



UNIVERSIDADE METODISTA DE PIRACICABA
FACULDADE DE CIÊNCIAS EXATAS E DA NATUREZA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**TÉCNICAS DE INSPEÇÃO APLICADAS À AVALIAÇÃO DE
REQUISITOS DE SISTEMAS DE SOFTWARE: UM ESTUDO
COMPARATIVO**

LILIAN APARECIDA BERTINI

ORIENTADORA: PROF^a. DRA. TEREZA GONÇALVES KIRNER

Dissertação apresentada ao Programa de Pós -
Graduação em Ciência da Computação, da
Faculdade de Ciências Exatas e da Natureza, da
Universidade Metodista de Piracicaba – UNIMEP,
como requisito para obtenção do Título de Mestre
em Ciência da Computação.

PIRACICABA
2006



UNIVERSIDADE METODISTA DE PIRACICABA
FACULDADE DE CIÊNCIAS EXATAS E DA NATUREZA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**TÉCNICAS DE INSPEÇÃO APLICADAS À AVALIAÇÃO DE
REQUISITOS DE SISTEMAS DE SOFTWARE: UM ESTUDO
COMPARATIVO**

LILIAN APARECIDA BERTINI

ORIENTADORA: PROF^a. DRA. TEREZA GONÇALVES KIRNER

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, da Faculdade de Ciências Exatas e da Natureza, da Universidade Metodista de Piracicaba – UNIMEP, como requisito para obtenção do Título de Mestre em Ciência da Computação.

PIRACICABA
2006

Bertini, Lílian Aparecida.

Técnicas de Inspeção Aplicadas à Avaliação de Requisitos de
Sistemas de Software: Um Estudo Comparativo.

Piracicaba, 2006.

152 p.

Orientadora: Prof^a. Dr^a. Tereza Gonçalves Kirner.

Dissertação (mestrado), Programa de Pós-graduação em Ciência da
Computação – Universidade Metodista de Piracicaba.

1. Engenharia de Software 2. Estudo de Metodologias.

TÉCNICAS DE INSPEÇÃO APLICADAS À AVALIAÇÃO DE REQUISITOS DE SISTEMAS DE SOFTWARE: UM ESTUDO COMPARATIVO

AUTORA: LILIAN APARECIDA BERTINI

ORIENTADORA: TEREZA GONÇALVES KIRNER

Dissertação de Mestrado defendida e aprovada em 08 de março de 2006, pela Banca Examinadora constituída dos Professores:

Prof^{fa}. Dr^a. Tereza Gonçalves Kirner (Orientadora)
UNIMEP

Prof^{fa}. Dr^a. Maria Imaculada de Lima Montebelo
UNIMEP

Prof^{fa}. Dr^a. Ariadne Maria Brito Rizzoni Carvalho
UNICAMP

À

Minha filha Mayne pelo amor e compreensão

Aos

Meus pais Arlindo e Aldaíza

AGRADECIMENTOS

A Deus, que ilumina meu caminho e dá forças para conquistar todos os meus objetivos, dando-me a oportunidade de realizar este trabalho.

Ao meu pai, a minha mãe e aos meus irmãos Adriana, Aldo e Júnior que sempre incentivaram e acreditaram no meu sucesso.

A minha filha, pela compreensão da minha ausência durante meus estudos e pelo seu carinho transmitido.

À Prof^a Dra Tereza Gonçalves Kirner, pela compreensão, amizade e conhecimentos transmitidos para o desenvolvimento do trabalho e pela dedicação ao curso.

Aos amigos que conquistei ao longo deste curso, em especial, para Cristine do C.S.B. de Moraes, Simone Franceto, Alessandra Vilches, José Inácio de Grande, Isabella Pioli Trevisani e Rodrigo Cândido Leme.

Aos Professores Idemauro Antonio R. de Lara e Maria Imaculada de Lima Montebelo da área de Estatística pela colaboração na análise dos resultados do estudo.

A empresa AGSI Sistemas, em especial para Jacqueline Petinon por ceder o Sistema PCMSO para a realização do meu trabalho.

Aos meus colegas do curso de mestrado que participaram do estudo para a realização e finalização desse trabalho.

“Não importa quais sejam os obstáculos e dificuldades. Se estamos possuídos de uma inabalável determinação, conseguimos superá-los.”

Dalai Lama

RESUMO

A inspeção de software auxilia a detectar defeitos e a removê-los antes que se propaguem para as próximas fases do ciclo de vida. Para realizar a inspeção, existem técnicas específicas conhecidas como técnicas de leitura. O objetivo desta dissertação de mestrado foi realizar um estudo comparativo entre três técnicas de leitura indicadas para a avaliação de documentos de requisitos de software: Leitura baseada em *Checklists*, Leitura baseada em Cenários e Leitura baseada em Perspectivas. A eficiência dessas três técnicas em detectar defeitos em um documento de requisitos de software foi avaliada através de um estudo empírico envolvendo profissionais de análise de sistemas/ engenharia de software. O trabalho relata o estudo empírico, seu planejamento, realização e análise dos resultados.

PALAVRAS-CHAVE: Inspeção de Software, Leitura baseada em *Checklists*, Leitura baseada em Cenários, Leitura baseada em Perspectivas.

TECHNIQUES OF INSPECTION APPLIED A EVALUATION OF REQUIREMENTS OF SOFTWARE SYSTEMS: A COMPARATIVE STUDY

ABSTRACT

Software inspection helps to detect faults and remove them before their propagation to the next phases of the lifecycle. There are specific techniques, known as reading techniques to execute such inspection. The goal of this master's thesis is to develop a comparative study among three reading techniques indicated to evaluate software requirements document: Checklist-based Reading, Scenario-based Reading and Perspective-based Reading. The efficiency of these three techniques applied to detect faults in a software requirements document was evaluated through an empirical study involving systems analysis and software engineers professionals. This thesis describes the empirical study, its planning, implementation and results analyses.

KEY WORDS: Software Inspection, Checklist-based Reading, Scenario-based Reading, Perspective-based Reading.

SUMÁRIO

CAPÍTULO 1- INTRODUÇÃO	16
1.1. CONTEXTUALIZAÇÃO DA PESQUISA	16
1.2. OBJETIVOS	17
1.3. IMPORTÂNCIA E JUSTIFICATIVAS DA PESQUISA	17
1.4. RESULTADOS E CONTRIBUIÇÕES ESPERADOS	19
1.5. ORGANIZAÇÃO DO TRABALHO	19
CAPÍTULO 2 - INSPEÇÃO DE SOFTWARE	21
2.1 PROCESSO DE INSPEÇÃO.....	21
2.1.1 Conceitos	21
2.1.2 Objetivos da Inspeção.....	22
2.1.3 Processo de Realização da Inspeção	25
2.1.4 Equipe de Inspeção.....	27
2.1.5 Dificuldades da Inspeção	28
2.1.6 Contribuições da Inspeção.....	29
2.2 ENGENHARIA DE REQUISITOS DE SOFTWARE.....	30
2.2.1 Documento de Requisitos de Software	34
2.2.2 Defeitos em Requisitos de Software	38
2.3 CLASSIFICAÇÃO DE DEFEITOS	39
2.3.1 Classificação de Defeitos no Documento de Requisitos de Software	39
CAPÍTULO 3 - TÉCNICAS DE INSPEÇÃO APLICADAS A REQUISITOS DE SOFTWARE ...	45
3.1 CONCEITOS	45
3.2 DOCUMENTAÇÃO.....	47
3.3 ESQUEMA DE CLASSIFICAÇÃO DE TÉCNICAS DE LEITURA	48
3.4 TÉCNICAS DE LEITURA.....	49
3.4.1 Técnica de Leitura <i>Ad Hoc</i>	51
3.4.2 Técnica de Leitura Baseada em <i>Checklists</i>	51
3.4.3 Técnica de Leitura Baseada em Cenários	53
3.4.4 Técnica de Leitura Baseada em Perspectivas	54
3.5 AVALIAÇÃO EMPÍRICA DE TÉCNICAS DE LEITURA PARA INSPEÇÃO DE SOFTWARE	56
3.5.1 Experimentos com a Técnica de Leitura Baseada em Perspectivas	56
3.5.2 Estudos Empíricos em Inspeção de Requisitos de Software	58
CAPÍTULO 4 - UM ESTUDO EMPÍRICO PARA AVALIAÇÃO DE TÉCNICAS DE INSPEÇÃO APLICADAS A DOCUMENTOS DE REQUISITOS DE SOFTWARE.....	66
4.1 OBJETIVO	66
4.2 PERGUNTA DA PESQUISA.....	67
4.3 HIPÓTESES DO ESTUDO.....	67
4.4 DEFINIÇÃO DOS PARTICIPANTES	69
4.5 DOCUMENTO INSPECIONADO.....	69
4.6 CLASSIFICAÇÃO DE DEFEITOS EMPREGADA.....	70
4.7 REALIZAÇÃO DO ESTUDO EMPÍRICO.....	71
4.8 FORMULÁRIOS E DOCUMENTOS	75

4.8.1	Formulário das Técnicas de Leitura	76
4.9	METODOLOGIAS ESTATÍSTICAS PARA ANÁLISE DOS DADOS.....	77
4.9.1	Análise da Hipótese 1	78
4.9.2	Análise da Hipótese 2	79
CAPÍTULO 5 - RESULTADOS E DISCUSSÃO		80
5.1	APRESENTAÇÃO DOS DADOS	80
5.2	ANÁLISE DA HIPÓTESE 1 COM O TESTE KRUSKAL WALLIS E TUKEY-KRAMER.	83
5.2.1	Análise da Classificação de Defeitos por Comissão e Omissão	85
5.2.2	Análise dos Tipos de Defeitos por Comissão.....	86
5.2.3	Análise dos Tipos de Defeitos por Omissão.....	87
5.3	ANÁLISE DA HIPÓTESE 2 COM O TESTE DE CORRELAÇÃO DE PEARSON.....	89
5.4	ANÁLISE DO FORMULÁRIO DE ASPECTOS GERAIS	91
5.5	CONSIDERAÇÕES FINAIS.....	102
CAPÍTULO 6 - CONCLUSÃO.....		107
6.1	RESUMO DA PESQUISA	107
6.2	CONTRIBUIÇÕES DO ESTUDO	108
6.3	PERSPECTIVAS FUTURAS.....	109
REFERÊNCIAS BIBLIOGRÁFICAS.....		111
APÊNDICE		117
APÊNDICE A – DOCUMENTOS DE INSPEÇÃO		117
A.1	DRS – PCMSO	118
A.2	INSTRUÇÕES DO ESTUDO	137
A.3	ANOTAÇÃO DE DEFEITOS.....	138
A.4	QUESTIONÁRIO.....	139
A.5	RELATÓRIO DE PARTICIPANTES.....	141
APÊNDICE B – TÉCNICAS DE INSPEÇÃO		142
B.1	TÉCNICA DE INSPEÇÃO: LBCH	143
B.2	TÉCNICA DE INSPEÇÃO: LBCE	145
B.3	TÉCNICA DE INSPEÇÃO: LBPE	148
APÊNDICE C – ASPECTOS GERAIS DO ESTUDO		150
C.1	QUESTÕES DO ESTUDO	151
C.2	INFORMAÇÕES PESSOAIS	152

LISTA DE FIGURAS

FIGURA 2.1 - PROCESSO DE VERIFICAÇÃO DE DOCUMENTOS DE SOFTWARE	23
FIGURA 2.2 - FASES DO PROCESSO DE INSPEÇÃO	26
FIGURA 2.3 - ETAPAS DA CRIAÇÃO DO DRS	31
FIGURA 2.4 - PROCESSO DE ENGENHARIA DE REQUISITOS LOPES (2002)	32
FIGURA 2.5 - MODELO CASCATA NO PROCESSO DE INSPEÇÃO	34
FIGURA 4.1 - FASES DO PROCESSO DE INSPEÇÃO NO ESTUDO.....	75

LISTA DE ABREVIATURAS E SIGLAS

DRS	Documento de Requisitos de Software
LBCh	Leitura Baseada em <i>Checklists</i>
LBCe	Leitura Baseada em Cenários
LBPe	Leitura Baseada em Perspectivas
PCMSO	Programa de Controle Médico e Saúde Ocupacional

LISTA DE QUADROS

QUADRO 2.1	FUNÇÕES E OBJETIVOS DA EQUIPE NO PROCESSO DE INSPEÇÃO	28
QUADRO 2.2	PROCESSO DE ENGENHARIA DE REQUISITOS	32
QUADRO 2.3	DOCUMENTO DE REQUISITOS DE SOFTWARE	37
QUADRO 2.4	CLASSIFICAÇÃO DE DEFEITOS PARA DOCUMENTOS DE SOFTWARE	40
QUADRO 2.5	CLASSIFICAÇÃO DE DEFEITOS PARA REQUISITOS DE SOFTWARE	41
QUADRO 2.6	CLASSES DE DEFEITOS PARA DOCUMENTOS DE SOFTWARE	42
QUADRO 2.7	ESQUEMA DE CLASSIFICAÇÃO DOS DEFEITOS EM SOFTWARES	43
QUADRO 2.8	CLASSIFICAÇÃO DE DEFEITOS PARA DOCUMENTOS DE SOFTWARE	44
QUADRO 3.1	RESUMO DAS CARACTERÍSTICAS DOS TIPOS DE REVISÃO	46
QUADRO 3.2	EXEMPLO DE <i>CHECKLIST</i>	52
QUADRO 3.3	REPLICAÇÕES DO ESTUDO DE LBPE	57
QUADRO 4.1	TABELA DE CLASSIFICAÇÃO DOS DEFEITOS	71
QUADRO 4.2	PLANO DO ESTUDO EMPÍRICO	74
QUADRO 5.1	DEFEITOS COLETADOS COM LBCH	81
QUADRO 5.2	DEFEITOS COLETADOS COM LBCE	81
QUADRO 5.3	DEFEITOS COLETADOS COM LBPE	82
QUADRO 5.4	TEMPO MÉDIO DE EXPERIÊNCIA/ NÚMERO DE DEFEITOS	82
QUADRO 5.5	CORRELAÇÃO LINEAR – NÚMERO DE DEFEITOS DETECTADOS X TEMPO MÉDIO DE EXPERIÊNCIA DOS INSPETORES.....	90

LISTA DE TABELAS

TABELA 5.1 RESULTADO DO TESTE DE KRUSKAL WALLIS NA INSPEÇÃO ENTRE AS EQUIPES, POR TÉCNICA E TIPO DE DEFEITO	83
TABELA 5.2 RESULTADO DO TESTE KRUSKAL WALLIS NA INSPEÇÃO, POR TÉCNICA, NA CLASSE DE DEFEITOS “COMISSÃO”	85
TABELA 5.3 RESULTADO DO TESTE DE KRUSKALL WALLIS NA INSPEÇÃO, POR TÉCNICA, NA CLASSE DE DEFEITOS “OMISSÃO”	85
TABELA 5.4 TESTE DE KRUSKALL WALLIS PARA DETECÇÃO DE DEFEITOS IA POR TÉCNICAS	86
TABELA 5.5 TESTE DE KRUSKALL WALLIS PARA DETECÇÃO DE DEFEITOS II POR TÉCNICAS	86
TABELA 5.6 TESTE DE KRUSKALL WALLIS PARA DETECÇÃO DE DEFEITOS FI POR TÉCNICAS	86
TABELA 5.7 TESTE DE KRUSKALL WALLIS PARA DETECÇÃO DE DEFEITOS SE POR TÉCNICAS	86
TABELA 5.8 TESTE DE KRUSKALL WALLIS PARA DETECÇÃO DE DEFEITOS FO POR TÉCNICAS	87
TABELA 5.9 COMPARAÇÃO PELO MÉTODO DE TUKEY (HSD) AO NÍVEL DE 0,05 DE SIGNIFICÂNCIA COM O VALOR ABSOLUTO DAS MÉDIAS DOS DADOS DO TIPO FO	87
TABELA 5.10 TESTE DE KRUSKALL WALLIS PARA DETECÇÃO DE DEFEITOS PO POR TÉCNICAS	88
TABELA 5.11 COMPARAÇÃO PELO MÉTODO DE TUKEY (HSD) AO NÍVEL DE 0,05 DE SIGNIFICÂNCIA COM O VALOR ABSOLUTO DAS MÉDIAS DOS DADOS DO TIPO PO	88
TABELA 5.12 TESTE DE KRUSKALL WALLIS PARA DETECÇÃO DE DEFEITOS IO POR TÉCNICAS	88
TABELA 5.13 COMPARAÇÃO PELO MÉTODO DE TUKEY (HSD) AO NÍVEL DE 0,05 DE SIGNIFICÂNCIA COM O VALOR ABSOLUTO DAS MÉDIAS DOS DADOS DO TIPO IO.....	89
TABELA 5.14 TESTE DE KRUSKALL WALLIS PARA DETECÇÃO DE DEFEITOS AO POR TÉCNICAS	89

LISTA DE GRÁFICOS

GRÁFICO 5.1	CLASSIFICAÇÃO DE DEFEITOS POR TÉCNICA	84
GRÁFICO 5.2	TIPOS DE DEFEITOS POR TÉCNICA.....	84
GRÁFICO 5.3	DIAGRAMA DE DISPERSÃO.....	91
GRÁFICO 5.4	TEMPO DE EXPERIÊNCIA DOS PARTICIPANTES NA TÉCNICA LBCH.....	92
GRÁFICO 5.5	TEMPO DE EXPERIÊNCIA DOS PARTICIPANTES NA TÉCNICA LBCE.....	92
GRÁFICO 5.6	TEMPO DE EXPERIÊNCIA DOS PARTICIPANTES NA TÉCNICA LBPE.....	93
GRÁFICO 5.7	NÍVEL DE FORMAÇÃO DOS PARTICIPANTES.....	93
GRÁFICO 5.8	EXPERIÊNCIA DOS PARTICIPANTES EM INSPEÇÃO.....	94
GRÁFICO 5.9	CONHECIMENTO DOS PARTICIPANTES NAS TÉCNICAS DE INSPEÇÃO.....	94
GRÁFICO 5.10	CONHECIMENTO DOS PARTICIPANTES SOBRE INSPEÇÃO.....	95
GRÁFICO 5.11	CONHECIMENTO DOS PARTICIPANTES EM DRS.....	96
GRÁFICO 5.12	AVALIAÇÃO DA CLASSIFICAÇÃO DE DEFEITOS – ENTENDIMENTO.....	96
GRÁFICO 5.13	AVALIAÇÃO DA CLASSIFICAÇÃO DE DEFEITOS - COMPLETA.....	97
GRÁFICO 5.14	AVALIAÇÃO DO TIPO DE DEFEITOS - ENTENDIMENTO.....	97
GRÁFICO 5.15	AVALIAÇÃO DO TIPO DE DEFEITOS – COMPLETO.....	98
GRÁFICO 5.16	AVALIAÇÃO DO FORMULÁRIO DE INSPEÇÃO - ENTENDIMENTO.....	98
GRÁFICO 5.17	AVALIAÇÃO DE DETECÇÃO DE DEFEITOS – ENTENDIMENTO.....	99
GRÁFICO 5.18	AVALIAÇÃO DAS QUESTÕES DE DETECÇÃO DE DEFEITOS.....	100
GRÁFICO 5.19	AVALIAÇÃO DOS ARTEFATOS USADOS NA INSPEÇÃO.....	100
GRÁFICO 5.20	AVALIAÇÃO DA TÉCNICA DE INSPEÇÃO – ENTENDIMENTO.....	101
GRÁFICO 5.21	CONHECIMENTOS ADQUIRIDOS NA TÉCNICA DE INSPEÇÃO.....	101
GRÁFICO 5.22	USO DA INSPEÇÃO NO AMBIENTE DE TRABALHO.....	102

CAPÍTULO 1

INTRODUÇÃO

1.1. CONTEXTUALIZAÇÃO DA PESQUISA

Atualmente, uma grande parte das organizações utiliza uma quantidade enorme de informações e depende de aplicações de software para desenvolver seus trabalhos diante de um mercado competitivo que exige da empresa a informação para conduzir seus negócios (ROCHA, 2001). Para isso, as empresas devem possuir sistemas de informação adequados às mudanças impostas pelo mercado que convivem. O uso destas informações nos sistemas é uma atividade de grande importância, pois somente com um software desenvolvido a partir das necessidades reais de uma empresa, pode-se garantir que as organizações não serão prejudicadas pela falta de informações.

Para acompanhar essas exigências, é primordial que os sistemas em processo de desenvolvimento utilizem técnicas para inspecionar o produto que está sendo desenvolvido. Para isso é importante gerar uma boa documentação ao longo de todo o ciclo de vida do sistema, pela especificação de seus requisitos. Faz-se necessário avaliar o produto do software, ou seja, se os requisitos especificados estão de acordo com as necessidades e expectativas do usuário e da organização.

A preocupação dos engenheiros de software quanto a assegurar a qualidade do produto que está sendo desenvolvido contribuiu para a realização de vários estudos experimentais, bem como suas replicações realizadas por LANUBILE (1996), CIOLKOWSKI (1997), FUSARO (1997) e LAITENBERGER (2001) em inspeções de software que utilizam técnicas de leitura, com o principal propósito de detectar e remover defeitos de um produto, logo no início do seu ciclo de vida.

A partir desse contexto, pretende-se realizar um estudo empírico, para comparar a eficiência das técnicas de inspeção, conhecidas como técnicas de leitura, empregadas na detecção de defeitos de software existentes em um documento de requisitos de software.

1.2. OBJETIVOS

Este projeto de mestrado tem como objetivo estudar o método de inspeção, explorando seu emprego a partir das técnicas de leitura: Leitura Baseada em Cenários (LBCe), Leitura Baseada em *Checklists* (LBCh) e Leitura Baseada em Perspectivas (LBPe), utilizando como material para a inspeção, a análise de um Documento de Requisitos de Software, no padrão proposto pela *IEEE Computer Society ANSI/IEEE* (1998), preparado no início do ciclo de vida de desenvolvimento do sistema.

O entendimento detalhado do método de inspeção e das técnicas de leitura exigiu o levantamento e estudo da bibliografia pertinente, que foi completado pela realização de um estudo empírico para comparar a eficiência das técnicas de leitura aplicadas na inspeção de documentos de requisitos de software.

Os objetivos específicos desse trabalho são:

- Comparar a eficiência das técnicas de leitura em detectar defeitos em produtos de software, especificamente, em documentos de especificação de requisitos;
- Avaliar se existe associação entre a técnica mais eficiente e o nível de experiência dos inspetores; como analistas de sistemas e/ou engenheiros de software.

1.3. IMPORTÂNCIA E JUSTIFICATIVAS DA PESQUISA

Analisando o fato de que as organizações convivem em um mundo competitivo, faz-se necessário uma tomada de decisão concreta e clara para garantir uma posição no mercado. Estes aspectos refletem na necessidade de adaptar os

processos organizacionais, ou seja, os processos de negócio da empresa ao software, tendo como objetivo principal atingir as estratégias da organização (MAIDANTCHIK, 2002). A principal preocupação dos engenheiros de software é criar produtos que atendam às exigências da organização. Mas a especificação dos requisitos dos softwares, na organização, faz com que os engenheiros esbarrem em algumas dificuldades, como:

- Uma das maiores fontes de defeitos em projetos de sistemas é decorrente da comunicação ineficaz entre analistas e usuários. Isso requer uma compreensão das características dos defeitos introduzidos durante as interações entre o analista e o usuário, bem como do processo pelo qual tais defeitos foram introduzidos (VALENTI, 1998);
- Decorrente do problema citado, os requisitos não refletem a necessidade dos usuários para o sistema que está sendo desenvolvido.

O sucesso de um projeto de software em desenvolvimento depende do produto resultante. Para assegurar a qualidade do software, desde o princípio do projeto, os desenvolvedores utilizam a inspeção de software para verificar o produto produzido (especificação de requisitos, projeto, código) e então prevenir um custo exponencial no reparo dos seus defeitos (LANUBILE, 1997; BIFFL, 2001).

O método de inspeção tem contribuído para a obtenção de um software confiável, seguro e que satisfaça as reais necessidades do usuário no seu dia-a-dia.

Estudos realizados nesta área demonstram que, com o uso da inspeção, os produtos de software passaram a ter uma qualidade melhor do que os sistemas desenvolvidos anteriormente, ajudando na produtividade de sistemas novos e verificando se os requisitos especificados no documento do software estão sendo atendidos. Os benefícios das inspeções consistem em melhorar a produtividade do novo produto, melhorar a comunicação entre os especialistas e usuários e diminuir o esforço econômico (BRIAND, 2000).

1.4. RESULTADOS E CONTRIBUIÇÕES ESPERADOS

Este trabalho, por meio dos estudos e da prática, traz a contribuição de um entendimento detalhado sobre técnicas de leitura na inspeção dos produtos de software. Outras contribuições importantes são os resultados obtidos na análise comparativa das técnicas de leitura para inspecionar documentos de requisitos de software.

1.5. ORGANIZAÇÃO DO TRABALHO

O restante deste trabalho está organizado conforme descrito a seguir.

No Capítulo 2 são apresentados os conceitos que envolvem o estudo, ou seja, a revisão bibliográfica do assunto abordado na dissertação, como os conceitos e objetivos da inspeção de software, o processo de realização da inspeção, a equipe de inspeção, as contribuições e as dificuldades com o uso da inspeção; a importância da Engenharia de Requisitos de Software com ênfase no Documento de Requisitos de Software (DRS), defeitos encontrados em Requisitos de Software e as classificações dos defeitos.

O Capítulo 3 apresenta as técnicas de leitura existentes para auxiliar o processo de inspeção e alguns estudos empíricos realizados em inspeção de requisitos de software, com os principais resultados alcançados.

No Capítulo 4 é descrito o objetivo principal do estudo empírico, bem como as metodologias utilizadas para a obtenção dos resultados e os formulários e documentos utilizados no estudo.

O Capítulo 5 apresenta os resultados e análise das hipóteses formuladas no estudo empírico e os resultados obtidos.

O Capítulo 6 apresenta as conclusões do estudo empírico realizado. Após este capítulo, estão listadas as referências bibliográficas que embasaram a dissertação. Por fim, é apresentado o Apêndice A, “Documentos da Inspeção”, que traz os documentos do sistema PCMSO, Instruções do Estudo, Anotação de Defeitos, Questionário, Relatório de Participantes e Respostas Questionário,

utilizados no estudo, descrito nos capítulos 4 e 5. O Apêndice B, “Técnicas de Inspeção”, descreve as técnicas de inspeção LBCh, LBCe e LBPe, adotadas no estudo descrito no capítulo 4.

CAPÍTULO 2

INSPEÇÃO DE SOFTWARE

2.1 PROCESSO DE INSPEÇÃO

2.1.1 CONCEITOS

O processo de inspeção de software foi definido originalmente por Michael E. Fagan, em 1972, na IBM, Kingston, NY (FAGAN, 1986). Este processo consiste na subdivisão de uma tarefa em diversas atividades relacionadas com a transformação das informações de entrada em informações de saída, de forma organizada e que possa ser repetida (NUSEIBEH, 2002).

A inspeção é considerada como um dos melhores métodos para detectar defeitos nos diferentes subprodutos do processo de desenvolvimento de software, principalmente no Documento de Requisitos de Software (DRS), conforme destacado por LANUBILE (1996), BIFFL (2001) e WONG (2002). Inicialmente, a inspeção foi criada para inspecionar código-fonte de software (LANUBILE, 1996). Atualmente, o método é utilizado em todas as fases do ciclo de vida do software.

