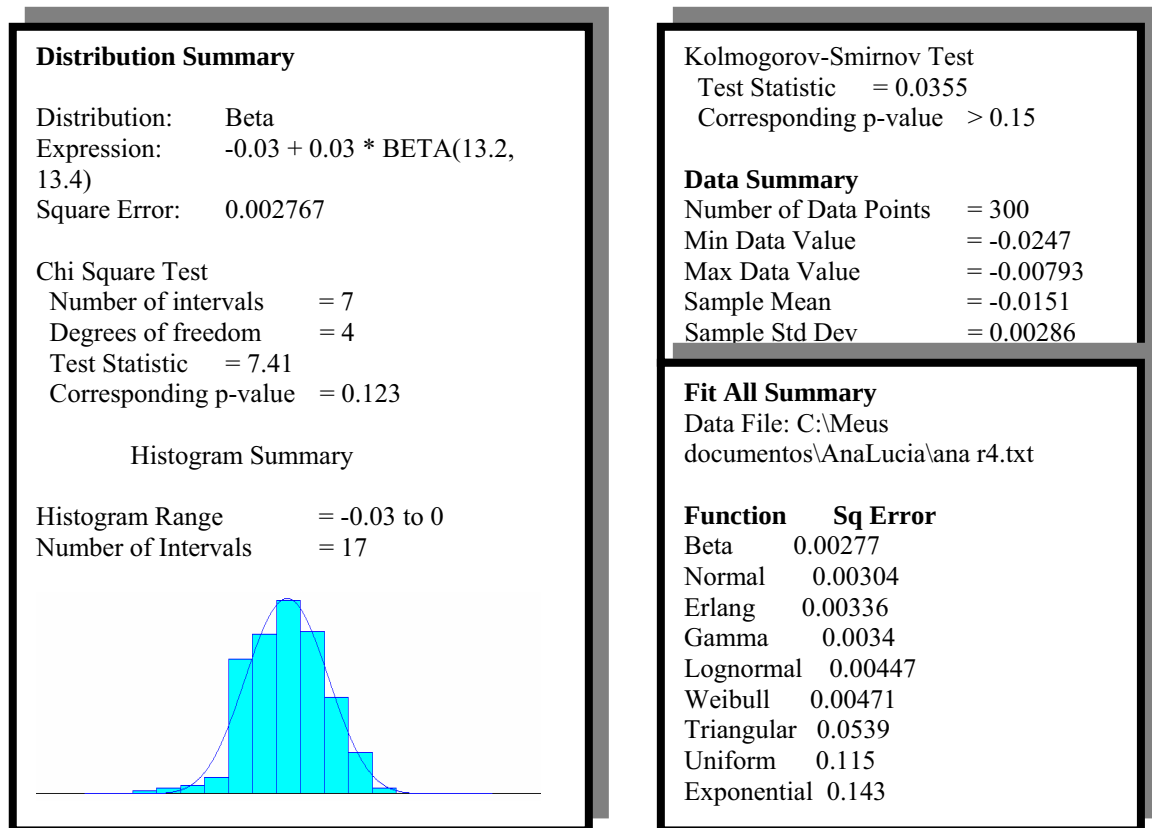


Figura D.13: Distribuição de R4 para história de tempo de 6 segundos.

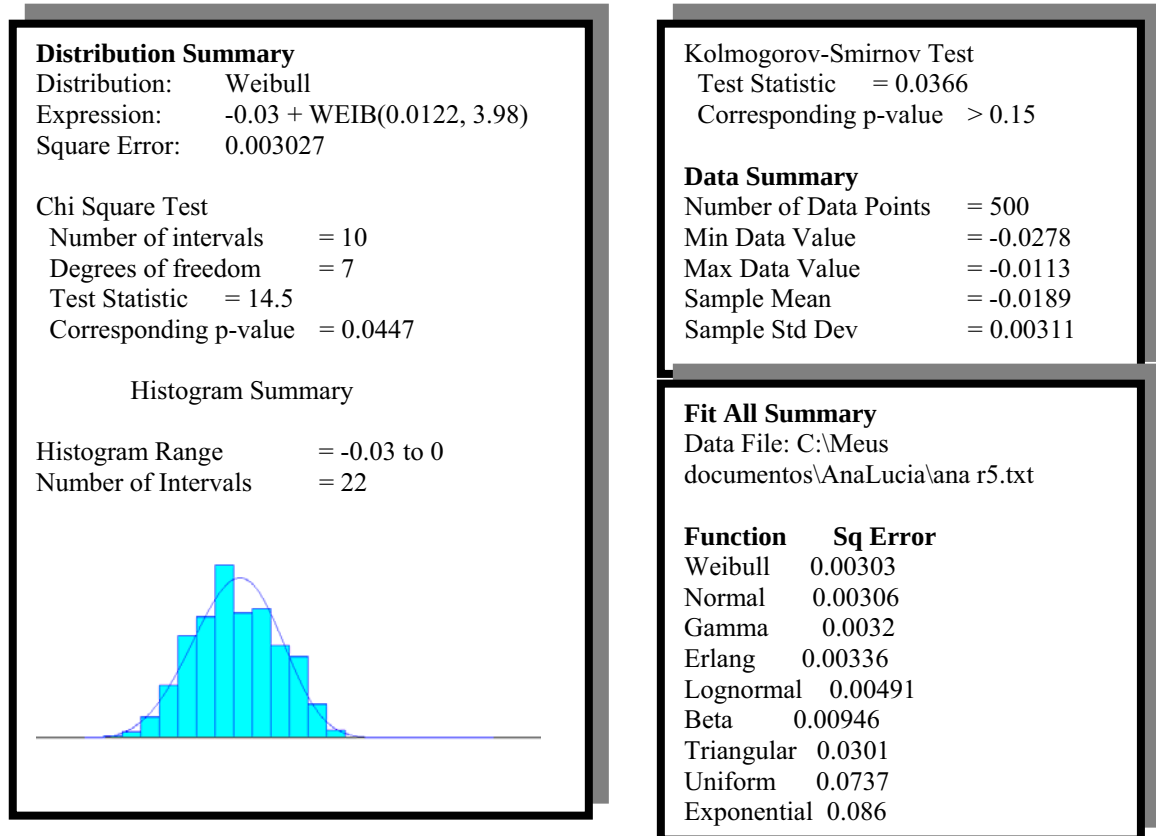


Figura D.14- Distribuição de R5 para história de tempo de 10 segundos.

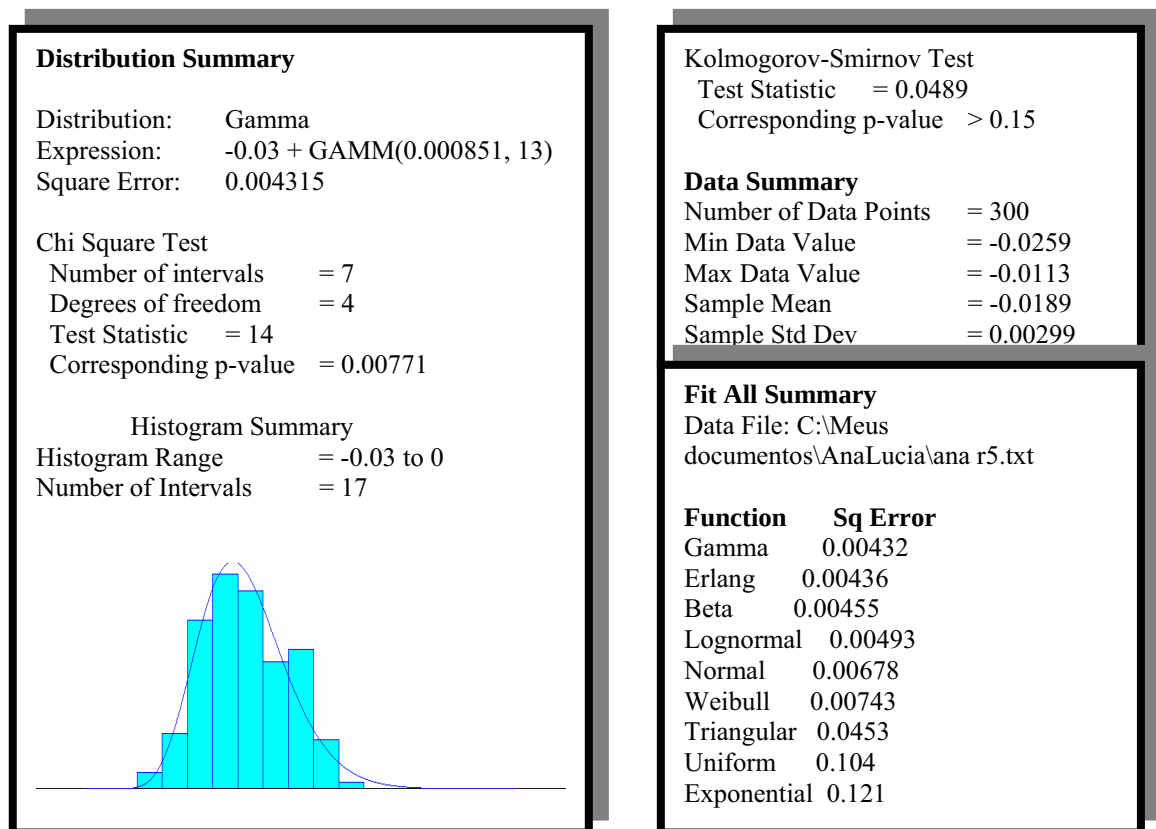


Figura D.15 - Distribuição de R5 para história de tempo de 6 segundos.