As principais características de um processo de inspeção de software, segundo FAGAN (1976), FAGAN (1986), DOOLAN (1992) e LANUBILE (1997) são:

- É um método estruturado para verificar documentos do software, tais como: especificação de requisitos, documentos do projeto, código-fonte, etc;
- É realizado em seis etapas consecutivas: Planejamento, Visão Geral, Preparação, Inspeção, Retrabalho e Revisão;

- Compreende a definição de uma equipe de inspeção para verificar o documento, que é composta pelo moderador, relator, inspetor e autor;
- Os participantes que irão inspecionar o produto têm suas funções bem definidas durante o processo, cada qual cumprindo a sua tarefa;
- O resultado final é gerar um documento do que foi obtido na inspeção individual e nas discussões das reuniões das equipes.

Pode-se concluir que a inspeção é um método de análise estruturado, eficiente e econômico, para verificar as propriedades de qualidade do produto de software que está sendo desenvolvido, tendo como principal objetivo minimizar os custos com o reparo de defeitos nos projetos (FAGAN, 1976; FAGAN, 1986; LANUBILE, 1997).

A inspeção não é aplicável somente na validação de especificações de requisitos de software, mas também pode ser usada em outros subprodutos, tais como: documentos de planejamento, documentação do usuário, projeto, código, testes, cobrindo praticamente todas as fases do desenvolvimento do sistema (DOOLAN, 1992).

O método de inspeção tem tido a contribuição de muitos autores, alguns estudos realizados abordam o estudo da inspeção para melhorar sua eficiência na detecção de defeitos realizados por CIOLKOWSKI (1999), BRIAND (2000), SAUER (2000) e BIFFL (2001); outros trabalhos envolvem o estudo, a melhoria e desenvolvimento de técnicas de leitura para auxiliarem os inspetores na detecção de defeitos durante a inspeção como LANUBILE (1996) e CIOLKOWSKI (1999) e estendendo o uso do método de inspeção com estudos realizados em projetos de orientação a objetos por SHULL (1998) e TRAVASSOS (1999).

2.1.2 OBJETIVOS DA INSPEÇÃO

O principal objetivo da inspeção é assegurar a qualidade do software que está sendo desenvolvido. É um método padrão, tendo como propósito descobrir os defeitos no projeto, utilizando o código do programa ou uma especificação do

sistema, antes do passo seguinte do processo de engenharia de software (PRESSMAN, 1995; TYRAN, 2002). A inspeção tem auxiliado na redução de custos do projeto, economizando no processo de desenvolvimento do software.

A inspeção assegura os seguintes objetivos na sua utilização, de acordo com FAGAN (1986) e PRESSMAN (1995):

- Melhorar a qualidade do produto de software que está sendo construído, atendendo às necessidades do usuário;
- Aumentar a produtividade durante a criação do sistema que está sendo desenvolvido, garantindo a sua qualidade e prevenindo futuros problemas, detectando os defeitos;
- Verificar se o software atende a todos os seus requisitos;
- Tornar os projetos mais administráveis.

O processo de inspeção de documentos de software é esquematizado na Figura 2.1. Conforme descrito em LAITENBERGER (1995) e ABIB (1998), a verificação de um documento de software inicia-se com a leitura individual do documento pelos avaliadores do processo de inspeção. Após a leitura do documento, há uma reunião em que os avaliadores o analisam para discutir os defeitos que encontraram durante o processo de leitura e encontrar mais defeitos. Durante a reunião, todos os defeitos são coletados. O próximo passo é removê-los e então iniciar o ciclo novamente, para obter a qualidade desejada no documento do software. Este ciclo pode ser repetido dependendo do método ou da qualidade desejada.

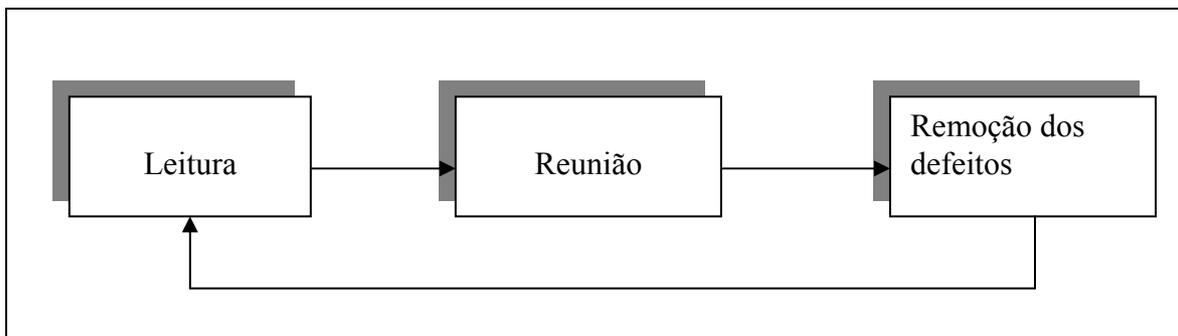


Figura 2.1 - Processo de Verificação de Documentos de Software

Os aspectos usados para classificar e caracterizar as técnicas de inspeção, conforme LAITENBERGER (1995) e ABIB (1998), são:

- **Estratégia.** É uma das características da técnica de inspeção que determina como a procura por defeitos deve ser realizada, nos documentos de software. Existem várias estratégias para detectar defeitos nesses documentos, entre elas, temos:
 - **Técnicas sistemáticas:** o avaliador recebe um treinamento de como ler o documento de software que será avaliado, definindo um procedimento claro no processo de inspeção.
 - **Técnicas não sistemáticas:** o avaliador não recebe nenhum treinamento ou diretriz sobre como ler o documento a ser inspecionado, não possuindo um procedimento claro a ser seguido na inspeção.
- **Responsabilidade dos Avaliadores.** Visa definir a responsabilidade de cada avaliador em encontrar os tipos de defeitos no documento do software a ser avaliado. Dentre as técnicas existentes podemos citar:
 - **Técnicas específicas:** os avaliadores procuram os defeitos a partir de uma classe de defeitos existente.
 - **Técnicas gerais:** os avaliadores procuram por qualquer tipo de defeitos, não se importando com a classe a qual pertencem.
- **Política de Responsabilidades.** É a característica que determina a responsabilidade de cada avaliador durante o processo de inspeção, dividida em:
 - **Técnicas distintas:** cada avaliador tem a sua responsabilidade, sendo que a sobreposição de responsabilidades é pouca ou nula.
 - **Técnicas idênticas:** não há divisão das responsabilidades entre os avaliadores que participam da inspeção, todos têm as mesmas responsabilidades.

Essas características, que foram apresentadas de acordo com a estratégia, a responsabilidade dos avaliadores e a política de responsabilidades, têm como objetivo determinar, para os avaliadores que participam da inspeção, os procedimentos na leitura do documento de software a ser inspecionado, os tipos de defeitos que deverão ser detectados e a responsabilidade que cada membro da equipe deverá assumir durante a inspeção.

2.1.3 PROCESSO DE REALIZAÇÃO DA INSPEÇÃO

A inspeção é um dos principais métodos para garantir a qualidade no produto de software, segundo AMBRUST (2002). Em todas as fases de desenvolvimento do software, podem ser introduzidos defeitos no produto que está sendo criado, e a inspeção visa identificar tais defeitos para poder removê-los.

O processo de inspeção inclui seis etapas principais (FAGAN, 1986; EBENAU, 1994; ABIB, 1998; TYRAN, 2002; WONG; 2002), cada qual com objetivos específicos:

a) Planejamento. Determina se os materiais que serão inspecionados são satisfatórios, organiza as pessoas que irão participar da inspeção e o local onde serão realizadas as sessões de inspeção.

b) Visão Geral. Inclui a apresentação do material a ser inspecionado aos participantes e a atribuição de funções a cada participante durante a inspeção.

c) Preparação. Nesta etapa, os participantes são treinados para executarem as funções que lhes foram atribuídas, visando encontrar os defeitos constantes do produto ou artefato de software.

d) Realização da Inspeção. Inclui sessões de trabalho, nas quais os participantes analisam o produto ou artefato de software enfocado, com o fim de detectar os defeitos existentes nesses produtos.

e) Retrabalho. Nesta etapa, os defeitos detectados, devidamente documentados, são encaminhados ao autor do produto que foi inspecionado, para que seja providenciada a remoção destes defeitos.

f) Revisão. Finalmente, o autor confere o produto revisado, juntamente com a equipe de inspeção, para assegurar-se de que todas as correções necessárias foram realizadas e que nenhum defeito novo foi introduzido.

Uma visão geral das fases do processo de inspeção é apresentada na Figura 2.2.

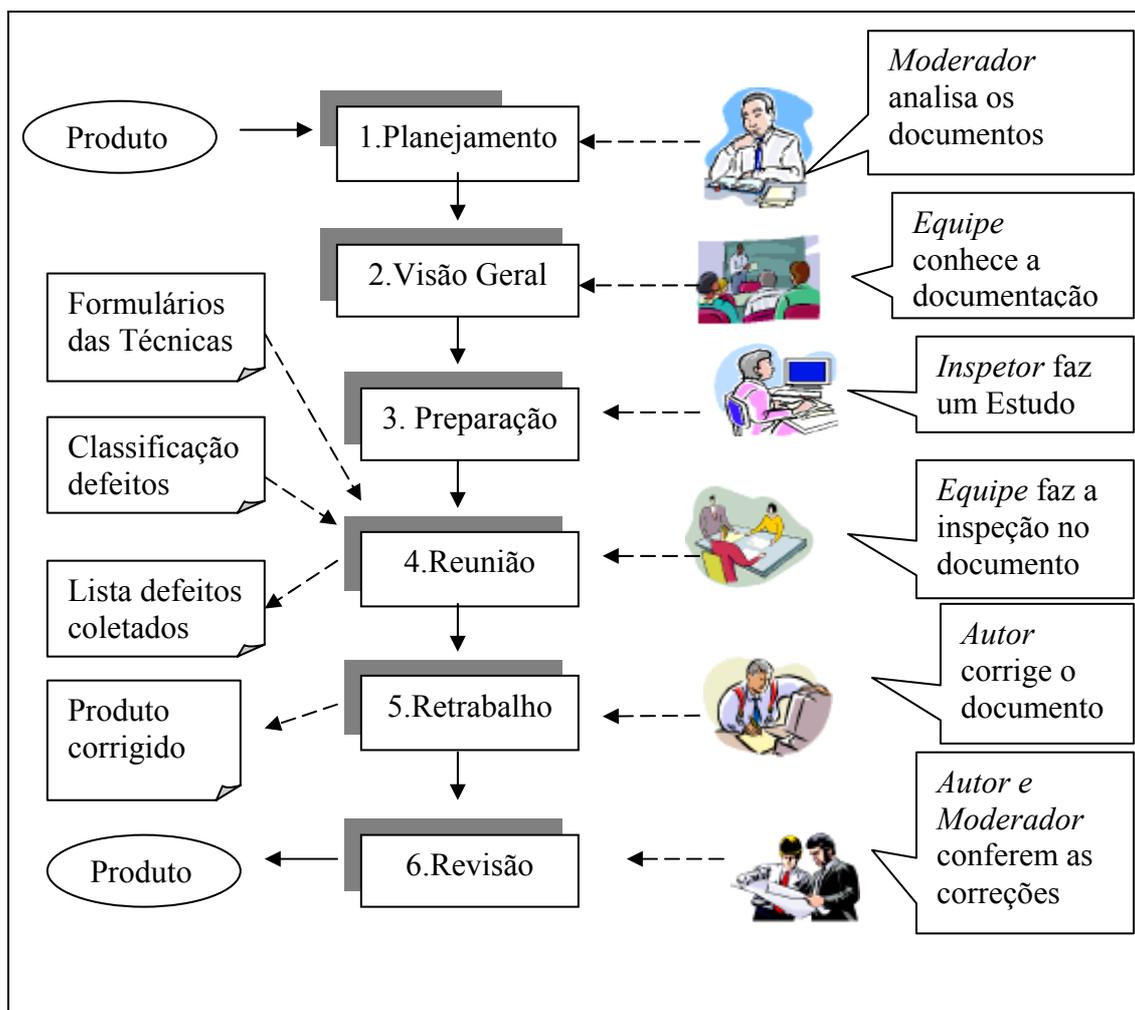


Figura 2.2 - Fases do Processo de Inspeção

Durante as etapas de planejamento e visão geral, um membro da equipe de inspeção envia o material para os outros participantes e faz um cronograma para as próximas etapas. Na preparação, cada inspetor da equipe, individualmente entende e revisa o documento para encontrar os defeitos. Na reunião, todos os inspetores encontram-se para coletar e discutir os defeitos das revisões individuais e mais adiante revisar o documento para encontrar defeitos adicionais. No retrabalho, o autor revisa o documento para corrigir os

defeitos. Finalmente, na revisão o retrabalho é conferido com a equipe de inspeção, assegurando que todos os defeitos foram solucionados e nenhum foi introduzido no material inspecionado.

2.1.4 EQUIPE DE INSPEÇÃO

Um dos fatores relevantes no processo de detecção de defeitos na inspeção de um software é a função dos membros da equipe envolvida. De acordo com FAGAN (1976), FAGAN (1986), DOOLAN (1992), EBENAU (1994) e CIOLKOWSKI (1999), uma inspeção envolve quatro funções, descritas abaixo:

a) Moderador. É a função principal para o sucesso da inspeção. O moderador deve ter os seguintes atributos: ser um profissional competente (programador ou analista), não sendo necessário ser um especialista técnico no documento que será inspecionado. Para preservar a objetividade e reforçar a integridade da inspeção, é muito importante usar um moderador não relacionado ao projeto. Este deve gerenciar e liderar a equipe, dando suporte a todos os membros que a compõem. Dentre as funções do moderador incluem-se também: a divulgação das reuniões, o resultado do relatório de inspeção e o acompanhamento das etapas do retrabalho e da revisão.

b) Autor. É o responsável pela criação do documento do projeto a ser inspecionado.

c) Relator. O responsável pela redação do relatório dos defeitos encontrados no projeto na reunião de inspeção.

d) Inspetor. É o responsável pela execução dos testes, ou seja, testar o produto do autor.

Ao longo da inspeção a equipe atua de maneira inter-relacionada, desempenhando papéis específicos. Na etapa de planejamento, o moderador analisa se o material a ser inspecionado é satisfatório para a inspeção, define o grupo de participantes e a função de cada um e agenda as reuniões. Na etapa da visão geral, o autor descreve com detalhes o documento a ser inspecionado para todos os membros da equipe. Na etapa da preparação, os inspetores

iniciam uma leitura individual do documento a ser inspecionado, podendo encontrar alguns defeitos que serão classificados de acordo com o seu respectivo tipo. Na etapa de reunião, um autor é selecionado pelo moderador para descrever como o projeto será implementado, e o moderador com a ajuda do relator cria um relatório de inspeção, com a classificação dos defeitos, que será enviado para a próxima etapa. Na etapa de retrabalho, o autor refaz o seu trabalho removendo os defeitos encontrados na etapa anterior usando o relatório de inspeção. Na revisão, é função do moderador verificar se todos os defeitos descobertos na operação de inspeção foram solucionados pelo autor do projeto.

O Quadro 2.1 resume as funções e objetivos da equipe no processo de inspeção.

Quadro 2.1 - Funções e Objetivos da Equipe no Processo de Inspeção

Operações do Processo	Objetivos da Operação	Membro da Equipe
1. Planejamento	Conhecimento do projeto.	Moderador
2. Visão Geral	Conhecimento da Documentação.	Todos Membros da Equipe
3. Preparação	Leitura da Documentação.	Todos Membros da Equipe
4. Reunião	Coletar os defeitos.	Todos Membros da Equipe
5. Retrabalho	Solucionar os defeitos encontrados no projeto.	Autor e Moderador
6. Revisão	Verificar se os defeitos foram solucionados.	Moderador

2.1.5 DIFICULDADES DA INSPEÇÃO

As dificuldades relatadas estão relacionadas a diversos aspectos que envolvem uma inspeção de software, tais como:

- O moderador que lidera o processo de inspeção não foi bem treinado (FAGAN, 1986);
- As pessoas que compõem a equipe de inspeção não assumem as suas tarefas;
- Dificuldade dos participantes entenderem o uso das técnicas de inspeção durante o treinamento (LANUBILE, 1996);

- O tempo disponível para detectar os defeitos é insuficiente, acarretando um projeto mal inspecionado (LANUBILE, 1996);
- Falta de conhecimento por parte dos participantes com relação ao domínio do projeto (LANUBILE, 1996);
- Para a maioria das organizações a produtividade ainda é vista e medida em termos de linhas de código, sendo que as inspeções requerem esforço sem produzir qualquer linha de código, não sendo vista como um benefício tangível (BRIAND, 2000);
- Para os desenvolvedores, é constrangedor avaliar o seu trabalho e deixá-lo ser avaliado por outros, pois pode comprometer a sua reputação profissional (O'NEIL, 1997).

2.1.6 CONTRIBUIÇÕES DA INSPEÇÃO

Uma inspeção de boa qualidade resulta em contribuições de qualidade para o produto que está sendo desenvolvido e auxilia na produtividade do software. Abaixo estão relacionados alguns itens com respeito à contribuição para a qualidade do software.

- A inspeção auxilia a detectar os problemas do produto no início do ciclo de desenvolvimento do software, evitando o alto custo de retrabalho na manutenção do sistema (BOEHM, 2001);
- O treinamento de pessoal inexperiente nas técnicas de inspeção é fácil de ser realizado, além de melhorar a comunicação entre a equipe que desenvolve o projeto e promove melhorias contínuas no decorrer da inspeção (BOEHM, 2001);
- A aplicação da inspeção nos projetos de software permite aos desenvolvedores ter dupla visão do projeto original; de como este projeto é, com seus defeitos e como ele ficará após a revisão do processo de inspeção (BOEHM, 2001);

- Alguns defeitos que afetam a confiabilidade de um sistema podem ser solucionados com a inspeção, garantindo um alto nível de remoção dos defeitos, com eficiência, na busca dos defeitos (BOEHM, 2001);
- As inspeções reduzem o tempo gasto no retrabalho do software que está sendo desenvolvido, permitindo aos desenvolvedores um tempo maior para desenvolver outros projetos (BOEHM, 2001);
- Um software que utiliza as técnicas de detecção de defeitos tem de 60% a 90% de todos os seus defeitos encontrados (FAGAN, 1986);
- O produto inspecionado permite produzir um *feedback* entre os desenvolvedores do software, evitando inserir defeitos em trabalhos futuros (FAGAN, 1986);
- A inspeção auxilia os engenheiros de software a remover uma série de defeitos operacionais e a minimizar os custos do produto (JONES, 1996);
- O processo de inspeção promove a equipe de trabalho, e é um meio de desenvolver uma série de princípios para todo o grupo (DOOLAN, 1992);
- Os inspetores identificam a causa dos defeitos detectados na raiz, propondo modificações no processo de desenvolvimento do software e prevenindo a ocorrência de defeitos similares em projetos futuros (DOOLAN, 1992).

2.2 ENGENHARIA DE REQUISITOS DE SOFTWARE

A Engenharia de Software é composta por três elementos principais: métodos, ferramentas e procedimentos (PRESSMAN, 1995). A disciplina Engenharia de Software orienta o desenvolvedor no processo de desenvolvimento do software, indicando os procedimentos a serem tomados para a construção de um produto de software com qualidade. Os métodos indicam “como fazer” para construir o software e envolvem o planejamento do projeto, a análise de requisitos do software, a codificação, entre outras atividades. As ferramentas

compreendem um apoio automatizado ou semi-automatizado aos métodos. Os procedimentos compreendem os métodos e as ferramentas. O ciclo de vida consiste em uma metodologia para auxiliar o Engenheiro de Software no desenvolvimento do sistema.

A Engenharia de Requisitos é uma das primeiras etapas do ciclo de vida do software e compõe-se das fases de Análise do Sistema e Análise de Requisitos do Software. A Análise do Sistema visa o entendimento do problema a ser solucionado, realizando a coleta dos requisitos em nível de sistema, tais como hardware e pessoas, apresentando uma análise de alto nível do sistema. Nesta fase um Plano de Projeto de Software é produzido e revisado pelo gerente do projeto. A Análise de Requisitos de Software é a fase da coleta e especificação dos requisitos do software, envolvendo uma descrição completa do sistema, bem como sua função, desempenho e interface do sistema. Estes requisitos, tanto do software quanto do sistema, são documentados e revistos pelo desenvolvedor, com a participação do usuário final ou cliente (PRESSMAN, 1995).

Um dos primeiros documentos da Engenharia de Requisitos é o Documento de Requisitos do Software (DRS). A Figura 2.3 ilustra as Etapas da Criação deste documento.

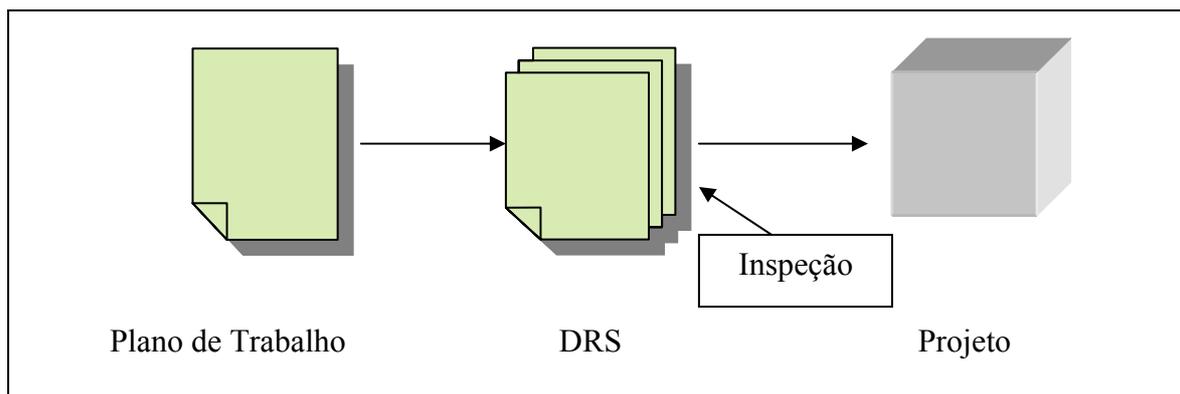


Figura 2.3 - Etapas da Criação do DRS

O modelo de processo de Engenharia de Requisitos (LOPES, 2002) mais usado destaca quatro atividades, conforme mostra o Quadro 2.2.

Quadro 2.2 - Processo de Engenharia de Requisitos

Atividade	Descrição
Elicitação de Requisitos	É a atividade para a identificação dos requisitos do sistema a partir do grupo de usuários, documentos de domínio e pesquisas de mercado.
Análise de Requisitos e Negociação	Os requisitos são analisados por grupo de usuários, devido a limitações como orçamento, prazo, inconsistências e diferenças de perspectivas entre os grupos.
Documentação de Requisitos	Os requisitos são documentados e formalizados num Documento de Requisitos e deverá ser entendido por todos os usuários.
Validação de Requisitos	Avaliação dos requisitos, com ênfase na sua consistência e completitude: são identificados os possíveis problemas nos requisitos, antes de se criar o documento que servirá de base para o desenvolvimento do sistema.

A Figura 2.4, a seguir, representa as tarefas a serem realizadas nas atividades descritas no Quadro 2.2 do Processo de Engenharia de Requisitos, visto que as atividades de Documentação e Validação de Requisitos exigem várias iterações até que seja aprovado na etapa da validação (LOPES, 2002).

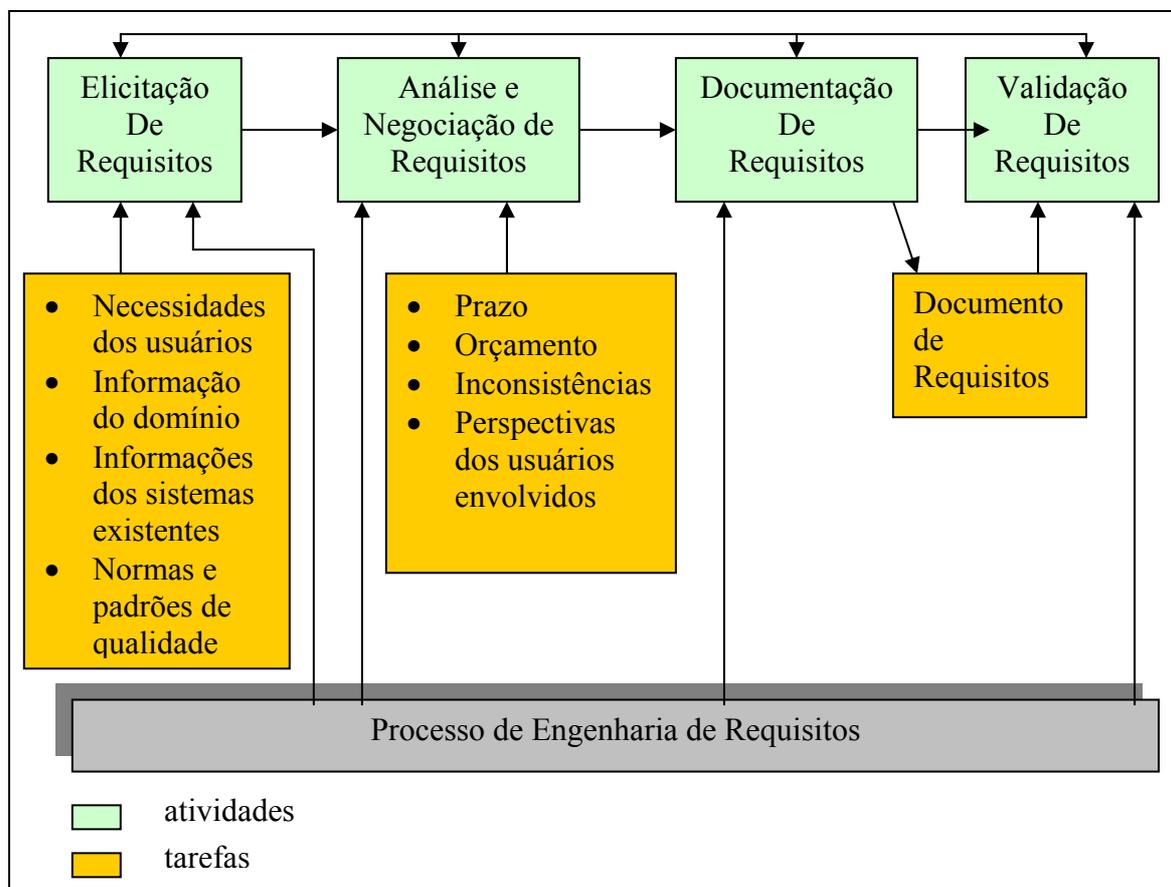


Figura 2.4 - Processo de Engenharia de Requisitos LOPES (2002)

Na verdade, a área de Engenharia de Requisitos vem sofrendo muitas alterações e estudos práticos de conhecimento científico estão sendo realizados para solucionar os problemas enfrentados, tais como os descritos a seguir (ZAVE 1997; VALENTI, 1998; MAIDANTCHIK, 2002; NUSEIBEH, 2002):

- A dificuldade do usuário em definir os requisitos desejados;
- A dificuldade dos analistas em especificar e extrair do usuário os requisitos do sistema;
- A existência de um conjunto de requisitos de difícil adequação ao mundo real da organização.

A Engenharia de Software destaca a importância do controle na condução do processo de construção do software para garantir-lhe a qualidade (PRESSMAN, 1995). A partir dos diferentes modelos de ciclo de vida do software como o cascata, o protótipo entre outros, é importante entender a transição de uma atividade para outra, pois todos os ambientes produzem um documento ao final de cada fase, caracterizando o final de uma fase e o início da próxima. Cada um desses documentos produzidos em cada fase representa o software em um nível de abstração mais baixo, incluindo mais e mais detalhes, até alcançar a representação do código fonte na linguagem utilizada. O ciclo de vida representado na Figura 2.5 é o modelo cascata, conhecido também como modelo clássico, com as principais fases de desenvolvimento e a documentação gerada no final de cada fase para ser utilizada na inspeção.

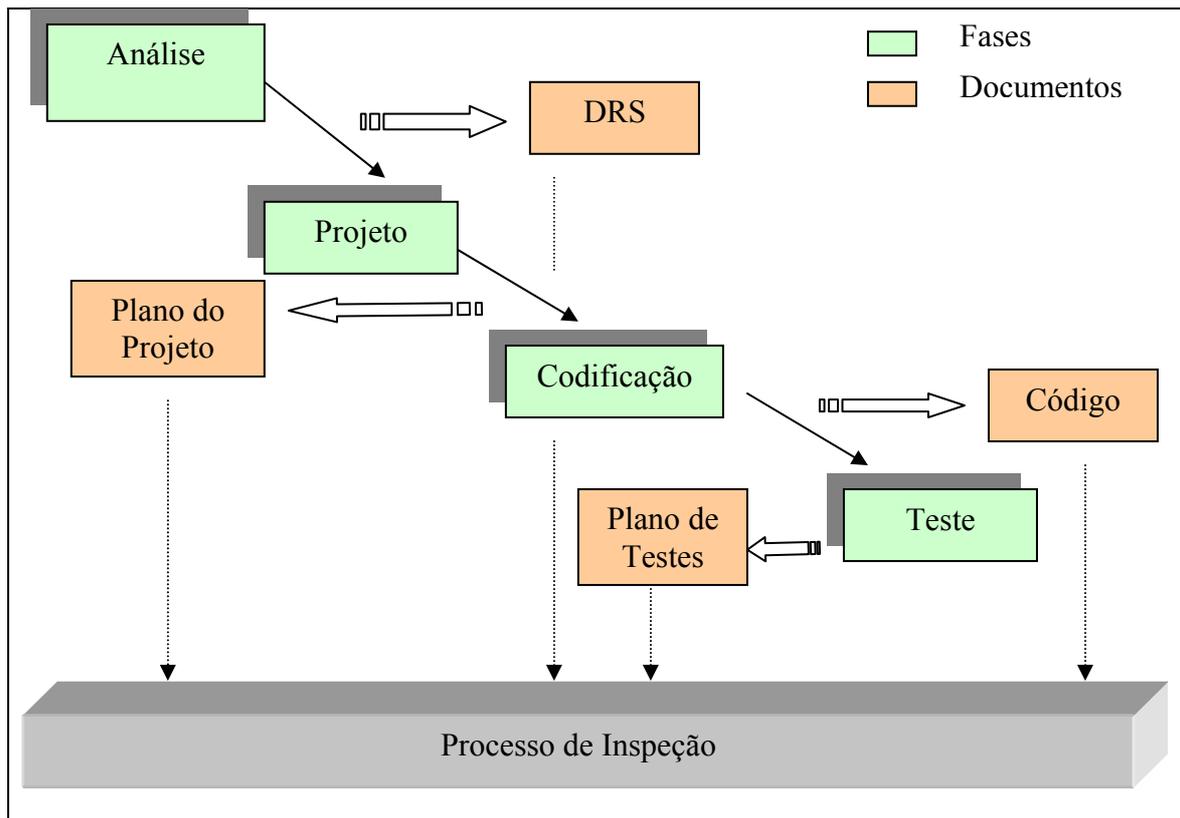


Figura 2.5 - Modelo Cascata no Processo de Inspeção

No processo de inspeção, o modelo cascata pode requerer uma tarefa de análise no término de cada fase do ciclo de vida, para detectar o maior número de defeitos possíveis nos documentos produzidos, antes de serem usados nas fases posteriores (SHULL, 1998). Em resumo, o processo de inspeção garante a qualidade do produto de software para cada fase do ciclo de vida, auxiliando o desenvolvedor a solucionar as dificuldades citadas acima. Durante a fase de engenharia de requisitos, o processo de inspeção assegura que no próximo projeto, os defeitos não serão cometidos novamente, lembrando que o documento de especificação de requisitos é a base fundamental para todas as outras fases e para a qualidade do sistema.

2.2.1 DOCUMENTO DE REQUISITOS DE SOFTWARE

Uma etapa importante no desenvolvimento do software é a criação do Documento de Requisitos de Software (DRS). Esse documento é chamado também de Documento de Especificação de Requisitos de Software, pela *IEEE Computer Society* (ANSI/IEEE, 1990), e representa a funcionalidade esperada

de um sistema de software para dar início ao seu desenvolvimento. Segundo Laudon (1999), o DRS deve incluir medidas de qualidade para que o sistema possa ser avaliado enquanto está sendo desenvolvido e também quando estiver finalizado. Como o cliente participa da criação desse documento, normalmente ele é utilizado como um acordo, uma forma de contrato com o cliente e a empresa que o está desenvolvendo.

É imprescindível avaliar a qualidade do DRS para evitar problemas futuros durante o processo de desenvolvimento do software.

Um bom DRS deve cumprir os seguintes fatores de qualidade: corretude, não ambigüidades, completude, consistência, verificabilidade, modificabilidade, rastreabilidade, segundo *IEEE Computer Society* (ANSI/IEEE Std 1012, 1998), e essencialidade, segundo *IEEE Computer Society* (ANSI/IEEE Std 830, 1998). As definições seguem abaixo:

- Corretude. Um DRS está correto somente se os requisitos declarados no documento estão sintática e semanticamente corretos. Um requisito está semanticamente correto se o documento do sistema está de acordo com a descrição do problema a ser solucionado; ou seja, o sistema satisfaz sua especificação e atinge as necessidades do cliente PRESSMAN (1995). Um requisito está sintaticamente correto se está de acordo com os padrões do documento (quais seções deveriam ter o documento de requisitos e como os requisitos deveriam ser formulados).
- Não Ambigüidades. Um DRS não é ambíguo somente se todos os requisitos declarados possuem uma única interpretação, ou seja, algum requisito foi definido de forma que possa causar confusão.
- Completude. Um DRS está completo somente se contém todos os requisitos principais para todas as situações que serão realizadas. Qualquer requisito declarado num outro documento de um sistema tem que ser tratado no DRS. Completude é um dos problemas principais na descrição do Documento, porque os clientes normalmente trabalham em áreas diferentes dos desenvolvedores e dispõem de procedimentos e

funcionalidades que não são óbvias aos desenvolvedores, gerando dificuldades na criação do Documento.

- **Consistência.** Um DRS está consistente se nenhum dos requisitos individuais do subconjunto definido está em conflito.
- **Verificabilidade.** Um DRS é verificável se para cada um dos requisitos contidos no documento existir um processo finito e economicamente viável pelo qual uma pessoa ou máquina possa assegurar que o produto de software atenda ao requisito.
- **Modificabilidade.** Um DRS é modificável somente se a sua estrutura e estilo forem adaptáveis a qualquer mudança nos requisitos do sistema, mantendo a estrutura e o estilo.
- **Rastreabilidade.** Um DRS é rastreável somente se a origem de cada um de seus requisitos está clara e for fácil de referenciar a um requisito num desenvolvimento futuro. A rastreabilidade é importante quando os requisitos declarados mudarem durante o processo de desenvolvimento do sistema; ou quando for encontrado um defeito no DRS; ou quando o cliente quiser mudar ou acrescentar requisitos. Neste caso as mudanças no documento devem ser facilmente rastreadas em documentos posteriores (por exemplo, através da numeração dos requisitos).
- **Essencialidade.** Um DRS é essencial somente se todo requisito declarado apresenta o que se pretende com ele. Um requisito é essencial somente se este está relacionado com o que é necessário para o sistema especificando na descrição dos requisitos a funcionalidade do sistema, ou seja, o que o sistema faz.

Entre os padrões existentes para a preparação do DRS destaca-se o padrão proposto pela *IEEE Computer Society* (ANSI/IEEE Std 830, 1998) e que será adotado neste projeto. Este padrão define uma visão geral básica que um DRS deve apresentar, além dos tópicos específicos formulados para o sistema a ser desenvolvido.

O Quadro 2.3 abaixo apresenta um DRS dividido em uma série de seções e subseções listadas a seguir.

Quadro 2.3 - Documento de Requisitos de Software

Seção / Subseção	Descrição
1. Introdução	Fornece uma visão de todo o DRS.
1.1. Objetivo	Descreve o objetivo de DRS e especifica o que se pretende com ele.
1.2 Escopo	Identifica os produtos de software a serem gerados, explica o que os produtos identificados farão, a aplicação do software e seus benefícios.
1.3. Definições de Acrônimos e Abreviações	Fornece as definições de todos os termos, acrônimos e abreviações usadas no DRS.
1.4. Referências	Fornece uma lista completa de todos os DRSS referenciados ao longo do DRS.
1.5. Visão Geral	Descreve o conteúdo das seções relevantes e explica como o DRS será organizado.
2. Descrição Geral	Descreve os fatores gerais que afetam os produtos e seus requisitos.
2.1. Descrição do Produto	Especifica os produtos dentro da perspectiva de outros produtos ou projetos relacionados.
2.2. Funções do Produto	Fornece um resumo das funções que o software executará.
2.3. Características do Usuário	Descreve as características gerais dos eventuais usuários do produto.
2.4. Restrições Gerais	Fornece uma descrição geral de quaisquer outros itens que possam limitar as opções do desenvolvedor durante o projeto do sistema.
2.5. Pressupostos e Dependências	Lista cada um dos fatores que afetam os requisitos declarados no DRS.
3. Requisitos Específicos	Contêm todos os detalhes que o projetista de software necessita para projetar o sistema.
3.1. Informação Necessária como parte dos Requisitos Específicos	Descreve os dados dos produtos de software.
3.1.1. Requisitos Funcionais	Especifica como as entradas do produto de software devem ser transformadas em saídas.
3.2. Requisitos de Desempenho	Especifica os requisitos numéricos, estáticos e dinâmicos, localizados no software ou na interação do software com o homem.
3.3. Restrições de Projeto	Descreve as limitações do projeto impostas por outros padrões, limitações de hardware, etc.
3.3.1. Padrões a serem atendidos	Especifica os requisitos derivados de padrões ou regras existentes.
3.3.2. Limitações de Hardware	Inclui os requisitos para o software operar dentro das limitações do hardware.
3.4. Atributos	Descreve os atributos que podem especificar requisitos do software.
3.4.1. Disponibilidade	Especifica os fatores necessários para garantir um nível de disponibilidade definido para todo o sistema.
3.4.2. Segurança	Especifica os fatores necessários para proteger o software de acessos indevidos.

Quadro 2.3 - Continuação Documento de Requisitos de Software

3.4.3. Manutenção	Especifica os fatores assegurando que o software é passível de manutenção.
3.4.4. Transferência e Conversão	Especifica os procedimentos do usuário, as limitações de interface, necessários para transportar o software de um meio a outro.
3.5. Requisitos de Interface Externa	Descreve as interfaces do sistema.
3.5.1. Interface do Usuário	Especifica o formato da tela requerida, o layout da página e o tempo relativo de entradas e saídas.
3.5.2. Interface de Hardware	Especifica as características entre o produto de software e os componentes de hardware do sistema.
3.5.3. Interface de Software	Especifica o uso de outros produtos de software necessários e interfaces com outros sistemas de aplicação.
3.5.4. Interface de Comunicação	Especifica as interfaces de comunicação, tais como, protocolos de redes locais, etc.
3.6. Outros Requisitos	Descreve outros requisitos do sistema.
3.6.1. Banco de Dados	Especifica os requisitos para qualquer banco de dados que tenha sido desenvolvida como parte de um produto.
3.6.2. Operações	Especifica as operações normais e especiais requeridas pelo usuário.
3.6.3. Requisitos de Adaptação	Define os requisitos para quaisquer dados ou seqüência de inicialização que sejam especificados para um dado.
3.7. Organização dos Requisitos Específicos	É a maior e a mais completa do DRS e seu objetivo é especificar adequadamente os requisitos de maneira legível.
4. Informação de Suporte	Contêm a tabela de conteúdo, os apêndices e o índice, que facilitam o uso do DRS.

2.2.2 DEFEITOS EM REQUISITOS DE SOFTWARE

No contexto de requisitos de software, é considerado como defeito qualquer fato no documento que possa causar um comportamento diferente do esperado. Então, os defeitos são representados por qualquer aspecto no documento de requisitos que não cumpre os fatores de qualidade mencionados. Ou seja, qualquer ambigüidade, falta ou inconsistência representa um defeito no documento de requisitos. Estudos apontam que 56% dos defeitos de um software podem ser identificados durante a etapa de engenharia de requisitos (FAGAN, 1986). As seguintes considerações são relevantes, a respeito dos defeitos em requisitos de software:

- Quanto mais tarde um defeito é detectado, maior o custo de corrigi-lo;

- Muitos defeitos são inseridos durante a elicitação e definição de requisitos;
- Muitos defeitos nos requisitos podem ser detectados cedo no ciclo de desenvolvimento;
- Defeitos típicos incluem fatos incorretos, omissões, inconsistências e ambigüidades.

Neste trabalho serão considerados os defeitos que causam falhas (conseqüências) na funcionalidade do sistema, capazes de ocasionar fracassos no sistema.

2.3 CLASSIFICAÇÃO DE DEFEITOS

Para entender os tipos de defeitos que ocorrem em subprodutos ou artefatos de software, foram desenvolvidas várias pesquisas criando esquemas de classificação de defeitos. As classificações de defeitos podem ser úteis para comparações com outros estudos ou ambientes. Os esquemas agrupam os defeitos que acontecem em classes de um ambiente particular. A classificação de defeitos depende da fase do ciclo de vida na qual a inspeção irá ocorrer. Vários autores propuseram diferentes esquemas de classificações de defeitos; alguns são dirigidos a documentos específicos do ciclo de vida do software, como defeitos no documento de requisitos, ou ainda defeitos em código fonte. De acordo com CARVER (2002), os defeitos são classificados em esquemas de: classificação de defeitos para requisitos, classificação de defeitos por projeto, classificação de defeitos por código.

2.3.1 CLASSIFICAÇÃO DE DEFEITOS NO DOCUMENTO DE REQUISITOS DE SOFTWARE

No estudo realizado por BASILI (1981), os autores descrevem a evolução de um documento de especificação de requisitos para um programa de voo operacional. Os defeitos são representados como defeitos reais elaborados a partir da evolução do documento. Os autores definiram um nível de esquema

de classificação com sete categorias. Estas classes foram usadas para elaborar as hipóteses sobre os tipos de defeitos que eram mais comuns.

O Quadro 2.4 representa a classificação das classes.

Quadro 2.4 - Classificação de Defeitos para Documentos de Software

Categoria	Descrição
De digitação	Problemas relativamente simples no documento de requisitos
Ambigüidade	Algo no documento que possui mais de um significado
Omissão	Algo que foi omitido no documento
Inconsistência	Duas partes no documento são incompatíveis com uma ou outra
Fato Incorreto	Algo no documento está incorreto com respeito ao domínio
Informações em Seções Erradas	Informações incluídas no documento em seção errada
Fato de Implementação não Fornecido	Informação necessária para o propósito da implementação não foi fornecido
Outros	Defeitos que não fazem parte das outras classes

No estudo de ACKERMAN (1989) sobre os benefícios que foram auferidos aplicando-se inspeções ao longo do ciclo de desenvolvimento do software, discutiu-se um esquema de classificação de defeitos encontrados em documentos de especificação de requisitos. Os autores declaram que as inspeções podem ser utilizadas nos documentos de requisitos, projeto, código, planos de teste e especificações de teste, mas só definiram a classificação para documentação de requisitos. O esquema de classificação definido pelos autores apresenta dois níveis, mas não foi dada a razão para tal. Na verdade o primeiro nível representa as informações da organização e o segundo nível é representado por perguntas mais específicas para a descoberta dos defeitos.

O Quadro 2.5 representa o esquema de classificação de defeitos para requisitos.

Quadro 2.5 - Classificação de Defeitos para Requisitos de Software

Nível	Perguntas Específicas
Completeness	1. Todas as fontes de entrada foram especificadas?
	2. Qual é a entrada principal?
	3. Existem conjuntos "sem cuidados" na entrada principal?
	4. Há necessidade de maior robustez na entrada de dados?
	5. Há alguma restrição com relação ao tempo nas entradas?
	6. Todos os tipos de saídas foram identificadas?
	7. Qual é a saída principal?
	8. Há algum estrangulamento de tempo nas saídas?
	9. Há conjuntos que estão em ambos, na entrada e na saída?
	10. Quais são os tipos de execução?
	11. Qual é a entrada e a saída para cada tipo de execução?
	12. Para cada tipo de execução, um valor de saída é especificado para cada valor de entrada?
	13. Estão definidos todos os mecanismos dos estados iniciais ?
	14. Todas as restrições do ambiente foram descritas?
	15. São providos exemplos de amostras de entrada/saída apropriados ?
	16. Todos os requisitos de desempenho necessários estão descritos?
	17. Todos os requisitos de confiabilidade foram especificados?
Ambiguity	1. Todos os termos especiais estão claramente definidos?
	2. Cada sentença tem uma interpretação única no domínio do problema?
	3. O delineamento da entrada-para-saída está claramente definido para cada tipo de execução?
Consistency	1. Qualquer um dos requisitos designados estão em conflito com o material analisado?
	2. Existe alguma entrada que está delineada para mais de uma saída?
	3. Há um conjunto de unidades quantitativas a ser usada? Todas as quantidades numéricas são consistentes?

Em MARTIN (1992), os autores investigaram uma variação na inspeção tradicional. Nesta variação, em vez de uma inspeção que é executada em um documento de requisitos, foram investigados os benefícios ganhos em inspeções múltiplas, independente de ser o mesmo documento de requisitos a ser inspecionado. O DRS usado no estudo foi de uma Companhia Ferroviária. Os defeitos que estavam presentes no documento de requisitos consistiam em uma combinação que o autor do mesmo fez durante a criação do documento, como também alguns defeitos adicionados pelos investigadores para equilibrar os números entre as diferentes classes de defeitos. Foram criadas duas classes de alto nível com subclasses em cada uma. A razão para esta

classificação seria para verificar se um novo processo de inspeção seria afetado no desempenho dos inspetores em encontrar quaisquer dessas classes de defeitos.

O Quadro 2.6 relaciona as classes de defeitos nos dois níveis.

Quadro 2.6 - Classes de Defeitos para Documentos de Software

Classes de Alto Nível	Subclasses	Características Avaliadas
Informação Perdida	Funcionalidade Perdida ou Característica Perdida	A informação que descreve o comportamento interno do sistema foi omitida.
	Interface Perdida	A informação sobre como o sistema comunica-se com os objetos fora dele foi omitida.
	Desempenho Perdido	Especificações de desempenho foram omitidas ou não foram descritos os modos de teste.
	Ambiente Perdido	Informações externas sobre hardware/software/banco de dados/pessoal envolvido foram omitidas.
Informação Errada	Informação Ambígua	Um termo importante não foi definido ou apresenta interpretações múltiplas
	Informação Inconsistente	Duas partes do documento contradizem uma com a outra.

Em estudos desenvolvidos por PORTER (1994; 1995) discute-se a técnica de inspeção de Leitura Baseada em Cenários, em que cada inspetor executa uma série de etapas para descobrir uma classe particular de defeitos. Os domínios dos requisitos inspecionados foram um sistema de controle de cruzeiro e um sistema que monitora o nível de água em um reservatório. A classificação de defeitos adotada no estudo foi obtida a partir dos estudos de MARTIN (1992) e LANUBILE (1998). Neste experimento, os autores comparam as técnicas de leitura *Ad Hoc*, *Checklist* e *Cenários*. A classificação de defeitos tem dois níveis, sendo que o primeiro é muito abstrato e o segundo é mais útil para a descoberta de defeitos. Os autores combinaram o esquema de classificação de acordo com o ambiente de trabalho.

O Quadro 2.7 apresenta a classificação de defeitos.

Quadro 2.7 - Esquema de Classificação dos Defeitos em Software

Classe	Subclasse	Características
Omissão	1. Funcionalidade Perdida	A informação descreve que o desejado comportamento interno operacional do sistema foi omitido.
	2. Desempenho Perdido	A informação descrita para a especificação do desempenho desejada foi omitida ou foi descrita de modo inaceitável para os testes.
	3. Ambiente Perdido	Informação descreve que os requisitos de hardware, software, banco de dados e o pessoal envolvido foram omitidos do sistema.
	4. Interface Perdida	Informação descreve que a comunicação do sistema proposto com os objetos que estão fora do escopo do sistema foram omitidos.
Comissão	1. Informação Ambígua	Um termo importante, frase ou sentença para o entendimento do sistema foi indefinido ou definido de modo que possa causar confusão ou mal entendimento.
	2. Informação Inconsistente	Duas sentenças contradizem-se ou expressam ações de que não estão corretas.
	3. Funcionalidade Incorreta	Alguma sentença afirma um fato que não pode ser verdade diante das condições especificadas.
	4. Seção Errada	Informação está extraviada dentro do documento.

Em BASILI (1996) é descrito um estudo empírico de Leitura Baseada em Perspectivas (LBPe), com o objetivo de determinar se a técnica LBPe é mais efetiva que a técnica de leitura *Ad Hoc*. Neste estudo é usado um esquema de classificação de defeitos diferente, em alguns aspectos, do mencionado acima. O Quadro 2.8 apresenta a classificação de defeitos nivelada criada pelos autores.

Quadro 2.8 - Classificação de Defeitos para Documentos de Software

Classificação do Defeito	Características
Omissão	Um defeito é classificado como omissão se a informação necessária sobre o sistema não está presente na documentação de requisitos do software, as informações foram omitidas do artefato de software.
Informação Ambígua	Um defeito é classificado como ambíguo, se o desenvolvedor pode interpretar as informações de diversas maneiras, ou seja, um requisito pode ser interpretado de várias maneiras, informações que influenciam na funcionalidade do sistema.
Inconsistência	Defeitos de inconsistência são representados por qualquer informação na documentação de requisitos do software que contradiz outra informação dentro da documentação de requisitos, ou seja, dois ou mais requisitos estão em conflito.
Fato Incorreto	Um defeito é classificado como fato incorreto se qualquer informação no Documento de Requisitos de Software descreve um fato que não é verdadeiro, de acordo com as solicitações da aplicação em desenvolvimento.
Informação Estranha	Uma informação estranha representa qualquer informação presente na documentação de requisitos que não é necessária ou mesmo utilizada.
Outros	É representado por qualquer defeito que os participantes do experimento detectam, mas que não pertence às outras classes de defeitos relatadas.

A classificação de defeitos detectados nas demais etapas do ciclo de vida do software é discutida em BASILI (1987), BASILI (1996), CARVER (2002), PARNAS (1985), SHULL (1998) e TRAVASSOS (1999).

O próximo capítulo apresenta as técnicas de leitura LBCh, LBCe e LBPe, utilizadas no estudo empírico, e alguns estudos realizados por autores na área de inspeção e os resultados obtidos, de acordo com as técnicas usadas.

CAPÍTULO 3

TÉCNICAS DE INSPEÇÃO APLICADAS A REQUISITOS DE SOFTWARE

3.1 CONCEITOS

A inspeção é um método para detectar defeitos em documentos de requisitos de software. Foi iniciada primeiramente por FAGAN (1976) na IBM. Outras técnicas usadas para descoberta são: as revisões, *walkthroughs* e auditorias. O padrão *IEEE Computer Society* (ANSI/IEEE, 1997) descreve os tipos de revisões formais de software:

- A revisão técnica (*Technical review*) tem o objetivo de avaliar artefatos específicos para verificar se eles apresentam os respectivos padrões e especificações e se eventuais modificações no artefato foram efetuadas de maneira correta. As revisões técnicas são aplicadas a documentos, como Especificação de Requisitos e Descrição de Testes, com o objetivo principal de verificar a conformidade com padrões do processo, e seus atributos de qualidade; tais como completude, corretude, consistência, verificabilidade, modificabilidade, rastreabilidade.
- A inspeção (*inspection*), mais formal que a revisão técnica, tem como objetivo principal identificar e remover defeitos. É obrigatória a geração de um lista de defeitos com classificação padronizada, exigindo-se a ação dos autores para remoção destes defeitos. As inspeções são aplicadas em documentações, implementação e testes. Já as revisões técnicas têm maior enfoque na qualidade da documentação.

- A revisão de apresentação (*walkthrough*) é uma revisão na qual o autor apresenta o material em ordem lógica, sem limite de tempo, a um grupo que verifica o material, conforme ele vai apresentando. Este tipo de revisão não exige muita preparação prévia, e pode ser feito com um número maior de participantes pelo fato dos participantes terem um papel mais passivo. As revisões são consideradas como de eficácia média para a detecção de defeitos. São usadas em marcos do projeto em que são necessárias as apresentações ao cliente.
- A auditoria (*audits*) tem o objetivo de verificar a conformidade de produtos e projetos com padrões e processos. O padrão prevê as auditorias da qualidade, que verificam aspectos de conformidade com o processo e com o Plano de Qualidade do projeto.

O Quadro 3.1 apresenta um resumo das características dos principais tipos de revisão, adaptado do padrão *IEEE Computer Society (ANSI/IEEE, 1997)*:

Quadro 3.1 - Resumo das características dos tipos de revisão

Atributos/Categoria	Revisão Técnica	Inspeção	Revisão de Apresentação
Objetivo	Avaliar conformidade, e verificar as alterações	Detectar e identificar defeitos, acompanhar a resolução	Detectar defeitos, avaliar alternativas, apresentar resultados
Método de Decisão	Revisores recomendam, gerentes agem ou se justificam	Revisores apontam defeitos, gerentes providenciam remoção	Autores e gerentes decidem sobre alterações
Verificação da Decisão	Acompanhamento pelo líder ou auditorias da qualidade	Acompanhamento pelo líder, auditorias da qualidade	Análise pela Gerência Executiva, aceitação pelo cliente
Tamanho da Equipe	5 - 8	5 - 8	Limites físicos
Equipe	Autores e revisores	Autores e revisores	Autores
Apresentador	Autores, líder	Autores, leitor	Autores
Dados Coletados	Defeitos, esforço de preparação, revisão e correção	Defeitos, esforço de preparação, revisão e correção	Defeitos, preparação e correção

A partir da comparação do Quadro 3.1, a meta de uma revisão, normalmente, é avaliar o estado do projeto e são usadas as auditorias para certificar um

produto, processo, projeto, ou programa de qualidade. Assim, a descoberta de defeitos em revisões são entregues às auditorias como um subproduto. Já o foco do processo de *walkthrough* está na descoberta do defeito, mas a sua utilização é menos formal que o método de inspeção. Basicamente, inspeções consistem em três passos: a fase de revisão individual em que os inspetores lêem a documentação com o propósito de detectar defeitos; a fase da reunião, em que todos os inspetores que pertencem à equipe de inspeção se encontram para discutir os resultados da revisão individual; e a fase do retrabalho, em que os defeitos identificados durante a reunião serão removidos.