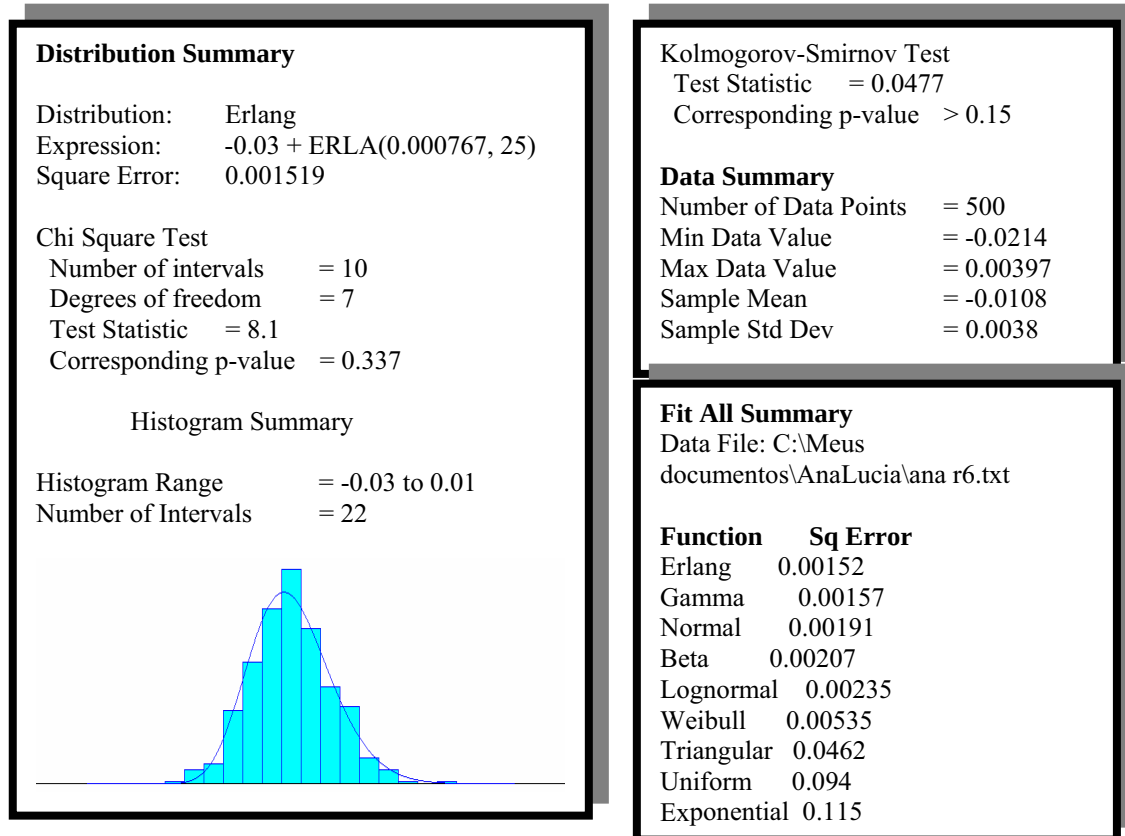


Figura D.16 - Distribuição de R6 para história de tempo de 10 segundos.