3.2 DOCUMENTAÇÃO

Para reduzir o custo de desenvolvimento de software, é importante descobrir e remover os defeitos no documento do software o mais cedo possível. A razão é simples: quanto mais cedo se detecta o defeito, certamente o custo da correção será menor. O interesse na eficiência da inspeção está em ajudar a descobrir os defeitos. Um passo importante nas inspeções é a fase de revisão individual. Assim sendo, é necessário utilizar técnicas de leitura eficientes que apóiam a fase de revisão individual. Nota-se que muitas pesquisas são realizadas sobre processo de inspeção, mas muito pouco sobre técnicas de leitura que apóiam a etapa de revisão individual, a principal atividade relacionada à eficiência de inspeções. Recente pesquisa mostrou que os defeitos coletados na reunião, não aumentam o número de defeitos descobertos, comparados ao número de defeitos descobertos na fase de revisão individual (CIOLKOWSKI, 1999). Assim, as inspeções eficientes confiam em técnicas de leitura eficientes que apóiam a fase de revisão individual.

Nas próximas seções serão apresentadas algumas técnicas existentes para inspecionar documento de requisitos. Primeiramente, será apresentado um esquema de classificação para técnicas de leitura, depois será descrita cada técnica de leitura para documentos de requisitos, caracterizando-as de acordo com o esquema de classificação. Mais adiante serão apresentados estudos

empíricos que foram administrados para examinar a eficiência das técnicas de leitura.

3.3 ESQUEMA DE CLASSIFICAÇÃO DE TÉCNICAS DE LEITURA

O interesse do presente estudo está na eficiência da inspeção em diferentes técnicas de leitura. A seguir é apresentado um esquema de classificação para caracterizar as técnicas de leitura com respeito à sua aplicação por equipes de inspeção. A classificação segue abaixo (PORTER, 1995).

a) Lingüística. Um modo para caracterizar técnicas de leitura é o grau de sua lingüística: Como é a orientação que o inspetor recebe e como são as instruções técnicas dadas? O grau da Lingüística pode ser intuitivo (sem nenhuma orientação, por exemplo, técnica *AD HOC*), não-sistemático (os inspetores recebem alguma orientação, por exemplo, onde procurar os defeitos, mas não em como olhar), ou sistemático (os inspetores recebem orientação em como e onde procurar defeitos), ou ainda altamente sistemático (técnicas completamente formais; por exemplo, provas formais de corretitude). Para simplificar, a lingüística está dividida em dois níveis: não-sistemático (a lingüística percorre do intuitivo para o não-sistemático) ou sistemático (sistemático ou altamente sistemático). Assim, uma técnica é considerada como sistemática se prover orientação explícita dos defeitos a serem procurados e como a atividade de descoberta é realizada.

b) Responsabilidade do inspetor. Um segundo assunto para caracterizar técnicas de leitura é a responsabilidade específica que um inspetor tem. Inspetores podem ter responsabilidades gerais ou específicas. Como responsabilidade geral, um inspetor deve conferir a documentação completa para todo tipo de defeito. Se um inspetor tem responsabilidade específica, ele só deve conferir aspectos específicos dos defeitos da documentação. Assim, um inspetor pode concentrar-se em um (pequeno) subconjunto de aspectos e trabalhar efetivamente neste subconjunto.

c) Responsabilidade da equipe. Um terceiro modo para caracterizar técnicas de leitura para inspeções é a responsabilidade da equipe; isto é, como a

técnica é usada nas inspeções. A responsabilidade da equipe pode ser idêntica ou distinta. É idêntica se todos os inspetores usam a mesma técnica de leitura. É distinta, se os membros aplicam técnicas de leitura diferentes. A responsabilidade da equipe não é completamente independente da responsabilidade do inspetor: Se a responsabilidade do inspetor for específica, é provável que outros inspetores não a tenham. Assim, a responsabilidade da equipe seria distinta. Porém, com a responsabilidade do inspetor geral, diferentes inspetores podem aplicar diferentes técnicas de leitura (todos têm responsabilidades de inspetor geral). Assim, a responsabilidade da equipe ainda seria distinta. Tipicamente, técnicas de leitura são usadas com inspetores com responsabilidade geral de tal modo que todo inspetor usa a mesma técnica; isto é, a responsabilidade da equipe é idêntica.

Na prática, as companhias não conduzem todas as inspeções. Se conduzirem as inspeções, tipicamente os inspetores aplicam técnicas de leitura que podem ser classificadas como não-sistemática, geral, e idêntica (PARNAS, 1985; PORTER, 1995). Tais técnicas de leitura não descrevem explicitamente como o inspetor deve procurar defeitos e que defeitos procurar (não-sistemático); todo inspetor deve conferir todos aspectos do documento inspecionado (geral); e, dentro de uma equipe de inspeção, todos os inspetores usam a mesma técnica (idêntica). PARNAS (1985) afirma que este caminho resulta em um alto grau de defeitos descobertos por todos os inspetores, então o esforço está perdido e a eficiência da inspeção sofre conseqüências. Além disso, todo inspetor deve conferir todos os aspectos do documento, porém nenhum é conferido em detalhes, já que eles só podem se concentrar em poucos aspectos. Assume-se que as técnicas de leitura que seriam mais eficientes nas inspeções são: as sistemáticas (têm orientação no quê e como procurar), as específicas (os inspetores podem se concentrar em certos aspectos), e as distintas (todos os inspetores da equipe conferem aspectos diferentes).

3.4 TÉCNICAS DE LEITURA

Técnicas de leitura apóiam a fase de revisão individual. Os inspetores aplicam as técnicas ao ler o documento, como por exemplo, no documento de

requisitos ou de código, com o propósito de encontrar defeitos. Essas técnicas, nas quais o termo “leitura” é usado para enfatizar as semelhanças com os processos mentais, para tentar entender suficientemente qualquer texto.

Uma técnica de leitura pode ser caracterizada como uma série de passos para o indivíduo analisar um produto textual de software alcançando a compreensão precisa de uma tarefa particular (BASILI, 1996). Esta definição envolve três aspectos importantes:

a) Uma série de passos. A técnica tem que dar orientação ao desenvolvedor. O nível de orientação pode variar, dependendo do nível da meta a ser atingida na tarefa, e pode variar de um procedimento específico passo a passo para um conjunto de perguntas focalizadas.

b) Análise individual. Técnicas de leitura estão relacionadas a um processo de compreensão que envolve um indivíduo. Apesar do método no qual esta técnica está embutida, pode-se requerer que os desenvolvedores trabalhem juntos, depois da revisão do documento feita por eles (por exemplo, discutir defeitos potenciais encontrados individualmente em um documento); a compreensão de algum aspecto do documento ainda é uma tarefa individual.

c) A compreensão precisa de uma determinada tarefa particular. As técnicas de leitura possuem uma meta particular que é apontar e produzir um certo nível de entendimento de alguns (ou todos) aspectos de um documento.

Existem grandes diferenças nas técnicas de leitura com respeito ao grau de lingüística; as técnicas de leitura não-sistemáticas não prescrevem nada, ao contrário das técnicas de leitura altamente sistemáticas, que orientam o inspetor ao que ele procura e no modo como fará a tarefa. A descoberta de defeitos com as técnicas de leitura não-sistemáticas dependem da experiência do inspetor (CIOLKOWSKI, 1997). Portanto, para garantir uma certa eficiência no processo de inspeção é importante o uso de técnicas de leitura sistemáticas.

3.4.1 TÉCNICA DE LEITURA *Ad Hoc*

Leitura *Ad Hoc* é ainda amplamente usada na prática. Os inspetores não são orientados e assim, a descoberta de defeitos depende da sorte e/ou da experiência/esforço do inspetor. A leitura *Ad Hoc* é classificada como não-sistemática (os inspetores não recebem nenhuma orientação), geral (os inspetores devem conferir todos os aspectos do documento) e idêntica (todos os inspetores usam a mesma técnica). As companhias, freqüentemente, aplicam leitura *Ad Hoc* conforme as diferentes especialidades (por exemplo, especialista de testes e especialista de interfaces) na leitura dos documentos inspecionados. Estes especialistas usam leitura *Ad Hoc* para revisar. Não obstante, como todo inspetor tem um ponto de vista diferente e concentra-se em aspectos diferentes do documento, a responsabilidade da equipe pode ser classificada como distinta, mesmo que a técnica seja não-sistemática.

3.4.2 TÉCNICA DE LEITURA BASEADA EM *CHECKLISTS*

Ao aplicar Leitura Baseada em *Checklists* (LBCh), os inspetores recebem uma lista de perguntas (freqüentemente de questões sim/não) sobre assuntos do documento a ser inspecionado (CIOLKOWSKI, 1999). Tipicamente, é utilizado um “não” como sugestões de respostas para um defeito. As perguntas são desenvolvidas sobre conhecimento dos defeitos típicos, relacionando ao tipo de documento a ser inspecionado. Os inspetores deverão responder estas perguntas juntamente com um guia para achar os defeitos. Alguns autores acreditam que LBCh não é uma técnica sistemática (PORTER, 1995), pois as perguntas têm sentido geral e não existe nenhuma estratégia para responder. No entanto, o *Checklist* oferece alguma orientação para os inspetores na forma do que conferir (mas não em como olhar os defeitos). Mesmo assim, esta técnica representa uma boa melhoria, se comparada com a técnica *Ad Hoc*. Além disso, o *Checklist* pode ser melhorado continuamente baseado nas experiências. Por exemplo, em um certo tipo de análise em que um defeito que ocorra com bastante freqüência, pode-se alterar a lista e o tipo de defeito pode ser inserido.

O *Checklist* pode ser utilizado para um domínio de aplicação particular e para tipos específicos de documentos, tal como documento de requisitos ou documento de código. É recomendado na literatura que toda a lista deve caber em uma página (CHERNAK, 1996). Entradas na lista consistem basicamente de dois componentes: “O que olhar” e “Como detectar os defeitos”. Assim, o primeiro componente representa uma lista de assuntos (por exemplo, fatores de qualidade) que será conferido, e o segundo é uma lista de sugestões sobre como identificar defeitos que pertencem a esses assuntos. O Quadro 3.2 mostra um exemplo de um *Checklist*.

Quadro 3.2 - Exemplo de *Checklist*

Nº	Fatores de Qualidade (O que olhar)	Avaliação (Como Detectar)
1	Entrada - Forma	O sistema é de fácil operação ?
2		Existem sobreposições de telas ?
3		Existe Help On Line ?
4	Entrada - Consistência	Existe validação de sintaxe nos campos?
5		Existe validação de arquivos nos campos existentes ?
6		Existe validação encadeada (If's Aninhados) nos campos?

O *Checklist* apresenta algumas dificuldades, segundo PARNAS (1985) e LAITENBERGER (2001), como, por exemplo:

a) Como já mencionado, as perguntas são muito gerais e as estratégias para respondê-las são raras. Isto é, deve-se conhecer qual é a informação que o inspetor deve ter para responder as perguntas. O problema pode ser evitado acrescentando perguntas mais específicas às listas. Porém, isso pode resultar em listas muito longas e com várias páginas.

b) Inspectores não precisam documentar a sua inspeção, isto é, não precisam anotar as respostas e nem justificá-las. Os resultados não são repetíveis e dependem da experiência e eficiência dos inspetores.

c) Os *Checklists* geralmente contêm muitas páginas e também muitas perguntas. Por isso, os inspetores tendem a ignorar algumas ou a maioria delas, pois responder todas seria muito tedioso e demorado.

d) Os *Checklists* exigem que todos os inspetores confirmem toda a informação no documento a ser inspecionado, causando uma redução da eficiência da descoberta de defeitos com detalhes desnecessários ou mal entendidos.

e) Todos os inspetores usam a mesma lista. Assim, dentro de uma equipe de inspeção, todos os inspetores conferem os mesmos aspectos do documento e usam a mesma técnica de leitura.

Então, pode-se caracterizar a técnica LBCh como não-sistemática (porque não existe nenhuma estratégia para responder as perguntas), geral (porque os inspetores precisam conferir todos os aspectos do documento) e idêntica (porque todos os membros da equipe de inspeção inspecionam os mesmos aspectos).

3.4.3 TÉCNICA DE LEITURA BASEADA EM CENÁRIOS

Tipicamente, os *Checklists* são usados na fase de revisão individual nas inspeções (FAGAN, 1976). Entretanto, PARNAS (1985) critica esta técnica afirmando que as inspeções seriam mais eficientes se todo inspetor usasse uma técnica de leitura diferente, concentrando-se em aspectos diferentes dos documentos a serem inspecionados. Eles alegam que se os inspetores são responsáveis somente por certos aspectos do documento inspecionado, na verdade eles são especialistas, podendo ignorar as partes da documentação que não conhecem.

PARNAS (1985) propôs revisões ativas de projetos, uma técnica de leitura em que todo inspetor lê para um propósito específico, usando uma técnica especializada (responsabilidade individual específica). Assim, o documento é inspecionado por vários especialistas (como o especialista de interface e especialista de domínio da aplicação), e cada especialista faz o *Checklist* da sua área específica. Nesta revisão os inspetores conferem somente certas informações no documento a ser inspecionado para detectar os defeitos, ou seja, as informações que cabem a eles na sua especialidade, ignorando as que eles não possuem conhecimento. A combinação de vários inspetores com diferentes responsabilidades resulta numa equipe eficiente, porque os

inspetores podem concentrar em certos aspectos da documentação, não tendo que conferir todo o documento. (responsabilidades distintas na equipe).

Técnicas de Leitura Baseadas em Cenários usam a idéia de revisões ativas de projeto para atribuir responsabilidades específicas aos inspetores de uma equipe de inspeção. Embora as revisões ativas de projeto não mostrem como a inspeção deve ser feita, técnicas baseadas em Cenários descrevem mais especificamente como procurar defeitos. O inspetor segue um conjunto de instruções e diretrizes chamado Cenário. O Cenário força o inspetor a fazer um papel de leitura ativo, exigindo por exemplo, que ele escreva os casos do teste. Um ganho em Cenários é que, seguindo um cenário, o inspetor adquire a real perspicácia do sistema que está descrito no documento de requisitos comparado a inspetores que aplicam LBCh. Os inspetores são forçados a exercer um papel mais ativo e pensam mais profundamente no sistema. Com um maior entendimento do sistema, inspetores conseguem achar defeitos mais sutis. Os Cenários usam questões e *checklists* altamente especializados que ajudam a descobrir defeitos usando o trabalho já feito.

Dentro de uma equipe de inspeção, todo inspetor usa um cenário diferente tendo conseqüentemente uma responsabilidade diferente. O objetivo da criação da Técnica de Leitura Baseada em Cenários é encontrar uma combinação de cenários que cubra todos os aspectos do documento a ser inspecionado e, ao mesmo tempo, seja o menos redundante possível. Duas famílias de técnicas de Leitura Baseada em Cenários para documentos de requisitos foram propostas na literatura: Leitura Baseada em Defeito (LBDe) (PORTER, 1995) e Leitura Baseada em Perspectivas (LBPe) (BASILI, 1996).

3.4.4 TÉCNICA DE LEITURA BASEADA EM PERSPECTIVAS

Atualmente, a maioria dos documentos de requisitos são declarados em linguagem natural. Então, o próximo passo do LBPe é aplicar a idéia baseada em Cenários no documento de requisitos natural. A razão básica da Leitura Baseada em Perspectivas (LBPe) é que diferentes consumidores ou

*stakeholders*¹ do documento inspecionado estão interessados nos diferentes fatores de qualidade do documento inspecionado ou vêem o mesmo fator de qualidade diferentemente (CIOLKOWSKI, 1999). Então, cada inspetor deveria denotar o seu interesse no documento (PARNAS, 1985), dando sentido para se ter o documento inspecionado do ponto de vista do *stakeholder* (LAITENBERGER, 2001). Correspondentemente, LBPe define perspectivas para o documento de requisitos dos *stakeholders*.

A primeira família de técnicas de Leitura Baseada em Perspectivas, desenvolvida na Universidade de Maryland, BASILI (1996), identificou três *stakeholders*: o usuário do sistema, o projetista e o responsável pelos testes. As perspectivas desta técnica foram construídas de um modo que os inspetores são forçados a enxergar o ponto de vista do usuário, do projetista e do testador do sistema, utilizando a perspectiva correspondente.

A técnica LBPe consiste em três partes principais: introdução, instruções e perguntas. A introdução é um resumo que explica como o inspetor deverá usar a perspectiva. As instruções informam o inspetor como ele deve extrair a informação do documento de requisitos. O inspetor deve construir um modelo físico do sistema. Por exemplo, a perspectiva do testador orienta o inspetor na criação de casos de testes para o documento de requisitos. O efeito destas instruções são três: a) as instruções ajudam os inspetores a decompor os requisitos do documento em partes menores, pois deste modo podem ser mais facilmente controlados; b) forçam os inspetores a trabalhar ativamente com os requisitos e com a perspectiva; c) focalizam a atenção do inspetor nas informações mais relevantes. Os inspetores ganham um entendimento mais profundo do sistema e, além disso, asseguram que eles estão bem preparados para as próximas atividades, como para a reunião para coletar os defeitos. As perguntas, finalmente, ganham uma compreensão mais profunda com as instruções. Basicamente, estas perguntas representam um questionário altamente especializado. O questionário focaliza a atenção do inspetor para aspectos específicos do seu trabalho, auxiliando a descobrir os defeitos.

¹ “*stakeholders*” são pessoas ou organizações capazes de influenciar as definições sobre a funcionalidade e as restrições impostas ao sistema a ser desenvolvido. Incluem usuários, gerentes, entre outros, envolvidos no processo organizacional.

Existem várias versões de LBPe. A versão atual de LBPe para documentos de requisitos (LANUBILE, 1998) define três perspectivas: a) a perspectiva do usuário que exige que o inspetor desenvolva casos de uso para o sistema; b) a perspectiva do projetista que exige que o inspetor construa um diagrama de fluxo de dados do sistema; e c) a perspectiva do testador que exige que o inspetor realize casos de teste.

Pode-se classificar LBPe como sistemática (porque uma perspectiva orienta o inspetor com respeito a como e onde procurar defeitos), específica (porque o inspetor concentra-se em certos aspectos) e distinta (porque os inspetores com a equipe de inspeção usam perspectivas diferentes).

3.5 AVALIAÇÃO EMPÍRICA DE TÉCNICAS DE LEITURA PARA INSPEÇÃO DE SOFTWARE

Nesta seção, serão apresentados estudos empíricos que foram realizados para investigar os benefícios de diferentes técnicas de leitura, comparando a eficiência entre elas, em certos contextos.

3.5.1 EXPERIMENTOS COM A TÉCNICA DE LEITURA BASEADA EM PERSPECTIVAS

A primeira família de técnicas de Leitura Baseada em Perspectivas foi desenvolvida na Universidade de Maryland por BASILI (1996). A experiência original consistiu de uma execução do teste na Universidade de Maryland, e de duas execuções por profissionais da NASA em 1994 e 1995. Comparou-se a técnica PBR com uma técnica não-sistemática utilizada na NASA. Houve replicações na Universidade de Kaiserslautern (CIOLKOWSKI, 1997), onde LBPe foi comparado com uma técnica *Ad Hoc* para estudar os efeitos nas reuniões de equipes; no Instituto Experimental de Engenharia de Software de Fraunhofer (IESE) em Kaiserslautern (Alemanha), os participantes eram desenvolvedores profissionais, mudando as perspectivas para inspeções de código (LAITENBERGER, 1997); e uma replicação na Universidade de Trondheim, (Noruega) alterou as técnicas de LBPe para estudar os aspectos do processo (SORUMGARD, 1996).

As experiências nas Universidades de Maryland e Kaiserslautern mostraram que, para equipes simuladas, LBPe elevou a eficiência da inspeção (i.e., eficiência da equipe) significativamente. Os experimentos da experiência original em Maryland não envolviam reuniões de equipes reais, e os experimentos da replicação em Kaiserslautern não formaram equipes para uma análise segura.

A replicação na Universidade de Trondheim não pôde confirmar os resultados esperados. Porém, as idéias conduziram a um redesenho das técnicas de LBPe, novamente em Maryland. Esta versão da técnica de LBPe deu base para o experimento de LBPe2 na Universidade de Maryland (LANUBILE, 1998). O Quadro 3.3 mostra um resumo das replicações da experiência inicial.

Quadro 3.3 - Replicações do estudo de LBPe

Local	Nº de Execuções	Pessoas Envolvidas	Tratamentos	Resultados
Universidade de Maryland	1	Desenvolvedores Profissionais da NASA	LBPe e técnicas de leitura usuais	Nenhuma diferença significativa
Universidade de Maryland, NASA	1	Desenvolvedores Profissionais da NASA	LBPe e técnicas de leitura usuais	Simulação das equipes de LBPe significativamente melhores
Universidade de Kaiserslautern	2	Estudantes de Mestrado	LBPe e <i>Ad Hoc</i>	Simulação das equipes de LBPe significativamente melhores, sem resultados reais das equipes
IESE	1	Desenvolvedores Profissionais	Versão LBPe para leitura de código	Alta efetividade quando comparada com as três perspectivas, baixa parcial entre as perspectivas
Universidade de Trondheim	1	Estudantes de Mestrado	LBPe e perspectivas modificadas de LBPe	Nenhuma Confirmação
Universidade de Maryland (LBPe 2)	1	Estudantes graduados	Leitura LBPe2 e <i>Ad Hoc</i>	Nenhuma Confirmação

3.5.2 ESTUDOS EMPÍRICOS EM INSPEÇÃO DE REQUISITOS DE SOFTWARE

Estudos Empíricos em Inspeção de Requisitos de Software são motivados pela necessidade de se avaliar a qualidade de um artefato de software e saber se as tecnologias existentes auxiliam na construção de uma base sólida de informações, contribuindo para a evolução dos métodos existentes e para a área de computação de forma geral. Avaliar o custo/benefício das técnicas de inspeção tem sido uma preocupação constante dos pesquisadores. Isso se deve ao fato do custo elevado na manutenção de sistemas decorrente dos defeitos inseridos durante o seu desenvolvimento. Durante as últimas décadas pesquisas apontam a necessidade da replicação de experimentos com o objetivo de comparar técnicas de detecção de defeitos com relação à eficiência em observar inconsistências e falhas para isolar os defeitos dos projetos desenvolvidos, bem como analisar os aspectos complementares das técnicas de inspeção. Nesta seção serão relatados alguns resultados de estudos empíricos de autores que contribuíram para a qualidade dos produtos com o auxílio do processo de inspeção de um artefato de software.

Estudo 1 - Uma Experiência para Avaliar Diferentes Métodos de Descoberta de Defeito para Inspeções em Requisitos de Software PORTER (1994).

Porter e Votta executaram um experimento controlado num documento de Especificação de Requisitos de Software (SRS) que, normalmente, é validado nas inspeções nas quais vários revisores lêem tudo ou parte da especificação à procura de defeitos. Este estudo mostra que métodos diferentes de detecção de defeitos podem apresentar resultados diferentes durante a mesma inspeção. Neste projeto experimental foi feita a comparação dos métodos de descoberta de defeitos que são flexíveis e econômicos, e permitem ao investigador medir o efeito de várias ameaças em garantir a validade da experiência. Em particular, ficou determinado no experimento que nem a experiência adquirida pelos revisores, bem como as composições das equipes tiveram alguma influência significativa na eficiência da inspeção. Os resultados da experiência inicial foram retirados de dezesseis observações, resumidos a seguir:

- a) A taxa de detecção de defeitos, quando se utiliza Cenários, é superior à que foi obtida com as técnicas *Ad Hoc* ou *Checklist* - uma melhoria de aproximadamente 35%;
- b) A técnica de Cenários foi mais eficiente porque os revisores podem focalizar os defeitos em classes específicas;
- c) O método de *Checklist* foi o mais pobre dos três métodos de descoberta. Isso se deve ao fato do método Cenários ter sido construído a partir do método *Checklist* que foi um dos projetos pioneiros em inspeção.

Os resultados desse trabalho têm implicações importantes para engenheiros de software. Eles indicam que o desempenho global da inspeção pode ser melhorado quando revisores individuais usam procedimentos sistemáticos para se dirigir a pequenos assuntos específicos. Isso contrasta com a prática habitual na qual os revisores não têm nenhum procedimento sistemático e nem responsabilidades claramente definidas. Do ponto de vista dos pesquisadores, este trabalho demonstra a viabilidade de construir e executar experiências com fundamento econômico, validando as recomendações de pesquisa. Com economia nos projetos é possível replicar o experimento em outros ambientes, com populações diferentes de engenheiros de software permitindo a análise e comparação de novos resultados.

Estudo 2 - Replicação de Estudos Empíricos em Engenharia de Software DÓRIA (2001)

Este trabalho descreve a replicação do experimento de BASILI (1996), com o uso de documentos de requisitos de software para analisar a eficiência das técnicas *Ad Hoc*, LBCh e LBPe, na detecção de defeitos.

O projeto experimental foi baseado em indivíduos divididos em dois grupos (Grupo 1 e Grupo 2). Cada grupo foi dividido em três subgrupos contendo dois indivíduos cada. Cada subgrupo foi relacionado a uma das perspectivas (projeto, teste e uso) da técnica LBPe.

No primeiro dia os indivíduos utilizaram as técnicas *Ad Hoc* e LBCh para revisarem os documentos de domínio genérico *Automatic Teller Machine* (ATM)

e *Parking Garage Control System* (PG). No segundo dia, cada indivíduo foi treinado em uma das perspectivas (projeto, teste e uso) para iniciar o processo de inspeção dos documentos de domínio genérico ATM e PG.

Nesta replicação foram consideradas técnicas convencionais as técnicas *Ad Hoc* e LBCh. Os documentos de requisitos de software utilizados foram os mesmos do experimento de BASILI (1996), são eles: ATM e PG. Os defeitos foram classificados de acordo com o esquema de classificação do experimento de BASILI (1996).

Um total de 18 alunos do curso de graduação em Ciência da Computação foi selecionado para participar do experimento, classificados em relação ao nível de proficiência com leitura em Inglês, anos/meses de experiência usando documentos de requisitos de software e anos/meses de experiência escrevendo documentos de requisitos de software.

A fase de treinamento foi realizada em dois dias consecutivos e contou com um total de 2 sessões. Foi utilizado o documento de requisitos do software ABC Vídeo *System*.

A fase de execução do experimento foi realizada em dois dias consecutivos. No 1º dia ocorreu uma sessão de inspeção com duração de 4 horas aplicando as técnicas *Ad Hoc* e LBCh nos documentos de requisitos ATM e PG. No 2º dia ocorreu uma sessão de inspeção com duração de 4 horas aplicando a técnica LBPe nos documentos de requisitos PG e ATM.

Os principais resultados obtidos com a replicação do experimento de BASILI (1996) foram:

- As taxas médias de detecção de defeitos alcançadas por cada técnica pelos grupos de inspeção foram similares;
- Indivíduos que aplicaram a técnica LPBe tiveram um desempenho melhor que os indivíduos que aplicaram as técnicas *AD Hoc* e LBCh.

Após a realização do experimento foram realizadas entrevistas com os participantes; a seguir apresentam-se algumas considerações feitas pelos participantes:

- A técnica LBPe apresentou maior facilidade de ser aplicada pelo fato de ser mais específica e direcionada;
- Os tempos destinados ao treinamento e a execução foram satisfatórios;
- O documento ATM foi mais fácil de entender, apesar de apresentar requisitos mais complexos, por fazer parte de um domínio de aplicação conhecido pela maioria dos participantes;
- O documento PG tinha requisitos mais claros, mas não fazia parte do domínio de aplicação conhecido pela maioria dos participantes.

Estudo 3 – Abordagem Goal-Question-Metric (GQM) para Avaliação da Qualidade de Software ABIB (1998).

Esta tese de mestrado teve como principal objetivo estudar a abordagem *Goal-Question-Metric (GQM)*, na avaliação da qualidade de produtos de software. Durante o experimento foi desenvolvida a ferramenta GQM-PLAN para avaliar a qualidade do software, fundamentada na abordagem GQM, exemplificando um estudo sobre avaliação de técnicas de inspeção utilizando o DRS (Documento de Requisitos de Software) no processo de inspeção.