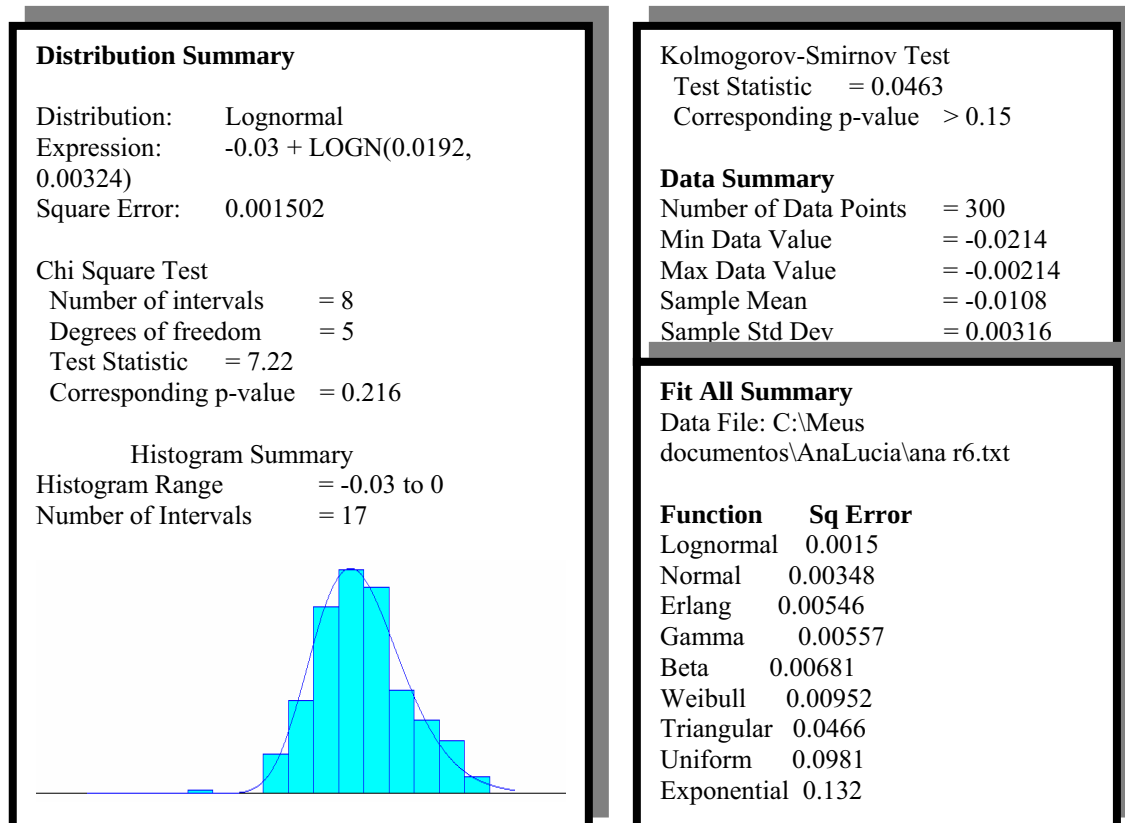


Figura D.17 - Distribuição de R6 para história de tempo de 6 segundos.

D.4 PARTE DO CÓDIGO DO SUBPRODUTO DE SOFTWARE DE REDUÇÃO DE DADOS - SRD

O subproduto SRD é usado neste trabalho para processar a redução de dados da plataforma PC. Os dados obtidos com os softwares da plataforma HP e PC dos projetos aerodinâmicos ensaiados no TA-2, são processados com o subproduto SRD. Partes do código podem ser vistas nas figuras D.18 a D.20 a seguir.

<pre>extern float Zq_R1_Lin, Zq_R1_Qdr, Zq_R1_Cub, Zq_R2_Lin, Zq_R2_Qdr, Zq_R2_Cub, Zq_R3_Lin, Zq_R3_Qdr, Zq_R3_Cub, Zq_R4_Lin, Zq_R4_Qdr, Zq_R4_Cub, Zq_R5_Lin, Zq_R5_Qdr, Zq_R5_Cub, Zq_R6_Lin, Zq_R6_Qdr, Zq_R6_Cub;</pre>	<pre>extern float Zab_R1_TIn, Zab_R1_Lin, Zab_R1_Qdr, Zab_R1_Cub, Zab_R2_TIn, Zab_R2_Lin, Zab_R2_Qdr, Zab_R2_Cub, Zab_R3_TIn, Zab_R3_Lin, Zab_R3_Qdr, Zab_R3_Cub,</pre>	<pre>extern float Zab_R4_TIn, Zab_R4_Lin, Zab_R4_Qdr, Zab_R4_Cub, Zab_R5_TIn, Zab_R5_Lin, Zab_R5_Qdr, Zab_R5_Cub, Zab_R6_TIn, Zab_R6_Lin, Zab_R6_Qdr, Zab_R6_Cub;</pre>
---	---	---

Figura D.18 - Declaração de variáveis para calcular ZQ e ZAB

<pre>Programa Calcula_ZQ void Calcula_ZQ() { Zq_R1 = Zero_R1 - (Zq_R1_Lin * Qd[C] + Zq_R1_Qdr * pow(Qd[C],2.) + Zq_R1_Cub * pow(Qd[C],3.)); Zq_R2 = Zero_R2 - (Zq_R2_Lin * Qd[C] + Zq_R2_Qdr * pow(Qd[C],2.) + Zq_R2_Cub * pow(Qd[C],3.)); Zq_R3 = Zero_R3 - (Zq_R3_Lin * Qd[C] + Zq_R3_Qdr * pow(Qd[C],2.) + Zq_R3_Cub * pow(Qd[C],3.)); Zq_R4 = Zero_R4 - (Zq_R4_Lin * Qd[C] + Zq_R4_Qdr * pow(Qd[C],2.) + Zq_R4_Cub * pow(Qd[C],3.)); Zq_R5 = Zero_R5 - (Zq_R5_Lin * Qd[C] + Zq_R5_Qdr * pow(Qd[C],2.) + Zq_R4_Cub * pow(Qd[C],3.)); Zq_R6 = Zero_R6 - (Zq_R6_Lin * Qd[C] + Zq_R6_Qdr * pow(Qd[C],2.) + Zq_R6_Cub * pow(Qd[C],3.)); printf("\n ZQ \n"); printf("%f %f %f %f %f %f\n", Zq_R1, Zq_R2, Zq_R3, Zq_R4, Zq_R5, Zq_R6); }</pre>	<pre>Programa Calcula_ZAB void Calcula_ZAB() { Zab_R1 = Zq_R1 - (Zab_R1_TIn + Zab_R1_Lin * Ang_Red[C] + Zab_R1_Qdr * pow(Ang_Red[C],2.) + Zab_R1_Cub * pow(Ang_Red[C],3.)); Zab_R2 = Zq_R2 - (Zab_R2_TIn + Zab_R2_Lin * Ang_Red[C] + Zab_R2_Qdr * pow(Ang_Red[C],2.) + Zab_R2_Cub * pow(Ang_Red[C],3.)); Zab_R3 = Zq_R3 - (Zab_R3_TIn + Zab_R3_Lin * Ang_Red[C] + Zab_R3_Qdr * pow(Ang_Red[C],2.) + Zab_R3_Cub * pow(Ang_Red[C],3.)); Zab_R4 = Zq_R4 - (Zab_R4_TIn + Zab_R4_Lin * Ang_Red[C] + Zab_R4_Qdr * pow(Ang_Red[C],2.) + Zab_R4_Cub * pow(Ang_Red[C],3.)); Zab_R5 = Zq_R5 - (Zab_R5_TIn + Zab_R5_Lin * Ang_Red[C] + Zab_R5_Qdr * pow(Ang_Red[C],2.) + Zab_R5_Cub * pow(Ang_Red[C],3.)); Zab_R6 = Zq_R6 - (Zab_R6_TIn + Zab_R6_Lin * Ang_Red[C] + Zab_R6_Qdr * pow(Ang_Red[C],2.) + Zab_R6_Cub * pow(Ang_Red[C],3.)); printf("\n ZAB \n"); printf("%f %f %f %f %f %f\n",Zab_R1, Zab_R2, Zab_R3, Zab_R4, Zab_R5, Zab_R6); }</pre>
--	---

Figura D.19 - Programa para cálculo de ZQ e ZAB

Programa Calcula_Forca

```

void Calcula_Forca()
{
    Forca[0] = Zab_R1 / Const_R1 / VOLTS;
    Forca[1] = Zab_R2 / Const_R2 / VOLTS;
    Forca[2] = Zab_R3 / Const_R3 / VOLTS;
    Forca[3] = Zab_R4 / Const_R4 / VOLTS;
    Forca[4] = Zab_R5 / Const_R5 / VOLTS;
    Forca[5] = Zab_R6 / Const_R6 / VOLTS;

    C1 = 6;          /* alfa */
    if(Angulo == BETA)
    {
        Forca[6] = sin(Ang_Red[C]) * Deg_Rad;
        Forca[7] = cos(Ang_Red[C]) * Deg_Rad;
        C1 = 8;
    }

    Forca_R1 = 0.0;
    Forca_R2 = 0.0;
    Forca_R3 = 0.0;
    Forca_R4 = 0.0;
    Forca_R5 = 0.0;
    Forca_R6 = 0.0;

```

```

switch(Angulo)
{
    case ALFA: /* Angulo Alfa */

        /* Calculo da forca de R1 */

        for (I=0; I<6; I++)
        {
            Forca_R1 = Forca_R1 + Malfa_R1[I] *
Forca[I];
        }
        Rf = 6;

        for (J=0; J<6; J++)
        {
            for (I=J; I<C1; I++)
            {
                Forca_R1 = Forca_R1 + Malfa_R1[Rf]
* Forca[J] * Forca[I]; /*beta*/
                Rf++;
            }
        }
    }

```

Figura D.20 - Programa de cálculo de forças e momentos.