O estudo foi realizado em um ambiente acadêmico, ou seja, na Universidade Federal de São Carlos (UFSCar) e contou com a participação dos alunos de Pós-Graduação em Ciência da Computação. A turma de alunos foi considerada adequada e suficiente para o estudo, e contou com a participação de 24 alunos que possuíam conhecimentos básicos sobre documento de requisitos de software. Os participantes foram divididos em oito equipes, cada uma delas com três alunos, formadas através de um sorteio.

No estudo foi utilizada a padronização de documentos de requisitos de software proposto pelo IEEE (ANSI/IEEE, 1998). Foram utilizados documentos de requisitos de software de sistemas já empregados em estudos anteriores.

Foram escolhidos três sistemas: um Sistema de Vídeo Locadora, um Sistema de Controle de Estacionamento e um Sistema de Caixa Eletrônico. As três técnicas escolhidas para validação e verificação dos documentos de requisitos foram: *Ad Hoc*, LBCh e LBCe.

O foco do estudo foi analisar a eficiência das técnicas de detecção de defeitos *Ad Hoc*, LBCh e LBCe, na inspeção de documentos de requisitos de software, de acordo com o número total e o tipo dos defeitos encontrados nos documentos inspecionados.

O experimento foi dividido em duas fases: o treinamento e a aplicação da avaliação.

- a) Treinamento. Essa fase teve a duração de 2 horas. Cada equipe inspecionou o documento de requisitos de software de um Sistema de Locadora de Vídeo.
- b) Fase Experimental. Esta fase envolveu duas etapas de inspeção. Os sistemas utilizados foram as especificações de um Sistema de Caixa Eletrônico e um Sistema de Controle de Estacionamento. Durante a primeira etapa quatro das oito equipes inspecionaram o documento do Sistema de Caixa Eletrônico e as quatro equipes restantes inspecionaram o documento do Sistema de Controle de Estacionamento. Na segunda etapa as equipes que inspecionaram o documento do Sistema de Caixa Eletrônico na etapa 1, inspecionaram o documento do Sistema de Controle de Estacionamento na etapa 2 e vice-versa. As equipes 1 e 5 usaram a técnica *Ad Hoc*, as equipes 2, 3 e 6 a técnica LBCh e as equipes 4, 7 e 8 a técnica LBCe.

Os resultados do estudo estão sintetizados a seguir:

- A porcentagem de defeitos detectados pela técnica *Ad Hoc* (40%) é menor que a porcentagem relativa a técnica LBCh (50%), que é menor que a porcentagem relativa a técnica LBCe (60%);
- A porcentagem de defeitos de omissão (40%) detectados é menor que a porcentagem de defeitos de comissão (60%).

Algumas considerações do estudo apontam que:

- As inspeções em equipe são mais eficientes, detectando uma quantidade maior de defeitos nos documentos que foram inspecionados;
- Em relação às técnicas utilizadas, a técnica *Ad Hoc* é menos eficiente ou igual ao LBCh, se comparada com a técnica LBCE que obteve um resultado mais significativo em relação as outras técnicas.

Estudo 4 – Avaliando a Eficiência de Diferentes Técnicas de Inspeção em Documentos de Requisitos Informais CIOLKOWSKI (1999).

A tese descreve uma experiência controlada para comparar a eficiência de duas técnicas de leitura para inspeções em documentos de requisitos informais. As técnicas examinadas são: Leitura Baseada em *Checklist* (LBCh) e Leitura Baseada em Perspectivas (LBPe). Para comparar as técnicas nesta experiência, foram focalizados dois aspectos de eficiência: (1) eficiência no processo de inspeção, e (2) eficiência da técnica de leitura. O primeiro aspecto é como as duas técnicas de leitura afetam a eficiência de inspeções, e o segundo, porque a revisão individual é uma atividade fundamental para inspeções eficientes.

Foram realizadas três execuções da experiência: uma na Universidade de Maryland, uma na Universidade de Kaiserslautern, e uma replicação externa na Universidade de Maryland em Baltimore. No experimento, os alunos liam o documento de requisitos e com a ajuda de uma técnica de leitura detectavam os defeitos.

Os resultados do experimento apontam que:

- Foi confirmada uma diferença entre a eficiência de equipes que aplicaram Leitura Baseada em *Checklist* (LBCh) e equipes que aplicaram Leitura Baseada em Perspectivas (LBPe) que se mostrou mais eficiente;
- Não foi confirmada nenhuma diferença na eficiência na leitura individual. Os resultados podem ser explicados pelo fato de que o documento de

requisitos era muito difícil para os estudantes detectarem os defeitos, independente da técnica de leitura aplicada.

Estudo 5 – Um Estudo de Caso com Técnicas de Leitura em uma Companhia de Software BERLING(2004).

O estudo de caso foi realizado na empresa Ericsson que desenvolve sistemas para radares. Os sistemas são desenvolvidos em três partes: projeto, implementação e testes. A técnica de leitura padrão usada na companhia é a LBCh, porém a organização estava interessada em investigar uma outra técnica mais eficiente para inspecionar o documento de requisitos dos sistemas. O objetivo do estudo empírico foi comparar se a técnica LBPe é mais eficiente que a técnica LBCh em relação a quantidade de defeitos detectados no documento de requisitos de software inspecionado.

As questões a serem respondidas no estudo foram: (1) A técnica LBPe oferece mais recursos no processo de inspeção comparada com a técnica LBCh, (2) Os benefícios de usar LBPe comparado com a técnica LBCh e (3) Os tipos de defeitos encontrados com a técnica LBPe comparado com a Técnica LBCh. Os defeitos foram classificados de acordo com o esquema de classificação do experimento de BASILI (1996) e um documento de requisitos de software da empresa foi utilizado durante a inspeção.

Os resultados do estudo empírico realizado foram:

- Foi confirmada uma diferença entre a quantidade de defeitos localizados entre a Técnica de Leitura Baseada em *Checklist* (LBCh) e a Técnica de Leitura Baseada em Perspectivas (LBPe) que se mostrou mais eficiente;
- Dentre os benefícios o tempo despendido para realizar a inspeção no documento de requisitos foi menor com o uso da técnica LBPe do que com a técnica LBCh;
- A técnica LBPe oferece um direcionamento melhor aos inspetores para inspecionar o produto, dando uma melhor cobertura na busca dos defeitos se comparado com a técnica LBCh;

- Apesar dos inspetores não terem conhecimento da técnica LBPe foram localizados um maior número por tipos de defeitos em relação a técnica LBCh.

O próximo capítulo apresenta as hipóteses formuladas e testadas no estudo, a classificação de defeitos empregada, as técnicas de leitura utilizadas os documentos e formulários criados, a realização do estudo empírico e as metodologias estatísticas empregadas para análise dos dados analisados.

CAPÍTULO 4

UM ESTUDO EMPÍRICO PARA AVALIAÇÃO DE TÉCNICAS DE INSPEÇÃO APLICADAS A DOCUMENTOS DE REQUISITOS DE SOFTWARE

4.1 OBJETIVO

O presente estudo empírico teve como objetivo identificar e comparar a eficiência de três técnicas de inspeção – Leitura Baseada em *Checklist* (LBCh), Leitura Baseada em Cenários (LBCe) e Leitura Baseada em Perspectivas (LBPe), no que diz respeito à detecção de defeitos em um Documento de Requisitos de Software (DRS).

A consecução do objetivo proposto envolveu a realização da inspeção de um DRS referente a um sistema real de “Programa de Controle Médico e Saúde Ocupacional” (PCMSO, 2004), desenvolvido por uma micro-empresa de desenvolvimento de software (AGSI Sistemas) da cidade de Americana e utilizado nessa região por clínicas ocupacionais. O DRS foi apresentado de acordo com o padrão internacional Std-830 de engenharia de requisitos, proposto pelo *Institute of Electric and Electronic Engineering (IEEE)*. O DRS foi inspecionado por profissionais que atuam na área de análise de sistemas e/ou engenharia de software distribuídos em 3 equipes de inspeção.

4.2 PERGUNTA DA PESQUISA

O estudo empírico buscou avaliar as três técnicas de inspeção, com a finalidade de responder às seguintes perguntas:

“Qual a técnica de inspeção - LBCh, LBCe ou LBPe, é a mais eficiente em termos de detecção de defeitos no DRS inspecionado pelos inspetores? Existe associação entre o nível de eficiência da técnica e o nível de experiência dos inspetores como analistas de sistemas e/ou engenheiros de software?”

Os analistas de sistemas ou engenheiros de software, que atuaram como inspetores, são considerados profissionais com formação acadêmica na área e que possuíam alguma experiência em atividades de análise e projeto de sistemas, em empresas ou organizações.

Como nível de experiência dos inspetores, considerou-se o número de meses que cada inspetor possuía como analistas de sistemas e/ou engenheiros de software.

4.3 HIPÓTESES DO ESTUDO

As hipóteses testadas através do estudo empírico são apresentadas a seguir.

a) Hipótese 1

As três técnicas de inspeção - LBCh, LBCe e LBPe, possuem o mesmo nível de eficiência no tocante à detecção de defeitos, obtida pelos inspetores, em um documento de requisitos de software.

O teste desta hipótese foi realizado medindo-se o número total de defeitos que cada indivíduo detectou ao inspecionar o DRS, e comparando-se os resultados obtidos pelas três equipes de inspeção. O número de defeitos detectado pelos indivíduos através de cada técnica foi representado pelas variáveis : NRO-DEFEITOS-LBCh (número de defeitos detectados através da técnica de Leitura Baseada em *Checklist*), NRO-DEFEITOS-LBCe (número de defeitos

detectados através da técnica de Leitura Baseada em Cenários) e NRO-DEFEITOS-LBPe (número de defeitos detectados através da técnica de Leitura Baseada em Perspectivas).

b) Hipótese 2

Não existe associação entre o nível de eficiência da técnica de inspeção (LBCh, LBCe e LBPe) e o nível de experiência dos inspetores como analistas de sistemas e/ou engenheiros de software.

O teste desta hipótese foi realizado identificando-se o número de defeitos que cada participante detectou ao inspecionar o DRS (representado pela variável “NRO-DEFEITOS-INDIVIDUAL”), e medindo-se o grau de associação entre este número de defeitos e o tempo médio de experiência, em meses, do participante (representado pela variável MÉDIA-EXPERIÊNCIA-INSPECTOR).

Para o teste das hipóteses, foram formuladas as respectivas hipóteses H_0 e H_a .

Hipótese 1

Parte do pressuposto de que não há diferença na eficiência entre as técnicas utilizadas na inspeção individual (LBCh, LBCe e LBPe), no que diz respeito à média dos defeitos detectados, ou seja:

H_0 : As três técnicas são igualmente eficientes.

Parte do pressuposto de que há diferença na eficiência entre as técnicas utilizadas na inspeção individual (LBCh, LBCe e LBPe), no que diz respeito à média dos defeitos observados detectados, ou seja:

H_a : Pelo menos uma das técnicas difere das demais.

Hipótese 2

Parte do pressuposto de que a experiência dos participantes não apresenta influência na eficiência entre as técnicas utilizadas na inspeção individual

(LBCh, LBCe e LBPe), no que diz respeito à média dos defeitos detectados, ou seja:

H_0 : Não há correlação entre experiência dos inspetores e número de defeitos.

Parte do pressuposto de que a experiência dos participantes apresenta influência na eficiência entre as técnicas utilizadas na inspeção individual (LBCh, LBCe e LBPe), no que diz respeito à média dos defeitos detectados, ou seja:

H_a : Há correlação entre as variáveis.

4.4 DEFINIÇÃO DOS PARTICIPANTES

O estudo empírico foi realizado com a participação de 15 profissionais de análise de sistemas e engenharia de software, que realizaram a inspeção do DRS referente ao sistema PCMSO. Os participantes foram organizados aleatoriamente em 3 equipes de inspeção, cada equipe com cinco participantes, sendo que um assumiu o papel de moderador da equipe e outro assumiu o papel de relator dos defeitos encontrados pela equipe.

Os participantes foram convidados a participar da pesquisa, o que configurou um tipo de “amostra por conveniência”. Antes de serem convidados, foi constatado que todos possuíam alguma experiência como analistas de sistemas e/ou engenheiros de software. Todos atuam profissionalmente na região de Piracicaba.

4.5 DOCUMENTO INSPECIONADO

O artefato de software inspecionado foi o DRS referente ao sistema de “Programa de Controle Médico e Saúde Ocupacional” (PCMSO, 2004), desenvolvido por uma micro-empresa de desenvolvimento de software da cidade de Americana e utilizado na região de Americana por clínicas

ocupacionais. O DRS foi apresentado de acordo com o padrão internacional Std-830 de engenharia de requisitos, proposto pelo IEEE (ANSI/IEEE, 1998).

O objetivo principal do sistema PCMSO é de gerenciar as avaliações e controlar se as exposições aos riscos estão gerando algum tipo de lesão ou doença no funcionário. O sistema tem como função cadastrar o funcionário da empresa e realizar o controle de exames e riscos de cada um periodicamente nas consultas previstas. A partir das avaliações realizadas periodicamente a clínica terá o controle e cadastro das doenças pré-existentes e também um histórico dos riscos e dos exames a serem realizados.

4.6 CLASSIFICAÇÃO DE DEFEITOS EMPREGADA

A classificação de defeitos foi adaptada a partir das classificações propostas pelos autores LANUBILE (1998) e MARTIN (1992). O Quadro 4.1 mostra a tabela de classificação de defeitos utilizada no estudo empírico.

Quadro 4.1 - Tabela de Classificação dos defeitos

Classe	Subclasse	Características
1 - Omissão	1. Funcionalidade Omitida (FO)	Alguma informação, relativa a descrição do comportamento esperado do sistema, não aparece no documento.
	2. Performance Omitida (PO)	Alguma informação, relativa à descrição da performance desejada, não aparece no documento, ou aparece de forma inaceitável.
	3. Ambiente Omitido (AO)	Alguma informação, relativa à descrição do hardware, do software, do banco de dados e do pessoal envolvido, não aparece no documento.
	4. Interface Omitida (IO)	Alguma informação, relativa à forma como o sistema interagirá ou se comunicará com componentes que estão fora do escopo do sistema, não aparece no documento.
2 - Comissão	1. Informação Ambígua (IA)	Um termo importante, uma frase ou uma sentença, essenciais para o entendimento do sistema não foi definido no documento, ou foi definido de forma que possa causar confusão.
	2. Informação Inconsistente (II)	Duas sentenças contradizem-se mutuamente ou expressam ações de que não estão corretas ou não podem ser executadas.
	3. Funcionalidade Incorreta (FI)	Alguma sentença expressa um fato que não pode ser verdade de acordo com as condições especificadas.
	4. Seção Errada (SE)	Alguma informação está em um local errado dentro do documento.
3 – Outros (O)		Defeitos que não se enquadram nos tipos acima.

No DRS referente ao sistema de PCMSO foram inseridos vinte defeitos de acordo com a classificação de defeitos do Quadro 4.1 utilizado no estudo.

4.7 REALIZAÇÃO DO ESTUDO EMPÍRICO

Para a realização da inspeção, todos os participantes foram convidados a comparecerem a UNIMEP, no dia 18/11/2004.

A pesquisa foi conduzida através de duas sessões. Na primeira sessão, o responsável pela pesquisa realizou uma apresentação geral do estudo empírico aos participantes, que incluiu a caracterização e discussão dos seguintes tópicos:

- Inspeção de software: conceitos, processo de inspeção, composição de equipes de inspeção;
- Documento de Requisitos de Software (DRS): estrutura do documento e exemplo;
- Técnicas de inspeção aplicadas a DRSs: Leitura Baseada em *Checklist* (LBCh), Leitura Baseada em Cenário (LBCe) e Leitura Baseada em Perspectivas (LBPe).

Na segunda sessão, também realizada no dia 18/11/2004, foi realizada a pesquisa propriamente dita, que incluiu as seguintes atividades:

- Composição das equipes de inspeção, obtida através de sorteio;
- Distribuição do Material para Inspeção. Todos receberam o DRS que seria inspecionado e, além deste documento, cada participante recebeu o Formulário de Instruções do Estudo, o Formulário de Classificação de Defeitos e o Formulário de Anotação de Defeitos referente à técnica de inspeção que seria empregada pela equipe. Estes formulários são apresentados nos apêndices A1 e A2.

Nesta segunda seção, foi entregue também um formulário sobre os aspectos gerais do estudo, que consta no apêndice A3. Os dados coletados nesse formulário respondido pelos participantes continha alguns dados pessoais e algumas questões a respeito do estudo e das ocorrências deste.

No quesito de dados pessoais, foram avaliadas as seguintes informações:

- Nível de Formação. Qualificação (por exemplo, estudante da graduação, estudante de mestrado);
- Experiência em Inspeções de Software. Em sala de aula, na empresa ou nenhuma;
- Tempo de Experiência. Em meses (a partir de 12 meses ou mais);

- Conhecimento nas técnicas de inspeção. Em sala de aula, na empresa ou nenhum.

Nas questões, as respostas foram avaliadas numa escala de 1 (um) a 5 (cinco), onde 1 (um) representa insuficiente, 2 (dois) fraco, 3 (três) regular, 4 (quatro) bom e 5 (excelente).

- Conhecimento dos participantes. Em escrever e ler documentos de requisitos; em detectar defeitos no software e em documentos de requisitos; nas técnicas de leitura e no processo de inspeção;
- Classificação e tipos de defeitos. Fácil entendimento e completo;
- Formulários do estudo. Fácil entendimento e completo;
- Artefatos do estudo: Suficientes para encontrar os defeitos;
- Técnica de inspeção. Fácil entendimento;
- Conhecimento nas técnicas. Adquiriu conhecimento sobre inspeção;
- Ambiente de Trabalho. Utilizar inspeção para o desenvolvimento de software.

O Quadro 4.2 resume as duas sessões realizadas no estudo empírico.

Quadro 4.2 - Plano do Estudo Empírico

Estudo Empírico	
1ª Sessão – Apresentação do Estudo	2ª Sessão – Execução do Estudo
Definição de Inspeção/ Benefícios	Distribuição do Material
Introdução ao Processo de Inspeção/ Etapas	Leitura das Instruções e dúvidas
Desenvolvimento do Processo de Inspeção	Equipe 1
Apresentação DRS – Modelo Padrão IEEE	Leitura DRS – PCMSO/ Técnica LBCh – Anotar Defeitos
Apresentação das Técnicas – LBCh, LBCE e LBPE	Equipe 2
Classificação de Defeitos do Estudo	Leitura DRS – PCMSO/ Técnica LBCE – Anotar Defeitos
Exemplos de defeitos	Equipe 3
Apresentação Material do Experimento – (DRS - PCMSO e Formulários)	Leitura DRS – PCMSO/ Técnica LBPE – Anotar Defeitos
Processo de Inspeção no Experimento	Finalização do Experimento
Divisão das Equipes	Questionário – Aspectos Gerais do Estudo

A Figura 4.1 ilustra o processo de realização da pesquisa.

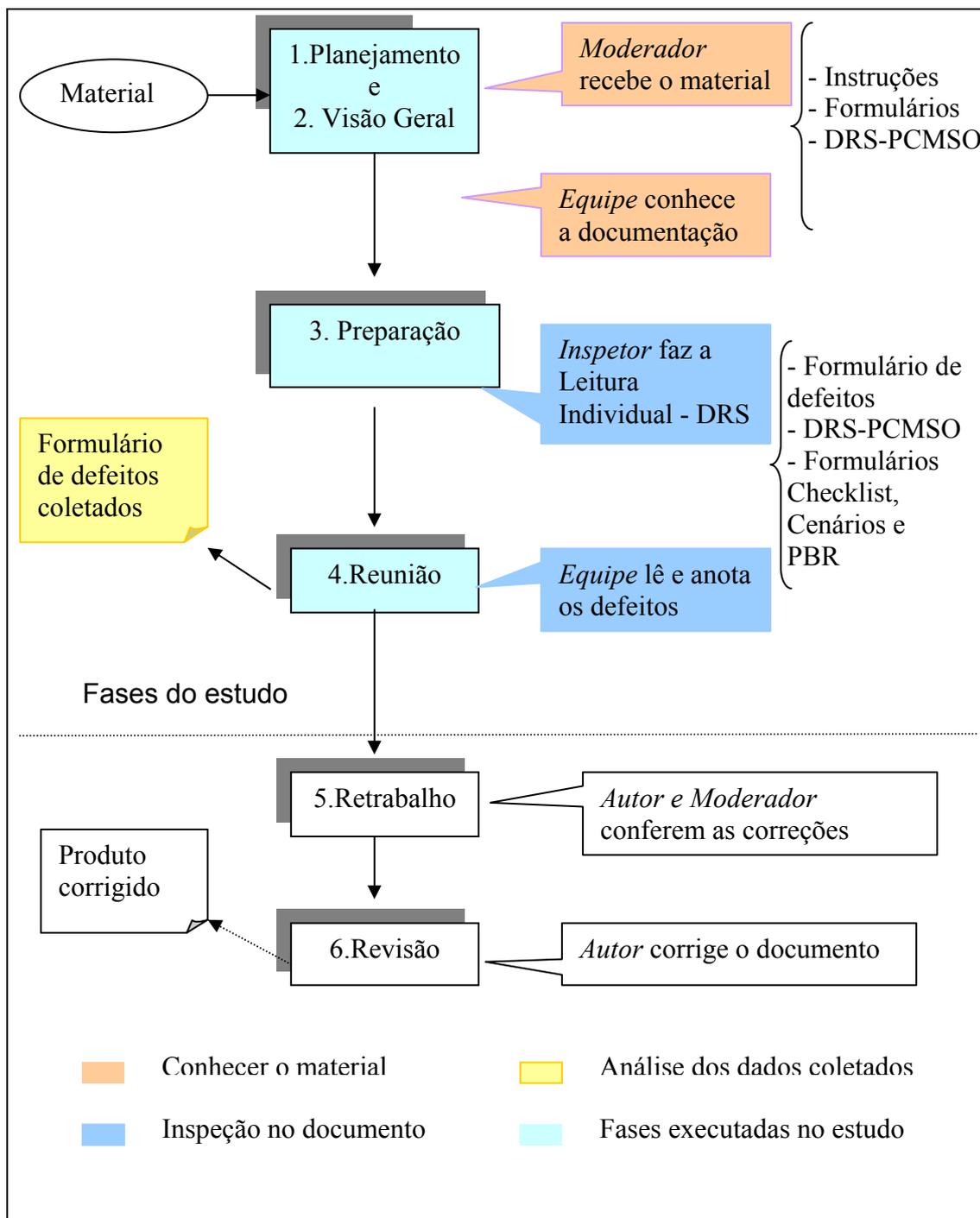


Figura 4.1 - Fases do processo de inspeção no estudo

4.8 FORMULÁRIOS E DOCUMENTOS

Na segunda sessão, os participantes utilizaram no estudo o formulário de respostas usado para auxiliar o inspetor a anotar os defeitos encontrados no documento inspecionado, nesse caso o sistema de PCMSO. No formulário, o

inspetor deveria identificar o nome do documento, o horário da inspeção, seu nome, sua função na equipe, o número do defeito, o tipo de defeito de acordo com a classificação definida no Quadro 4.1, o número da página, número da seção inspecionada, o número do requisito e uma breve descrição do defeito encontrado, o formulário de instruções a ser usado nas fases de Planejamento e Visão Geral do processo de inspeção no início do experimento, o Questionário sobre os aspectos gerais do estudo realizado e o relatório de participantes de acordo com a sua função. Os formulários utilizados estão nos apêndices A1, A2, A3 e A4.

4.8.1 FORMULÁRIO DAS TÉCNICAS DE LEITURA

a) Leitura Baseada em *Checklist* (LBCh)

Foi criada uma lista de perguntas, a partir do documento de requisitos inspecionado, partindo-se de três princípios: (1) todas as perguntas ajustar-se-iam em uma página; (2) as perguntas seriam aplicadas somente no documento de requisitos. Então, para isso o número de perguntas foi restringido para 20. Isto é recomendado na literatura CHERNAK (1996); e (3) o *checklist* seria estruturado de acordo com dois aspectos: com o esquema de classificação de defeitos usado no estudo, dividido em “Classe” e “Tipo”. Para isso o primeiro componente representa a classe a que o defeito pertence, dividido em: Comissão e Omissão e o segundo representa o tipo do possível defeito a ser detectado; e um conjunto de perguntas do *checklist* (sugestões) em como identificar defeitos que pertencem ao esquema de classificação de defeitos definido proposto por ABIB (1998). Foi criado, para cada tipo de defeito, um subconjunto de perguntas cobrindo os aspectos mais importantes do documento de requisitos do sistema PCMSO. O formulário da técnica LBCh encontra-se no apêndice B1.

b) Leitura Baseada em Cenários (LBCe)

Os Cenários usados foram definidos a partir de descrições parciais do sistema e do comportamento do macrosistema. Cada cenário é composto de: a) um nome (o que será avaliado no sistema), b) descrição (quais aspectos deverão ser analisados no documento), c) as questões, ou seja, os (*checklists* que

serão usados para auxiliar o inspetor a detectar os possíveis defeitos no documento) e d) a classe e tipo do possível defeito a ser detectado proposto por ABIB (1998). O apêndice B2 apresenta o formulário da técnica LBCe.

c) Leitura Baseada em Perspectivas (LBPe)

Nesse estudo, foi utilizada somente a Leitura Baseada em Perspectivas sob o ponto de vista do projetista. O formulário utilizado nesse experimento foi criado a partir da classificação de defeitos descrita no Quadro 4.1, com o objetivo de guiar o inspetor a ter uma visão do projeto do sistema, adotando alguma técnica de especificação de software de seu conhecimento. A partir dessa visão construída pelo inspetor, ele deveria responder a um conjunto de questões elaboradas, que apontam defeitos típicos no documento de requisitos, proposto por CIOLKOWSKI (1999). O apêndice B3 apresenta o formulário da técnica LBPe.

4.9 METODOLOGIAS ESTATÍSTICAS PARA ANÁLISE DOS DADOS

Seguindo os procedimentos formais de um estudo científico, ou seja, posterior à definição do problema, planejamento e coleta dos dados descrito abaixo, adotou-se metodologias para a apresentação e análise dos dados (BUSSAB, 1987).

a) Definição do problema. A fase consiste em uma definição ou formulação correta do problema a ser estudado, ou seja, o estudo empírico originou-se de uma pergunta da pesquisa e, posteriormente, da formulação das hipóteses a serem testadas.

b) Planejamento. Consiste na escolha das perguntas, bem como a sua formulação, e na escolha do tipo de levantamento a ser usado. Esse estudo foi realizado por uma amostra de conveniência buscando um perfil de participantes graduados na área de informática e com experiência profissional na área de análise de sistemas/ engenharia de software. Os materiais utilizados foram desenvolvidos para os participantes utilizarem na reunião de

inspeção, compreendidos entre alguns formulários e um questionário de aspectos gerais do estudo.

c) Coleta dos Dados. A coleta de dados refere-se à obtenção, reunião e registro dos dados com um objetivo determinado. No caso do estudo empírico, os participantes reuniram-se em três grupos, com cinco inspetores cada um, com o objetivo de encontrar o maior número possível de defeitos no DRS do sistema PCMSO, através das técnicas LBCh, LBCe e LBPe.

Neste trabalho, os dados foram analisados através de Técnicas de Análise Exploratória de Dados, ou seja, Estatística Descritiva e Testes de Hipóteses, que se trata de uma regra de decisão para aceitar ou rejeitar uma hipótese estatística. O estudo realizado foi para analisar se existem diferenças significativas entre as técnicas LBCh, LBCe e LBPe, no tocante à detecção de defeitos, pelas equipes, com o uso das metodologias estatísticas.

4.9.1 ANÁLISE DA HIPÓTESE 1

Para testar a hipótese 1 foi utilizado o teste de Kruskal Wallis, que é adequado para o tamanho da amostra. O teste de Kruskal Wallis para diferenças de medianas, tem como objetivo verificar se existem diferenças significativas entre as técnicas LBCh, LBCe e LBPe.

Para utilizar o teste de Kruskal Wallis é importante que a amostra em questão seja escolhida de forma aleatória (LEVINE, 2000).

No presente estudo, foi definido que, caso o teste de Kruskal Wallis apresentasse uma diferença significativa na análise das técnicas, a análise posterior dar-se-ia por meio de comparações múltiplas usando o Teste de Tukey-Kramer (LEVINE, 2000). Este teste é empregado como um procedimento de comparação posterior que consiste em:

- Examinar simultaneamente comparações entre todos os pares de grupos;
- Determinar quais das médias entre os pares dos grupos são significativamente diferentes das outras.

O estudo foi combinado através das três amostras das técnicas LBCh, LBCe e LBPe, com cinco inspetores em cada uma. As variáveis utilizadas para testar a hipótese foram NRO-DEFEITOS-LBCH, NRO-DEFEITOS-LBCE e NRO-DEFEITOS-LBPE que continham respectivamente o número de defeitos das técnicas de inspeção analisadas.

O resultado foi analisado a partir do nível de significância de 5%, comparado com o valor p obtido a partir das medianas entre o NRO-DEFEITOS-LBCH, NRO-DEFEITOS-LBCE e NRO-DEFEITOS-LBPE.

4.9.2 ANÁLISE DA HIPÓTESE 2

Para testar a hipótese 2, foi utilizado o teste de correlação linear de Pearson (BUSSAB, 1987), para avaliar a associação entre as variáveis do estudo. O objetivo deste teste é:

- Oferecer uma medida precisa da força e do sentido da correlação (que pode existir entre as variáveis) de acordo com a amostra;
- Pode-se verificar ainda se a associação obtida entre as variáveis existe de fato, e não resulta meramente de erro amostral.

O estudo foi combinado com as três amostras das técnicas LBCh, LBCe e LBPe, com cinco inspetores em cada uma. As variáveis utilizadas para testar essa hipótese foram NRO-DEFEITOS-INDIVIDUAL e MÉDIA-EXPERIÊNCIA-INSPECTOR, que continham respectivamente o número de defeitos das técnicas de inspeção analisadas e o tempo médio de experiência da amostra.

O resultado foi analisado a partir do nível de significância de 5%, comparado com o valor p obtido a partir da associação entre as variáveis NRO-DEFEITOS-INDIVIDUAL e MÉDIA-EXPERIÊNCIA-INSPECTOR.

O próximo capítulo apresenta a análise da hipótese 1 com o uso do teste de Kruskal Wallis e Tukey-Kramer e a análise da hipótese 2 com o teste de correlação linear de Pearson, uma análise do formulário de aspectos gerais e algumas comparações com estudos anteriores com esse estudo realizado.

CAPÍTULO 5

RESULTADOS E DISCUSSÃO

5.1 APRESENTAÇÃO DOS DADOS

Nessa fase do estudo foi realizado o trabalho de condensação e de tabulação dos dados, através de sua contagem e agrupamento.

A apresentação dos dados foi realizada com a contagem dos defeitos encontrados pelos participantes e pelos grupos envolvidos. Os dados foram apurados a partir dos formulários de anotações dos defeitos preenchidos individualmente pelos membros das equipes, e também pelos formulários preenchidos pelas equipes de acordo com os defeitos inseridos no documento, conforme apêndice A2.

O formulário de aspectos gerais do estudo forneceu o tempo de experiência em meses dos participantes, obtido através do apêndice A3. Por fim, os dados foram apresentados nos quadros 5.1, 5.2, 5.3 e 5.4 de acordo com as técnicas LBCh, LBCe e LBPe.

O Quadro 5.1 apresenta o número de defeitos detectados pelos indivíduos através da técnica LBCh, representado pela variável NRO-DEFEITOS-LBCH, utilizada para testar a Hipótese 1.

Quadro 5.1 - Defeitos Coletados com LBCh

Lista de Defeitos do DRS com LBCh – Grupo 1								
Classificação de Defeitos		Qtde Defeitos Originais	Qtde Defeitos Observados pelos Indivíduos do grupo NRO-DEFEITOS-LBCH					Qtde Defeitos Observados/Grupo
Classe	Comissão	11	5	8	5	10	4	10
	Omissão	9	7	6	4	8	7	8
Tipo	FO	4	3	3	3	3	3	3
	PO	2	1	1	0	2	1	2
	AO	2	2	2	0	2	2	2
	IO	1	1	0	1	1	1	1
	IA	2	0	1	0	1	0	1
	II	6	4	6	4	6	3	6
	FI	1	0	1	0	1	1	1
	SE	2	1	0	1	2	0	2

O Quadro 5.2 apresenta o número de defeitos detectados pelos indivíduos através da técnica LBCE, representado pela variável NRO-DEFEITOS-LBCE, utilizada para testar a Hipótese 1.

Quadro 5.2 - Defeitos Coletados com LBCE

Lista de Defeitos do DRS com LBCE – Grupo 2								
Classificação de Defeitos		Qtde Defeitos Originais	Qtde Defeitos Observados pelos Indivíduos do grupo NRO-DEFEITOS-LBCE					Qtde Defeitos Observados/Grupo
Classe	Comissão	11	4	5	7	6	6	7
	Omissão	9	7	7	7	7	6	7
Tipo	FO	4	3	3	3	3	2	3
	PO	2	2	2	2	2	2	2
	AO	2	2	2	2	2	2	2
	IO	1	0	0	0	0	0	0
	IA	2	0	0	1	1	1	1
	II	6	2	3	4	3	3	4
	FI	1	1	1	1	1	1	1
	SE	2	1	1	1	1	1	1

O Quadro 5.3 apresenta o número de defeitos detectados pelos indivíduos através da técnica LBPE, representado pela variável NRO-DEFEITOS-LBPE, utilizada para testar a Hipótese 1.

Quadro 5.3 - Defeitos Coletados com LBPe

Lista de Defeitos do DRS com LBPe – Grupo 3								
Classificação de Defeitos		Qtde Defeitos Originais	Qtde Defeitos Observados pelos Indivíduos do grupo NRO-DEFEITOS-LBPE					Qtde Defeitos Observados/Grupo
Classe	Comissão	11	9	6	3	7	8	10
	Omissão	9	8	7	7	8	8	8
Tipo	FO	4	4	3	3	4	4	4
	PO	2	2	2	2	2	2	2
	AO	2	2	2	2	2	2	2
	IO	1	0	0	0	0	0	0
	IA	2	1	2	0	1	1	2
	II	6	5	2	1	3	4	5
	FI	1	1	1	1	1	1	1
	SE	2	2	1	1	2	2	2

O Quadro 5.4 apresenta o número de defeitos que cada indivíduo de cada equipe de inspeção detectou ao inspecionar o DRS, representado pela variável NRO-DEFEITOS-INDIVIDUAL e o tempo médio de experiência em meses do indivíduo, representado pela variável MÉDIA-EXPERIÊNCIA-INSPECTOR. As duas variáveis foram utilizadas para testar a hipótese 2.

Quadro 5.4 - Tempo Médio de Experiência/ Número de Defeitos

Inspeção Individual / Experiência das Equipes										
Técnicas/ Variáveis	NRO-DEFEITOS- INDIVIDUAL	MÉDIA- EXPERIÊNCIA- INSPECTOR								
LBCh	12	80	14	20	9	60	18	80	11	80
LBCe	11	80	12	80	14	6	13	80	12	80
LBPe	17	40	13	40	10	40	15	80	16	20

5.2 ANÁLISE DA HIPÓTESE 1 COM O TESTE KRUSKAL WALLIS E TUKEY-KRAMER

A Tabela 5.1 apresenta os resultados do teste estatístico Kruskal Wallis, referente às medianas obtidas em relação aos tipos de defeitos detectados pelos inspetores, seguindo a classificação de defeitos empregada, através das técnicas de inspeção utilizadas.

Tabela 5.1 - Resultado do Teste de Kruskal Wallis na Inspeção entre as Equipes, por Técnica e Tipo de Defeito

Técnicas/ Tipos de Defeitos	LBCh	LBCe	LBPe	Valor p
Comissão	8	6,8	9,2	0,692662
Omissão	6,4	6,5	11,1	0,114092
FO	7	5,8	11,2	0,035084 ^{*2}
PO	4	10	10	0,006738 ^{*3}
IO	12	6	6	0,006152 ^{*4}
AO	7,0	8,5	8,5	0,367879
IA	6,3	7,7	10	0,33024
II	11,2	6,2	6,6	0,129126
FI	6	9	9	0,116037
SE	6,1	7	10,9	0,119496

A Tabela 5.1 mostra que existem evidências para a rejeição de H₀, para os tipos de defeitos FO, PO e IO, no resultado obtido através do teste de Kruskal Wallis. Neste caso, conforme destacado no capítulo 4, procedeu-se à análise por comparações múltiplas, usando o procedimento de Tukey-Kramer, a fim de verificar em quais técnicas de leitura LBCh, LBCe ou LBPe, elas se diferem.

Associando-se a análise da Tabela 5.1 com o Gráfico 5.1, observou-se que a técnica LBPe apresentou uma diferença mais acentuada para a detecção dos defeitos das classes adotadas no estudo, principalmente para os tipos de defeitos da classe Omissão, comparando-se com as técnicas LBCh e LBCe que também apresentaram evidências para detectar defeitos dessa classe.

^{2,3,4} * Resultado do valor p apresentado na Tabela 5.1, indica uma diferença significativa comparada com o nível de significância de 5%.

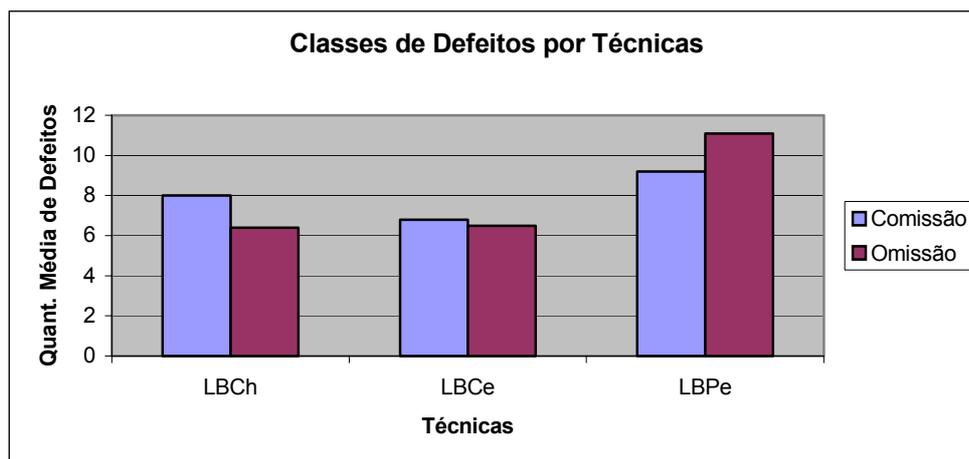


Gráfico 5.1 - Classificação de Defeitos por Técnica

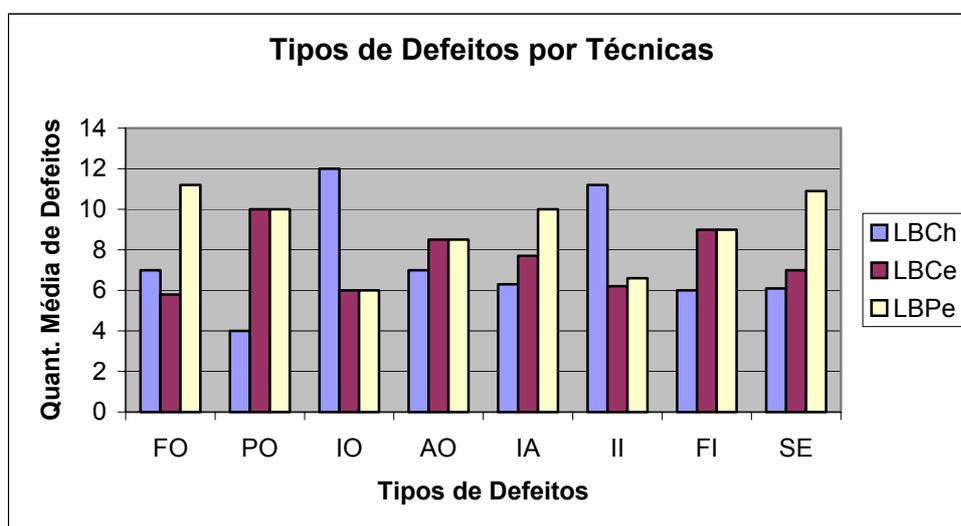


Gráfico 5.2 - Tipos de Defeitos por Técnica

Associando-se a análise da Tabela 5.1 com o Gráfico 5.2, observou-se que a técnica LBPe mostrou-se eficiente na detecção dos tipos de defeitos FO (Funcionalidade Omitida) e PO (Performance Omitida). Na análise do tipo de defeito FO, a técnica LBPe mostrou superioridade na detecção desse tipo de defeito.

No defeito PO, as técnicas LBPe e LBCe apresentaram um resultado significativo na detecção desse tipo de defeito, enquanto que a técnica LBCh não apresentou evidências significativas para esse tipo.

A técnica LBCh mostrou-se eficiente para a detecção de defeito do tipo IO (Interface Omitida), enquanto que as técnicas LBPe e LBCe não apresentaram um resultado significativo para esse tipo.

5.2.1 ANÁLISE DA CLASSIFICAÇÃO DE DEFEITOS POR COMISSÃO E OMISSÃO

As Tabelas 5.2 e 5.3 apresentam os resultados do teste estatístico Kruskal Wallis, referentes às classes de defeitos (Comissão e Omissão) detectados pelos inspetores, através das técnicas de inspeção LBCh, LBCe e LBPe.

Tabela 5.2 - Resultado do Teste Kruskal Wallis na Inspeção, por Técnica, na Classe de Defeitos “Comissão”

Técnicas	Tamanho da Amostra	Medianas
LBCe	5	6,8
LBCh	5	8,0
LBPe	5	9,2
Estatística do Teste = 0,734426		Valor p = 0,692662

Conforme a Tabela 5.2, observou-se que não existem diferenças significativas entre as técnicas LBCh, LBCe e LBPe, na detecção de defeitos da classe Comissão, na inspeção realizada.

Tabela 5.3 - Resultado do Teste de Kruskal Wallis na Inspeção, por Técnica, na Classe de Defeitos “Omissão”

Técnicas	Tamanho da Amostra	Medianas
LBCe	5	6,5
LBCh	5	6,4
LBPe	5	11,1
Estatística do Teste = 4,34151		Valor p = 0,114092

Para a classe de defeitos Omissão, observou-se na Tabela 5.3 que não existem diferenças significativas entre as técnicas LBCh, LBCe e LBPe, na detecção de defeitos dessa classe, na inspeção realizada.

Portanto, considerando-se a classificação de defeitos adotada no estudo, que inclui as classes Comissão e Omissão, verificou-se, com o auxílio da análise estatística, que não há diferença significativa entre as técnicas de inspeção LBCh, LBCe e LBPe nos defeitos detectados pelos inspetores.

5.2.2 ANÁLISE DOS TIPOS DE DEFEITOS POR COMISSÃO

Nas tabelas 5.4 a 5.7 foram analisados os defeitos da classe Comissão referentes aos tipos IA, II, FI e SE, usados para detectar os defeitos.

Tabela 5.4 - Teste de Kruskal Wallis para Detecção de Defeitos IA por Técnicas

Técnicas	Tamanho da Amostra	Medianas
LBCe	5	7,7
LBCh	5	6,3
LBPe	5	10,0
Estatística do Teste = 2,21587		Valor p = 0,33024

A partir da Tabela 5.4, observou-se que não existem diferenças significativas entre as técnicas LBCh, LBCe e LBPe na detecção de defeitos do tipo IA.

Tabela 5.5 - Teste de Kruskal Wallis para Detecção de Defeitos II por Técnicas

Técnicas	Tamanho da Amostra	Medianas
LBCe	5	6,2
LBCh	5	11,2
LBPe	5	6,6
Estatística do Teste = 4,09394		Valor p = 0,129126

Na Tabela 5.5, observou-se que não existem diferenças significativas entre as técnicas LBCh, LBCe e LBPe na detecção de defeitos do tipo II.

Tabela 5.6 - Teste de Kruskal Wallis para Detecção de Defeitos FI por Técnicas

Técnicas	Tamanho da Amostra	Medianas
LBCe	5	9,0
LBCh	5	6,0
LBPe	5	9,0
Estatística do Teste = 4,30769		Valor p = 0,116037

Na Tabela 5.6, observou-se que não existem diferenças significativas entre as técnicas LBCh, LBCe e LBPe, na detecção de defeitos do tipo FI.

Tabela 5.7 - Teste de Kruskal Wallis para Detecção de Defeitos SE por Técnicas

Técnicas	Tamanho da Amostra	Medianas
LBCe	5	7,0
LBCh	5	6,1
LBPe	5	10,9
Estatística do Teste = 4,24895		Valor p = 0,119496

A partir da Tabela 5.7, observou-se que não existem diferenças significativas entre as técnicas LBCh, LBCe e LBPe, na detecção de defeitos do tipo SE.

5.2.3 ANÁLISE DOS TIPOS DE DEFEITOS POR OMISSÃO

Nas tabelas 5.8 a 5.14 foram analisados os defeitos da classe Omissão, referentes aos tipos FO, AO, IO e PO, usados para detectar os defeitos no estudo.

Tabela 5.8 - Teste de Kruskal Wallis para Detecção de Defeitos FO por Técnicas

Técnicas	Tamanho da Amostra	Medianas
LBCe	5	5,8
LBCh	5	7,0
LBPe	5	11,2
Estatística do Teste = 6,7		Valor p = 0,0350844

A partir da Tabela 5.8, observou-se que existem diferenças significativas entre as técnicas LBCh, LBCe e LBPe na detecção de defeitos do tipo FO. Com esse resultado, procedeu-se à análise das comparações múltiplas entre médias.

Tabela 5.9 - Comparação pelo Método de Tukey (HSD⁵) ao nível de 0,05 de significância com o valor absoluto das médias dos dados do tipo FO

Técnicas	Tamanho da Amostra	Valor Absoluto Médias	Grupos Homogêneos
LBCe	5	2,8	X
LBCh	5	3,0	XX
LBPe	5	3,6	X
Comparação	Diferenças - HSD	Limites (+/-)	
LBCe - LBCh	(2,8-3,0)= -0,2	0,690733	
LBCe - LBPe	(2,8-3,6)= -0,8*	0,690733	
LBCh - LBPe	(3,0-3,6)= -0,6	0,690733	

* denota-se uma diferença estatisticamente significativa

Na tabela 5.9, dentre as técnicas analisadas e avaliadas, pode-se concluir que a técnica LBPe foi eficiente na detecção de defeitos do tipo FO, comparando-se principalmente com a técnica LBCe, que demonstra evidências de diferenças entre essas técnicas.

⁵ HSD a sigla encontrada em programas de computador representa a diferença honestamente significativa (em inglês, *honestly significant difference*)

Tabela 5.10 - Teste de Kruskal Wallis para Detecção de Defeitos PO por Técnicas

Técnicas	Tamanho da Amostra	Medianas
LBCe	5	10,0
LBCh	5	4,0
LBPe	5	10,0
Estatística do Teste = 10,0		Valor p = 0,00673795

Na Tabela 5.10, observou-se que existem diferenças significativas entre as técnicas LBCh, LBCe e LBPe na detecção de defeitos do tipo PO. Com esse resultado, procedeu-se à análise das comparações múltiplas entre médias.

Tabela 5.11 - Comparação pelo Método de Tukey (HSD) ao nível de 0,05 de significância com o valor absoluto das médias dos dados do tipo PO

Técnicas	Tamanho da Amostra	Valor Absoluto Médias	Grupos Homogêneos
LBCh	5	1,0	X
LBCe	5	2,0	X
LBPe	5	2,0	X
Comparação	Diferenças - HSD	Limites (+/-)	
LBCe - LBCh	$(2,0 - 1,0) = 1,0^*$	0,690733	
LBCe - LBPe	$(2,0 - 2,0) = 0,0$	0,690733	
LBCh - LBPe	$(1,0 - 2,0) = -1,0^*$	0,690733	

* denota-se uma diferença estatisticamente significativa

Na Tabela 5.11, observou-se que as técnicas LBCe e LBPe mostraram-se significativas em relação à técnica LBCh, na detecção do tipo de defeito PO.

Tabela 5.12 - Teste de Kruskal Wallis para Detecção de Defeitos IO por Técnicas

Técnicas	Tamanho da Amostra	Medianas
LBCe	5	6,0
LBCh	5	12,0
LBPe	5	6,0
Estatística do Teste = 10,1818		Valor p = 0,00615242

Na Tabela 5.12, observou-se que existem diferenças significativas entre as técnicas LBCh, LBCe e LBPe, na detecção de defeitos do tipo IO. Com esse resultado, procedeu-se a análise das comparações múltiplas entre médias.

Tabela 5.13 - Comparação pelo Método de Tukey (HSD) ao nível de 0,05 de significância com o valor absoluto das médias dos dados do tipo IO

Técnicas	Tamanho da Amostra	Valor Absoluto Médias	Grupos Homogêneos
LBCe	5	0,0	X
LBPe	5	0,0	X
LBCh	5	0,8	X
Comparação	Diferenças - HSD	Limites (+/-)	
LBCe - LBCh	(0,0-0,8)= -0,8*	0,436858	
LBCe - LBPe	(0,0-0,0)= 0,0	0,436858	
LBCh - LBPe	(0,8-0,0)= 0,8*	0,436858	

* denota-se uma diferença estatisticamente significativa

Na tabela 5.13 dentre as técnicas analisadas e avaliadas, pode-se concluir que a técnica LBCh mostrou-se eficiente na detecção dos defeitos do tipo IO, comparando-se com as técnicas LBPe e LBCe.

Tabela 5.14 - Teste de Kruskal Wallis para Detecção de Defeitos AO por Técnicas

Técnicas	Tamanho da Amostra	Medianas
LBCe	5	8,5
LBCh	5	7,0
LBPe	5	8,5
Estatística do Teste = 2,0		Valor p = 0,367879

Na Tabela 5.14, observou-se que não existem diferenças significativas entre as técnicas LBCh, LBCe e LBPe, na detecção de defeitos do tipo AO.

Dos tipos de defeitos analisados na classificação por Omissão, a técnica LBPe mostrou-se através da análise estatística, ser eficiente na detecção de defeitos dos tipos FO e PO. A técnica LBCh mostrou ser eficiente na detecção de defeitos do tipo IO e a técnica LBCe no tipo PO.

5.3 ANÁLISE DA HIPÓTESE 2 COM O TESTE DE CORRELAÇÃO DE PEARSON

A análise desta hipótese foi subsidiada pelos resultados do teste de Correlação Linear de Pearson, analisado a partir das variáveis NRO-DEFEITOS-INDIVIDUAL e MÉDIA-EXPERIÊNCIA-INSPEÇÃO, apresentado no Quadro 5.5.

Quadro 5.5 - Correlação Linear – Número de Defeitos Detectados X Tempo Médio de Experiência dos Inspetores

Inspeção Individual / Experiência das Equipes										
Técnicas	NRO-DEFEITOS-INDIVIDUAL	MÉDIA-EXPERIÊNCIA-INSPECTOR								
LBCh	12	80	14	20	9	60	18	80	11	80
LBCe	11	80	12	80	14	6	13	80	12	80
LBPe	17	40	13	40	10	40	15	80	16	20
Estat. t = - 0.6097 gl = 13 valor p = 0.5526 cor= -0.1667329										

O resultado obtido no coeficiente de correlação linear mostra que não há evidências para rejeição de H_0 , indicando uma fraca correlação linear entre o número de defeitos detectados na inspeção, através das técnicas LBCh, LBCe e LBPe, e o tempo médio de experiência dos inspetores como analista de sistemas ou engenheiros de software. O teste paramétrico da correlação confirmou que a experiência dos inspetores não apresentou influência na quantidade de defeitos detectados pelas técnicas LBCh, LBCe e LBPe, utilizadas na inspeção.

Pelo Gráfico 5.3, pode-se observar que os pontos do Diagrama de Dispersão não apresentam aspecto linear, ou seja, sugerem que o Número de Defeitos Detectados pelos Inspetores e o Tempo Médio de Experiência dos Inspetores não estão correlacionados. Conclui-se que não existe nenhuma correlação significativa entre a experiência dos inspetores e os resultados obtidos em relação ao número de defeitos detectados pelos inspetores.

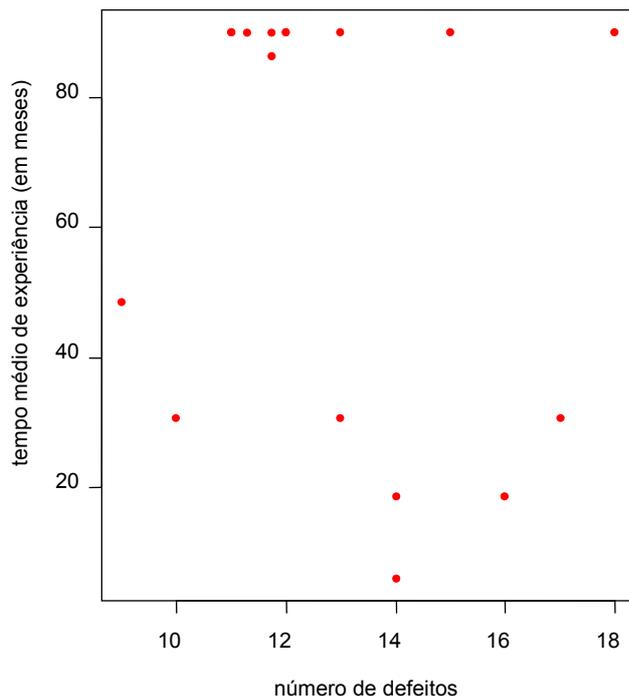


Gráfico 5.3 - Diagrama de Dispersão

5.4 ANÁLISE DO FORMULÁRIO DE ASPECTOS GERAIS

Após a realização do estudo empírico, os participantes avaliaram o estudo respondendo a um questionário com informações pessoais e questões referentes ao estudo objetivando uma avaliação das técnicas empregadas. Os apêndices C1 e C2 contêm os dados apurados.

a) Informações Pessoais.

- Tempo de Experiência.

O Gráfico 5.4 avalia o aspecto de experiência na área de informática na técnica LBCh, indicando que 60% dos participantes atuam na área há mais de 60 meses, 20% entre 37 a 60 meses e 20% entre 13 a 24 meses.



Gráfico 5.4 - Tempo de experiência dos participantes na técnica LBCh

Através do Gráfico 5.5, pode-se observar que no quesito de experiência na área de informática na técnica LBCe 80% dos participantes atuam na área há mais de 60 meses e 20% até 12 meses.



Gráfico 5.5 - Tempo de experiência dos participantes na técnica LBCe

O Gráfico 5.6, mostrou que, com relação a experiência na área de informática na técnica LBPe 60% dos participantes atuam na área entre 25 a 36 meses, 20% há mais de 60 meses e 20% entre 13 a 24 meses.



Gráfico 5.6 - Tempo de experiência dos participantes na técnica LBPe

Analisando o perfil da amostra nas técnicas empregadas observou-se que mais da metade dos participantes atuam na área há mais de 60 meses, visto que, esse é um dado importante para o estudo realizado.

- Nível de Formação.

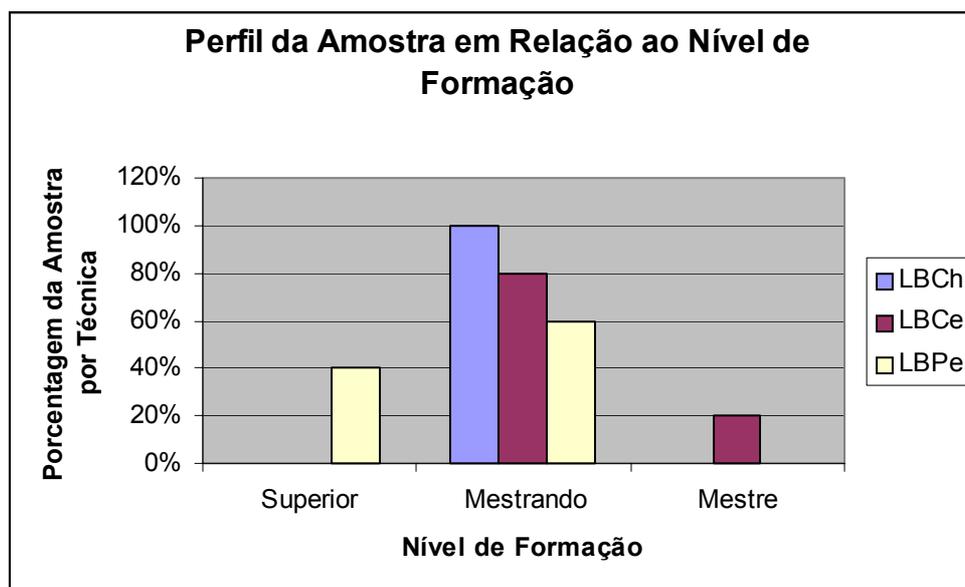


Gráfico 5.7 - Nível de formação dos participantes

Analisando o perfil da amostra quanto ao nível de formação nas técnicas empregadas, o Gráfico 5.7 demonstrou que 70% dos participantes são mestrandos na área de Ciência da Computação e 20% possuem graduação na área de informática e 10% tem nível de mestrado.

- Experiência em inspeção de software.

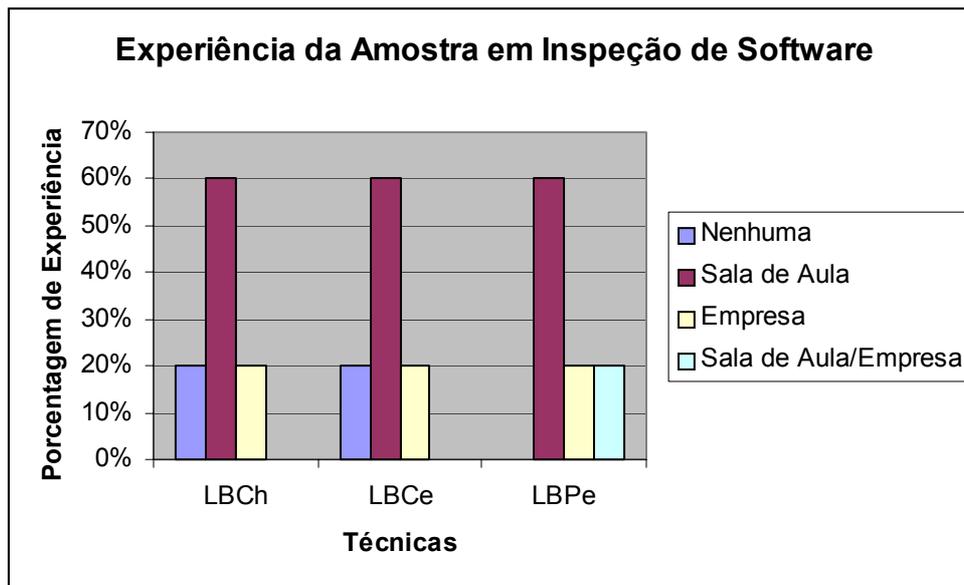


Gráfico 5.8 - Experiência dos participantes em inspeção

Analisando o gráfico 5.8, quanto à experiência da amostra sobre inspeção de software, observou-se que na área de inspeção 70% dos participantes tiveram conhecimento do método de inspeção em sala de aula, 10% trabalharam com inspeção na empresa, 10% não tinham conhecimento da técnica e 10% tiveram conhecimento na sala de aula e na empresa.

- Conhecimento nas técnicas de inspeção.

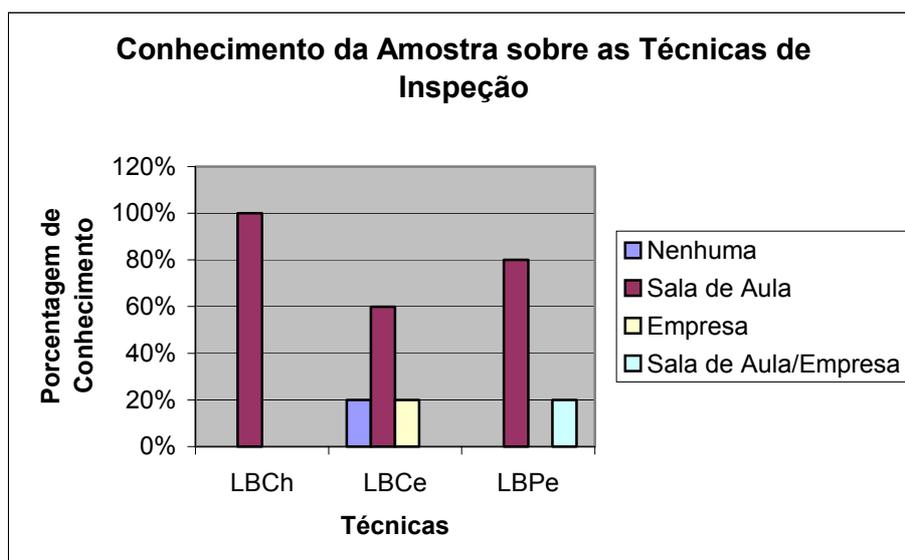


Gráfico 5.9 - Conhecimento dos participantes nas técnicas de inspeção

O gráfico 5.9 analisa o conhecimento da amostra sobre as técnicas de inspeção adotadas no estudo, e destacou que 70% dos participantes tiveram conhecimento das técnicas de inspeção em sala de aula, 10% trabalharam com alguma técnica na empresa, 10% na empresa e na sala de aula e 10% não tinham conhecimento das técnicas.

b) Questões do Estudo.

Nas questões respondidas pelos participantes foram realizados os seguintes levantamentos referentes ao perfil da amostra.

- Conhecimento sobre inspeção. No Gráfico 5.10, a amostra apresentou um conhecimento regular sobre o método de inspeção.

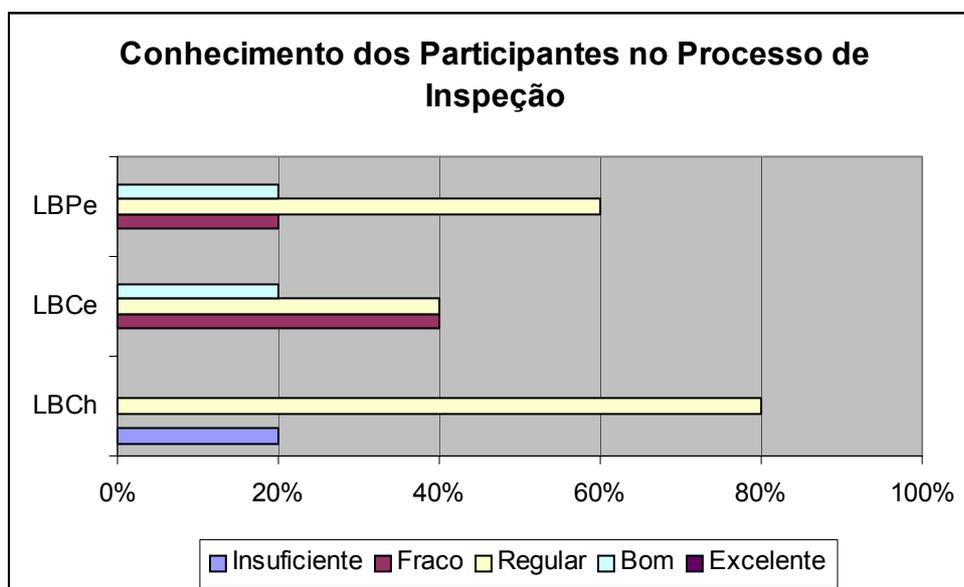


Gráfico 5.10 - Conhecimento dos participantes sobre inspeção

- Conhecimento sobre documentos de requisitos de software. No Gráfico 5.11, a amostra apresentou um conhecimento regular em escrever e ler documentos de requisitos de software.

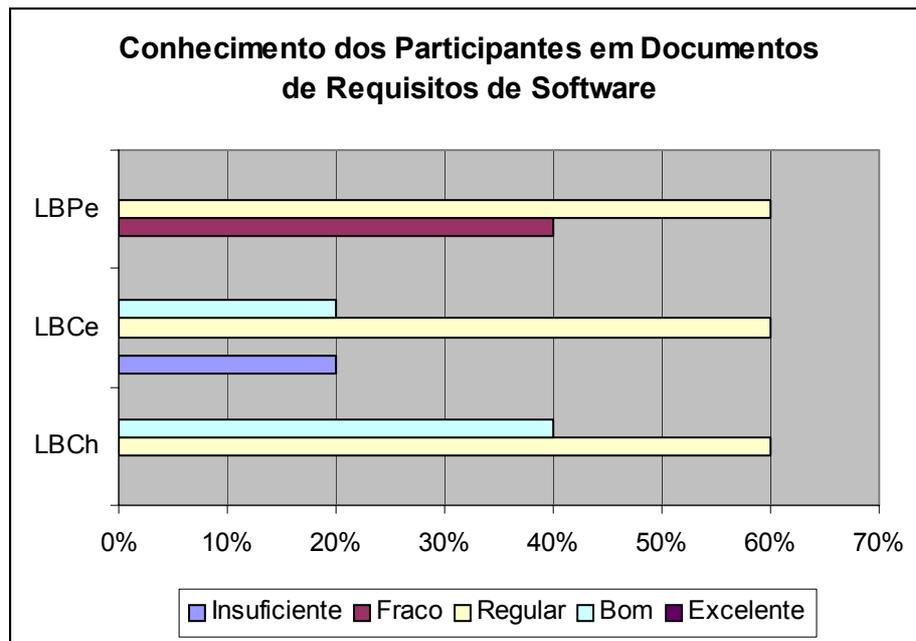


Gráfico 5.11 - Conhecimento dos participantes em DRS

- Classificação de defeitos. No Gráfico 5.12, a amostra declarou a classificação de defeitos (Comissão e Omissão) usada no estudo como boa no quesito de fácil entendimento.

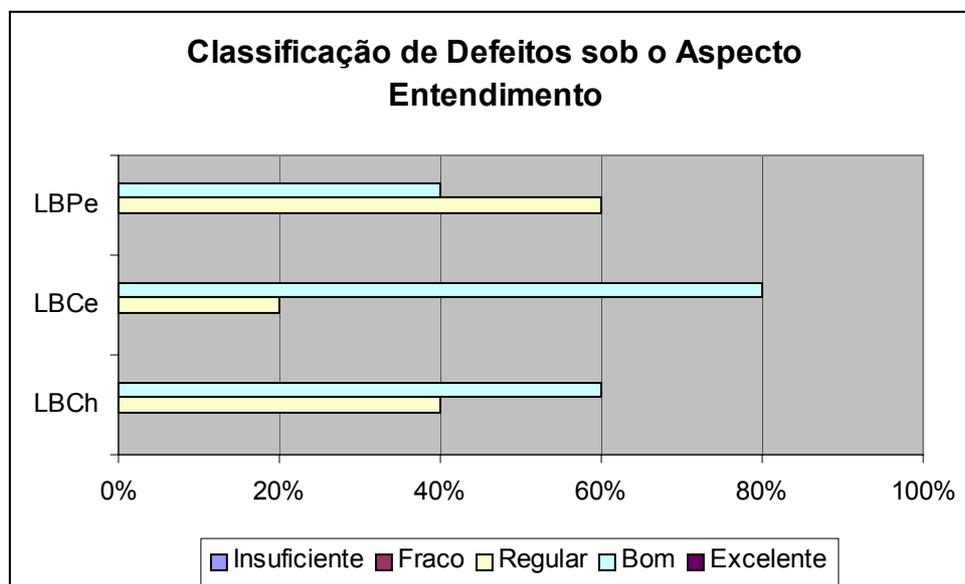


Gráfico 5.12 - Avaliação da classificação de defeitos – entendimento

- Classificação de defeitos. No Gráfico 5.13, a amostra declarou a classificação de defeitos (Comissão e Omissão) usada no estudo como regular no quesito de estar completo.

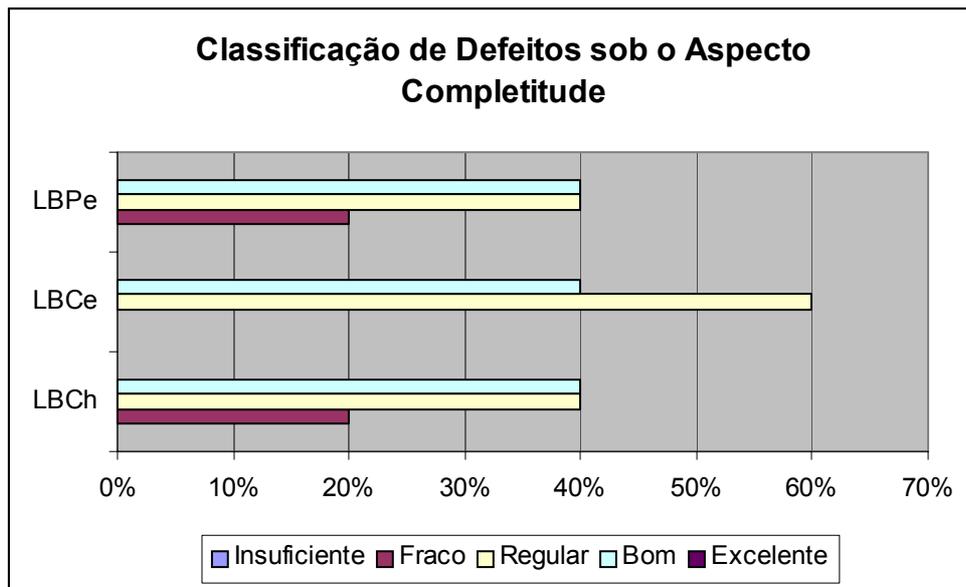


Gráfico 5.13 - Avaliação da classificação de defeitos - completa

- Tipo de defeitos. No Gráfico 5.14, a amostra declarou o tipo de defeitos usado no estudo como boa no quesito de fácil entendimento.

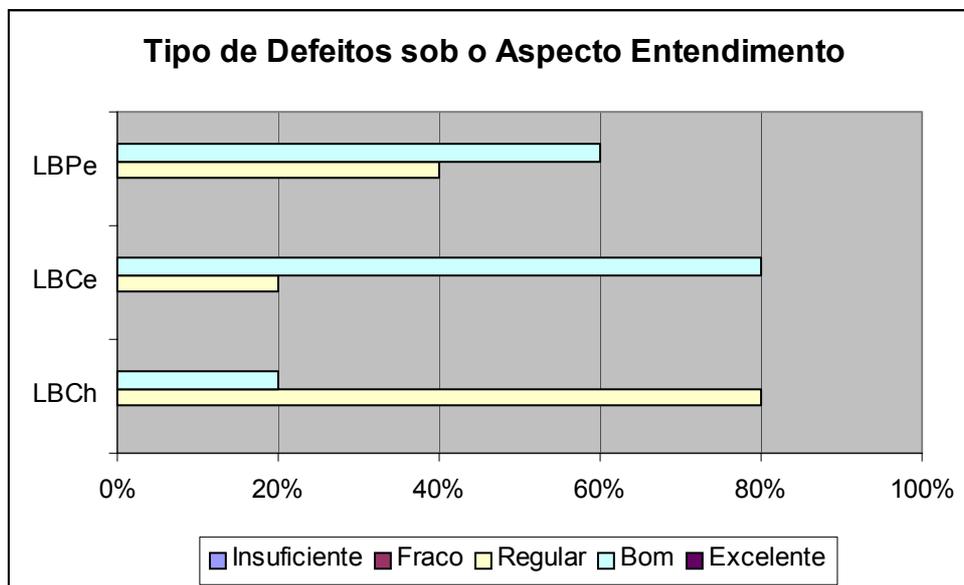


Gráfico 5.14 - Avaliação do tipo de defeitos - entendimento

- Tipo de defeitos. No Gráfico 5.15, a amostra declarou o tipo de defeitos usado no estudo como bom no quesito de estar completo.

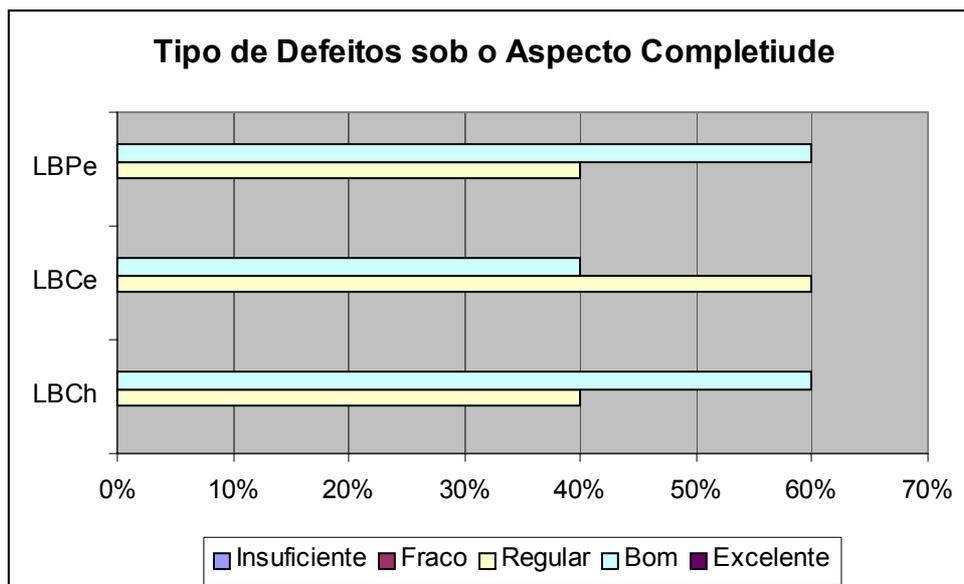


Gráfico 5.15 - Avaliação do tipo de defeitos – completo

- Formulário de inspeção. No Gráfico 5.16, a amostra declarou o formulário de inspeção usado no estudo para preenchimento dos defeitos como bom no quesito de fácil entendimento.

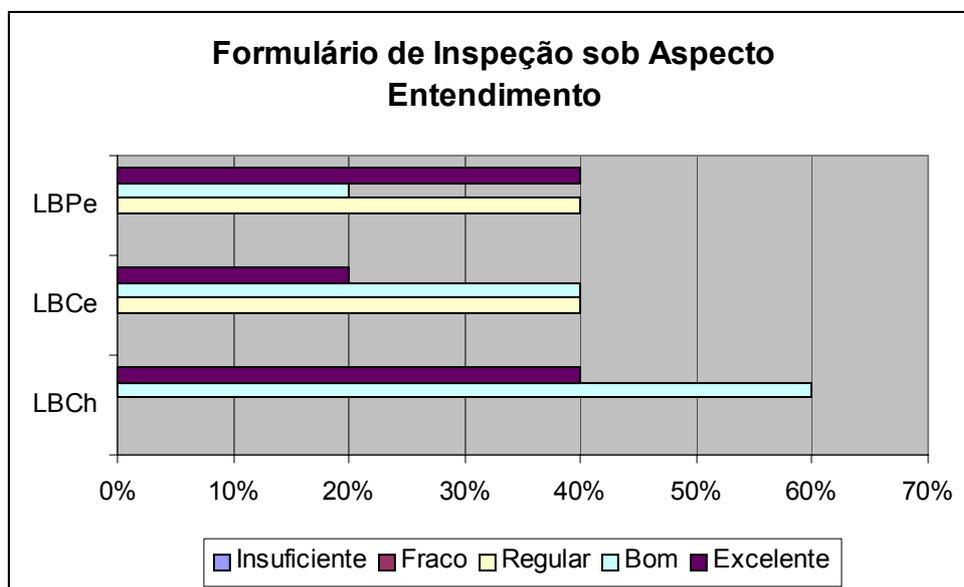


Gráfico 5.16 - Avaliação do formulário de inspeção - entendimento

- Formulário de detecção de defeitos. No Gráfico 5.17, os participantes da técnica LBCh declararam o formulário de detecção usado para guiar os inspetores na localização dos defeitos no estudo como bom no quesito

entendimento. Os participantes da técnica LBCe declararam o formulário como bom, e os participantes da técnica LBPe declararam como bom. No geral os formulários tiveram uma classificação boa no quesito entendimento.

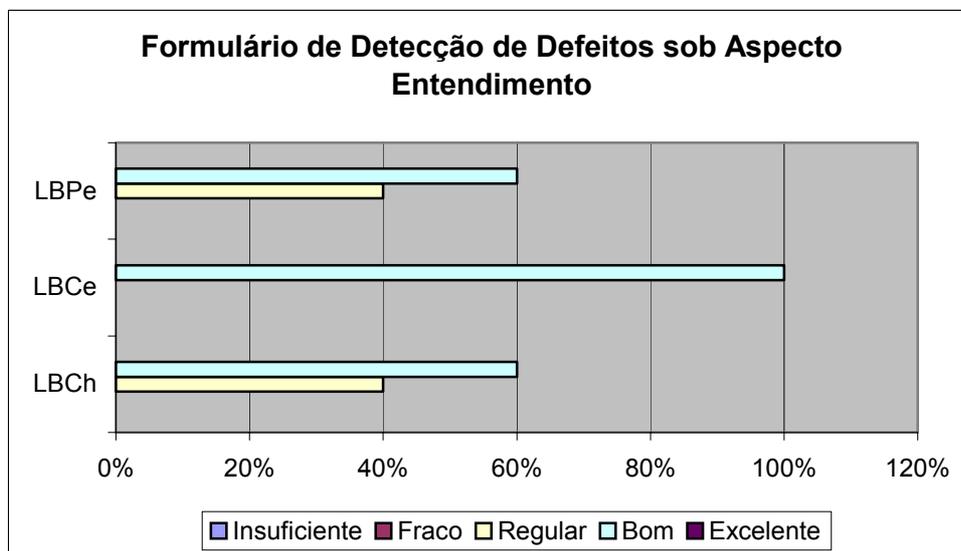


Gráfico 5.17 - Avaliação detecção de defeitos – entendimento

- Formulário de detecção de defeitos. No Gráfico 5.18, os participantes da técnica LBCh declararam as questões do formulário de detecção usado para guiar os inspetores na localização dos defeitos no estudo como bom e excelente. Os participantes da técnica LBCe declararam as questões do formulário como bom, e os participantes da técnica LBPe declararam as questões usadas na busca dos defeitos como bom. No geral os formulários apresentaram uma classificação boa no quesito das questões usadas nos formulários.

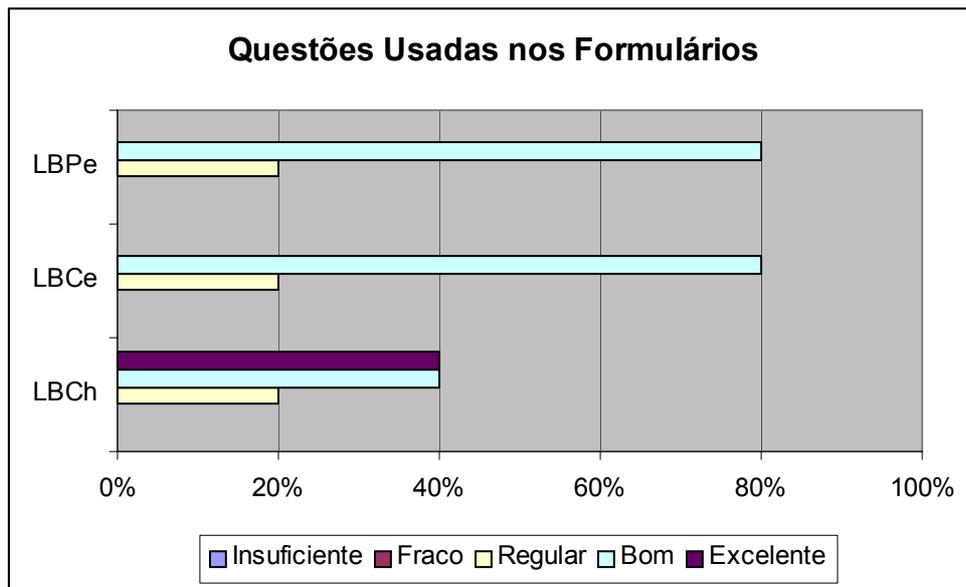


Gráfico 5.18 - Avaliação das questões de detecção de defeitos

- Artefatos usados na inspeção. No Gráfico 5.19, a amostra declarou os artefatos (formulários e documentos) usados na inspeção como bons no quesito de serem suficientes para a realização do estudo.

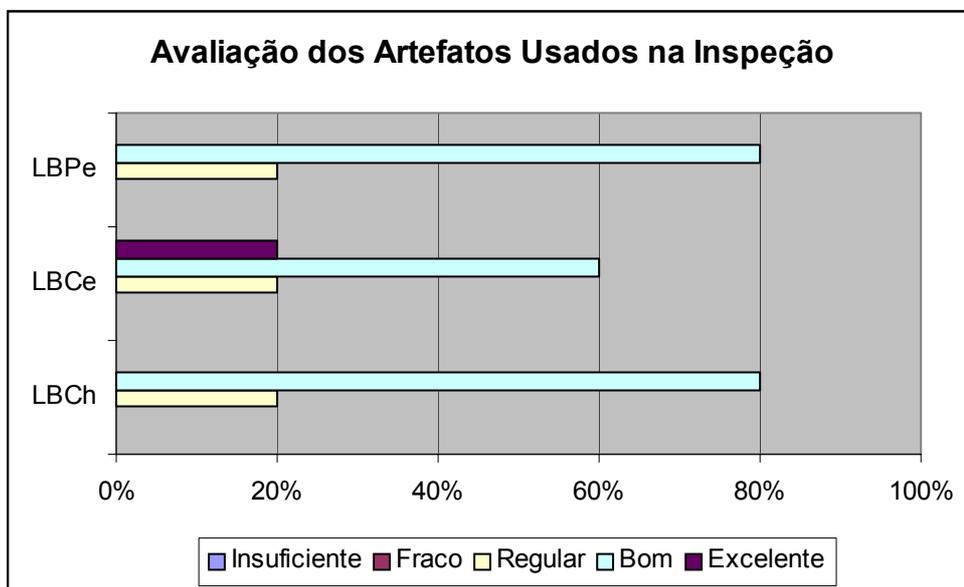


Gráfico 5.19 - Avaliação dos artefatos usados na inspeção

- Técnica de inspeção. No Gráfico 5.20, a amostra declarou a técnica de inspeção como boa no quesito de fácil entendimento.

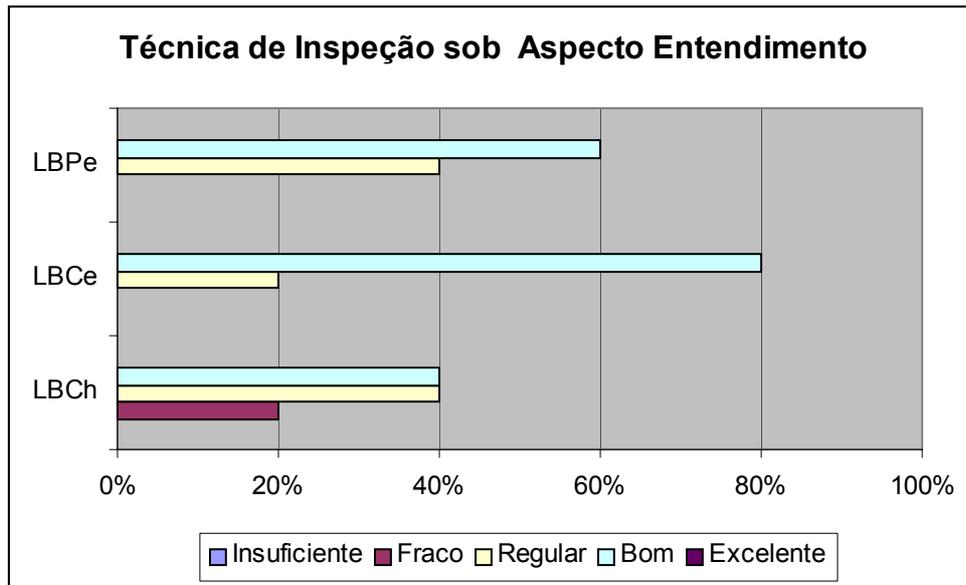


Gráfico 5.20 - Avaliação da técnica de inspeção – entendimento

- Conhecimentos adquiridos com a técnica de inspeção. No Gráfico 5.21, a amostra declarou ter adquirido um bom conhecimento no processo de inspeção e nas técnicas de leitura (LBCh, LBCe e LBPc) durante a participação no estudo.

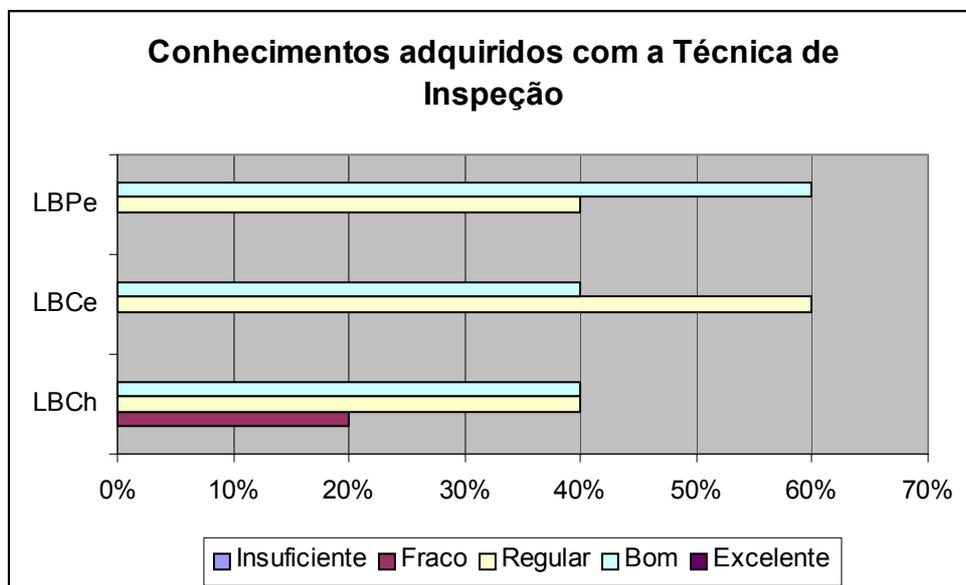


Gráfico 5.21 - Conhecimentos adquiridos na técnica de inspeção

- Uso da inspeção no ambiente de trabalho. No Gráfico 5.22, a amostra declarou como regular o uso da inspeção no ambiente de trabalho, no projeto de desenvolvimento de software.

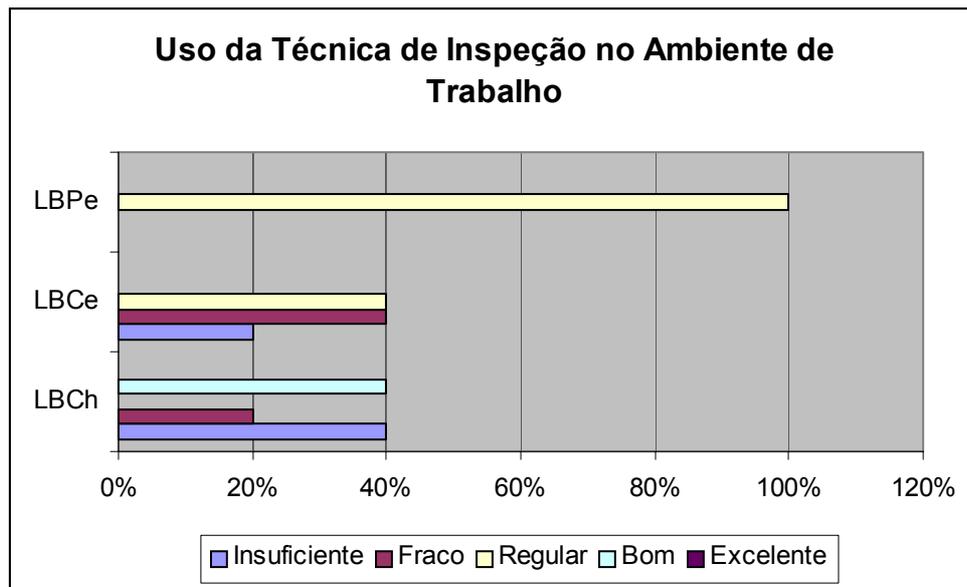


Gráfico 5.22 - Uso da inspeção no ambiente de trabalho

c) Comentários dos Participantes

- De um modo geral, os participantes declararam que o tempo utilizado para inspecionar o produto foi pequeno;
- Os participantes sentiram a necessidade de um treinamento nas técnicas utilizadas no estudo, o que garantiria um melhor conhecimento nas técnicas para realizar a busca pelos defeitos de acordo com a classificação de defeitos usada.

5.5 CONSIDERAÇÕES FINAIS

Este trabalho foi realizado com o objetivo de comparar a eficiência entre as técnicas LBCH, LBCe e LBPe, no que diz respeito ao número de defeitos detectados, na inspeção do DRS de um sistema real, utilizado por clínicas ocupacionais, denominado como PCMSO – Programa de Controle Médico e Saúde Ocupacional.

Para realizar a inspeção, foi traçada uma amostra por conveniência totalizando 15 profissionais graduados, com experiência na área de Análise de Sistemas e/ou Engenharia de Software.

Para comparar a eficiência das técnicas LBCh, LBCe e LBPe, foi realizada uma análise estatística para testar as hipóteses formuladas no estudo empírico.

A Hipótese 1 foi testada para avaliar a eficiência das técnicas LBCH, LBCe e LBPe em relação à quantidade de defeitos detectados pelas equipes de inspetores através das variáveis NRO-DEFEITOS-LBCH, NRO-DEFEITOS-LBCE e NRO-DEFEITOS-LBPE.

A Hipótese 2 foi subsidiada com dados obtidos no Formulário de Aspectos Gerais, traçando o perfil da amostra, associando-se as variáveis referentes a NRO-DEFEITOS-INDIVIDUAL e MÉDIA-EXPERIÊNCIA-INSPECTOR. Essa hipótese mostrou que a análise realizada na hipótese 1 não foi influenciada pelas variáveis NRO-DEFEITOS-INDIVIDUAL e MÉDIA-EXPERIÊNCIA-INSPECTOR.

Os principais resultados obtidos nas hipóteses testadas no estudo são sintetizados a seguir.

a) Análise da técnica LBPe

A técnica LBPe (Leitura Baseada em Perspectivas) mostrou superioridade, comparada à técnica LBCh, na detecção do tipo de defeito PO (Performance Omitida) e com a técnica LBCe na detecção do defeito FO (Funcionalidade Omitida), pertencentes à classe Omissão. É importante salientar que, nesse estudo, foi adotada a visão do projetista, tendo como objetivo auxiliar os inspetores na utilização de técnicas de especificação para inspecionar o documento.

b) Análise da técnica LBCe

A técnica LBCe (Leitura Baseada em Cenários) mostrou superioridade, comparada à técnica LBCh, na detecção do tipo de defeito PO (Performance Omitida), pertencente à classe Omissão.

c) Análise da técnica LBCh

A técnica LBCh (Leitura Baseada em *Checklists*) mostrou superioridade, comparada às técnicas LBCe e LBPe, na detecção do tipo de defeito IO (Interface Omitida), pertencente à classe Omissão.

d) Análise da experiência dos inspetores e do número de defeitos detectados

Observou-se que a relação entre o número de defeitos detectados pelos inspetores e a experiência dos inspetores foi fraca. Os participantes com mais experiência não executaram melhor a inspeção do que os participantes com menos experiência.

e) Comparação com Estudos Anteriores

- No estudo realizado por PORTER (1994), o autor comparou a eficiência das técnicas de leitura *Ad Hoc*, LBCh e LBCe e o resultado do estudo apresentou a técnica LBCe como mais eficiente em relação as outras, nesse estudo conclui-se que as técnicas LBCh e LBCe não apresentaram um resultado significativo quando comparadas. Ficou determinado no estudo que a experiência dos autores não teve nenhuma influência significativa. O mesmo ocorreu no presente estudo.
- Na replicação de DÓRIA (2001), foram usadas as técnicas *Ad Hoc*, LBCh e LBPe e confirmou-se um resultado significativo da técnica LBPe comparada com as técnicas *Ad hoc* e LBCh. Esse resultado também foi confirmado no presente estudo, com a técnica LBPe.
- No estudo de ABIB (1998), foram comparadas as técnicas *Ad Hoc*, LBCh e LBCe e observou-se que a técnica LBCe foi mais eficiente na detecção dos defeitos, comparado com as outras técnicas. No presente estudo as técnicas LBCh e LBCe apresentaram um resultado significativo em alguns tipos de defeitos, visto que a inspeção apresentou um resultado diferente em relação a classe de defeito

omissão por ter observado um resultado mais significativo em relação ao estudo de ABIB (1998) que demonstrou com a classe comissão.

- No estudo realizado por CIOLKOWSKI (1999), o autor comparou as técnicas LBCh e LBPe e no resultado foi confirmada uma diferença significativa entre as técnicas LBCh e LBPe que se mostrou mais eficiente. No presente estudo, foram confirmadas algumas diferenças nessas técnicas em relação a alguns tipos de defeitos.
- Os resultados obtidos no estudo empírico realizado por BERLING (2004), confirmaram uma diferença significativa entre a quantidade de defeitos localizados entre a Técnica de Leitura Baseada em *Checklist* (LBCh) e a Técnica de Leitura Baseada em Perspectivas (LBPe), que demonstrou ser mais eficiente. No presente estudo, observou-se o mesmo resultado, pois a técnica LBPe foi mais significativa na detecção dos defeitos, se comparada com a técnica LBCh.

Sendo assim, dos estudos analisados anteriormente, destacaram-se alguns fatores importantes na inspeção: em relação ao domínio de conhecimento dos participantes do documento que foi inspecionado, o treinamento realizado com os participantes e o tempo destinado para a realização da inspeção. Em relação às técnicas utilizadas nos estudos citados, conclui-se que as técnicas LBPe e LBCe apresentaram resultados mais significativos durante as inspeções realizadas. Isto se deve ao fato das técnicas fornecerem mais detalhes, ou seja, serem mais direcionadas na busca dos defeitos. Isto ocorreu também no presente estudo.

Em relação ao estudo empírico aqui relatado, observou-se a necessidade de treinamento para os participantes antes da realização da inspeção do estudo proposto. Ficou evidente também a necessidade de um tempo maior para a realização da inspeção.

Das técnicas analisadas no presente estudo empírico, observou-se, através da análise estatística, que a técnica LBPe detectou um número maior de inconsistências, ou seja, foi mais significativa na detecção dos defeitos no documento do sistema de PCMSO. Isto foi observado também no estudo

realizado pelos autores CIOLKOWSKI (1999), DÓRIA (2001) e BERLING (2004) nos quais a técnica LBPe foi mais significativa na detecção dos defeitos.

No próximo capítulo serão apresentadas as conclusões e contribuições deste projeto de mestrado e algumas idéias para trabalhos futuros.

CAPÍTULO 6

CONCLUSÃO

6.1 RESUMO DA PESQUISA

Esta dissertação de mestrado teve por objetivo estudar o método de inspeção e as técnicas de leitura *Checklists*, Cenários e Perspectivas enfocando o emprego das três técnicas através de um estudo empírico.

O estudo foi realizado em um Documento de Requisitos de Software do Sistema “Programa de Controle Médico e Saúde Ocupacional” PCMSO (2004), apresentado de acordo com o padrão IEEE ANSI/IEEE (1998). Buscou-se avaliar a eficiência das técnicas de inspeção na detecção dos defeitos.

As três técnicas de inspeção (LBCh, LBCe e LBPe) foram estudadas e o documento de requisitos do PCMSO foi obtido junto à empresa que desenvolveu o sistema. Além disso, o material para realização da inspeção, compreendendo formulários para aplicação de cada técnica, foi preparado.

O estudo empírico teve a participação de 15 profissionais da área de análise de sistemas e engenharia de software, que atuaram como inspetores. O projeto da pesquisa incluiu a definição de hipóteses, testes empregados na análise dos dados e interpretação dos resultados.

O objetivo do estudo foi comparar a eficiência das técnicas LBCH, LBCe e LBPe, em relação ao número de defeitos detectados pelos indivíduos e o tempo médio de experiência na área, a partir das hipóteses formuladas no estudo. Foi realizada uma análise estatística para testar as hipóteses.

A Hipótese 1 foi testada para avaliar a eficiência das técnicas LBCh, LBCe e LBPe em relação à quantidade de defeitos detectados pelas equipes de inspetores, através das variáveis NRO-DEFEITOS-LBCh, NRO-DEFEITOS-LBCE e NRO-DEFEITOS-LBPE. A análise dessa hipótese mostrou que os tipos de defeitos analisados, pertencentes à classificação por Omissão, através da análise estatística, mostrou que a técnica LBPe, foi eficiente na detecção de defeitos dos tipos FO e PO. A técnica LBCh foi eficiente na detecção de defeitos do tipo IO e a técnica LBCe no tipo PO.

A Hipótese 2 foi subsidiada com dados obtidos no Formulário de Aspectos Gerais, traçando o perfil da amostra, associando-se as variáveis NRO-DEFEITOS-INDIVIDUAL e MÉDIA-EXPERIÊNCIA-INSPECTOR. Essa hipótese mostrou que os participantes com mais experiência não executaram melhor a inspeção do que os participantes com menos experiência.

Comparando-se os resultados obtidos com estudos já realizados com as técnicas LBCh, LBCe e LBPe, observou-se que, em termos gerais, os resultados aqui obtidos foram compatíveis com os de estudos anteriores. Notou-se que a técnica LBPe mostrou-se mais eficiente que a LBCe que, por sua vez, mostrou-se mais eficiente que a LBCh.

Essa avaliação e análise de dados obtidos no estudo empírico representam a base de muitos campos científicos, dando um aspecto relevante, na necessidade de replicar, ou seja, dar continuidade a essa pesquisa.

6.2 CONTRIBUIÇÕES DO ESTUDO

O trabalho contribuiu para um entendimento detalhado do processo de inspeção e das técnicas de leitura LBCh, LBCe e LBPe, através do estudo empírico realizado com profissionais da área de informática.

Uma contribuição importante desse estudo foi a realização da pesquisa com profissionais da área, ou seja, que possuem experiência na área de

desenvolvimento de software e na criação de documentos de requisitos de software. Vários estudos anteriores foram realizados com alunos.

Outra contribuição do trabalho foi fornecer subsídios, com o auxílio da análise estatística, para responder a indagação referente ao emprego das técnicas LBCh, LBCe e LBPe no que diz respeito à eficiência das técnicas em detectar defeitos no documento de requisitos de software. Os resultados obtidos são úteis para profissionais que atuam na análise e especificação de requisitos de software, uma vez que podem subsidiar a escolha da técnica de inspeção mais apropriada. Além disso, o estudo contribui para a melhoria contínua da qualidade dos softwares desenvolvidos, pois os profissionais terão informações sobre os tipos de defeitos com maior índice de ocorrência em documentos de requisitos de software.

Os profissionais que participaram do estudo declararam que o método de inspeção e as técnicas de leitura, quando usadas no ambiente de trabalho, contribuem para conduzir a construção do software, garantindo um produto confiável e com qualidade.

Os profissionais observaram também a importância da inspeção em minimizar os custos de produção do software e o esforço gasto no seu desenvolvimento, garantindo um tempo maior para desenvolvimento de outros projetos, além de melhorar a comunicação entre a equipe.

Foi declarado pelos participantes, que a inspeção é um método objetivo e direcionado, podendo ser usado facilmente no ambiente de trabalho, principalmente no desenvolvimento de sistemas, pelo fato de ser uma técnica de fácil entendimento.

6.3 PERSPECTIVAS FUTURAS

No sentido de dar continuidade ao trabalho apresentado, os seguintes aspectos foram levantados como estudos futuros:

- Realizar um estudo empírico, estendendo o uso de outras técnicas de leitura, relacionadas à inspeção, como por exemplo, a Técnica de Leitura Baseada em Defeitos;
- Realizar replicações do estudo, refinando as técnicas usadas e a classificação de defeitos, para confirmar ou negar os resultados originais, podendo assim completar esse estudo, organizando os objetos de estudos e os seus dados, para compor uma base histórica de dados;
- A partir das técnicas estudadas, propor uma nova técnica de leitura e uma classificação de defeitos para direcionar o profissional na criação do documento de requisitos de software.

Finalmente, cabe destacar a importância do trabalho para a área de ensino e pesquisa e também para a prática profissional, enfocando a engenharia de requisitos e a inspeção de software.

REFERÊNCIAS BIBLIOGRÁFICAS

ABIB, J.C. *Abordagem Goal Question Metric (GQM) para Avaliação da Qualidade de Software*. Dissertação de Mestrado. DC/UFSCar, São Carlos, SP, 1998.

ACKERMAN, A.F.; BUCKWALK, L.S.; LEWSKI, F.H. Software Inspections: An Effective Verification Process. *IEEE Software*, n. 1, vol. 14, p. 31-36, 1989.

AMBRUST, O. *Developing a Characterization Scheme for Inspection Experiments*. Project Thesis. Department of Computer Science, University of Kaiserslautern, Alemanha , 2002.

ANSI/IEEE Std 610-1990. IEEE Standard Glossary of Software Engineering Terminology. *IEEE Computer Society Press*, 1990.

ANSI/IEEE Std 830-1998. IEEE Guide to Software Requirements Specifications. *IEEE Computer Society Press*, 1998.

ANSI/IEEE Std 1012-1998. IEEE Standard for Software Verification and Validation Plan. *IEEE Computer Society Press*, 1998.

ANSI/IEEE Std 1028-1997. IEEE Standard for Software Reviews. *IEEE Computer Society Press*, 1997.

BASILI, V.R.; WEISS, D.M. Evaluation of a Software Requirements Document by Analysis of Change Data. *In: Proceedings 5th International Conference On Software Engineering, IEEE CS Press, California, EUA, p. 314-323, 1981.*

BASILI, V.R.; SELBY, R.W. Comparing the Effectiveness of Software Testing Strategies. *IEEE Transactions on Software Engineering*, n. 12, vol. 13, p. 1278-1296, 1987.

BASIL, V.R.; GREEN, S.; LAITENBERGER, O.; LANUBILE, F.; SHULL, F. The Empirical Investigation of Perspective-based Reading. *Empirical Software Engineering*, n. 2, vol. 1, p. 133-164, 1996.

BERLING, T.; THELIN, T. A Case Study of Reading Techniques in a Software Company. In: *International Symposium on Empirical Software Engineering, IEEE Computer*, n. 1, vol. 50, p. 25-35, 2004.

BIFFL, B.; BED, F.; LAITENBERGER, O. Investigating the Cost-Effectiveness of Reinspections in Software Development. In: *International Conference on Software Engineering, IEEE Computer*, Toronto, Canada, p.155-164, 2001.

BOEHM, B.; BASILI, V. Software Defect Reduction Top 10 List. *IEEE Computer*, n.1, vol. 34, p. 135-137, 2001.

BRIAND, L.C.; FREIMUT, B.; VOLLEI, F. Assessing the Cost-Effectiveness of Inspections by Combining Project Data and Expert Opinion. *IEEE Transactions on Software Engineering*, n. 10, vol. 22, p. 124, 2000.

BUSSAB, W.O.; MORETTIN, P.A. Estatística Básica. 4ª edição, São Paulo: Atual, 1987.

CARVER, J. Impact of Background and Experience on Software Inspections. Ph.D. Thesis, University of Maryland, EUA, 2002.

CHERNAK, Y. A Statistical Approach to the Inspection Checklist Formal Synthesis and Improvement. *IEEE Transactions on Software Engineering*, n. 12, vol. 22, p. 866-874, 1996.

CIOLKOWSKI, M.; DIFFERDING, C.; LAITENBERGER, O.; MUNCH, J. Empirical Investigation of Perspective-based Reading: A Replicated Experiment. *Technical Report ISERN-97-13*, University of Kaiserslautern, Alemanha, 1997.

CIOLKOWSKI, M. Evaluating the Effectiveness of Different Inspection Techniques on Informal Requirements Documents. Master Thesis. Department

of Computer Science, University of Kaiserslautern, Alemanha, Junho, 1999.

DOOLAN, E.P. Experience with Fagan's Inspection Method. *Software-Practice and Experience*, n. 2, vol. 22, p. 173-182, 1992.

DÓRIA, E.S. Replicação de Estudos Empíricos em Engenharia de Software. Dissertação de Mestrado. IMC/USP, São Carlos, SP, 2001.

EBENAU, R.G.; STRAUSS, S.H. *Software Inspection Process*. 1ª edição, McGraw Hill, 1994.

FAGAN, M.E. Design and Code Inspections to Reduce Errors in Program Development. *IBM System Journal*, n. 3, vol. 15, p. 219-248, 1976.

FAGAN, M.E. Advances in Software Inspections. *IEEE Transactions on Software Engineering*, n. 7, vol. 12, p. 744-751, 1986.

FUSARO, P.; LANUBILE, F.; VISAGGIO, G. A Replicated Experiment to Assess Requirements Inspections Techniques. *Empirical Software Engineering Journal*, n. 1, vol. 2, p. 39-57, 1997.

JONES, C. Software Defect Removal Efficiency. *IEEE Computer*, n.4, vol. 29, p. 94-95, 1996.

LAITENBERGER, O. Perspective-based Reading: Technique, Validation and Research in Future. *Technical Report ISERN-95-01*, University of Kaiserslautern, Alemanha, 1995.

LAITENBERGER, O.; BAUD, J. Perspective-based Reading of Code Documents. *Journal of Information and Software Technology*, n. 8, vol. 39, p. 781-791, 1997.

LAITENBERGER, O.; EMAM, K.; HARBICH, T. An Internally Replicated Quasi-Experimental Comparison of Checklist and Perspective based Reading of Code Documents. *IEEE Transactions on Software Engineering*, n. 5, vol. 27, p. 387-421, 2001.

LANUBILE, F.; VISAGGIO, G. Assessing Defect Detection Methods for Software Requirements Inspections Through External Replication. *Technical Report ISERN-96-01*, University of Bari, Itália, 1996.

LANUBILE, F.; VISAGGIO, G. Evaluating Defect Detection Techniques for Software Requirements Inspections. *Technical Report ISERN-97-05*, University of Bari, Itália, 1997.

LANUBILE, F.; SHULL, F.; BASILI, V. Experimenting with Error Abstraction in Requirements Documents. *In: Proceedings of the 5th International Symposium on Software Metrics*, Maryland, EUA, 1998.

LAUDON, H.C.; LAUDON, J.P. *Gerenciamento de Sistemas de Informação*. 3ª edição. Rio de Janeiro: LTC Editora, 1999.

LEVINE, D.M.; BERENSON, M.L.; STHEPAN, D. *Estatística: Teoria e Aplicações*. Rio de Janeiro: LTC Editora, 2000.

LOPES, P.S.N.D. *Uma Taxonomia da Pesquisa na Área de Engenharia de Requisitos*. Dissertação de Mestrado. IME/USP, São Paulo, SP, 2002.

MARTIN, J.; SCHENEIDER, G.M.; TSAI, W.T. An Experimental Study of Fault Detection in User Requeriments Documents. *ACM Transactions on Software Engineering and Methodology*, n. 2, vol.1, p. 188-204, 1992.

MAIDANTCHIK, C.; MONTONI, M.; SANTOS, G. Learning Organizational Knowledge: An Evolutionary Proposal for Requirements Engineering. *Communications of the ACM*, n. 7, vol. 33, p. 151-157, 2002.

NUSEIBEH, B.; EASTERBROOK, S. Requirements Engineering: A Roadmap. *Communications of the ACM*, n. 4, vol. 10, p. 35-47, 2002.

O'NEIL, D. Issues in Software Inspection. *IEEE Software*, n. 1, vol. 14, p. 18-19, 1997.

PARNAS, D.L.; WEISS, D.M. Active Design Reviews: Principles and Practice.

In: Proceedings of 8th International Conference on Software Engineering, Maryland, EUA, p. 132-136, 1985.

PCMSO – Documento de Requisitos de Software do Programa de Controle Médico e Saúde Ocupacional. Americana, 20 p., 2004.

PORTER, A.A.; VOTTA, L.G. An Experiment to Assess Different Defect Detection Methods for Software Requirements Inspections. *IEEE Computer Society Press*, n. 3, vol. 10, p. 103-112, 1994.

PORTER, A. A.; VOTTA, L.G.; BASILI, V.R. Comparing Detection Methods for Software Requirements Inspections: A Replicated Experiment. *IEEE Computer Society Press*, n.6, vol. 21, p. 563-575, 1995.

PRESSMAN, R.S. *Engenharia de Software*. São Paulo: Makron Books, 1995.

ROCHA, A.R.C.; MALDONADO, J.C.; WEBER, K.C. *Qualidade de Software – Teoria e Prática*. São Paulo: Prentice Hall, 2001.

SAUER, C.; JEFFERY, D.R.; LAND, L.; YETTON, P. The Effectiveness of Software Development Technical Reviews: A Behaviorally Motivated Program of Research. *IEEE Transactions on Software Engineering*, n. 1, vol. 26, p. 1-14, 2000.

SHULL, F.J. Developing Techniques for Using Software Documents: A Series of Empirical Studies. Ph.D. Thesis. University of Maryland, EUA, 1998.

SORUMGARD, S. An Empirical Study of Process Conformance. *Goddard Space Flight Center SEL-96-002*, p. 115-124, 1996.

TRAVASSOS, G.H.; SHULL, F.J.; CARVER, J. Evolving a Process for Inspecting OO Designs. *In: XIII SBES: Workshop on Software Quality*. Santa Catarina, Brasil, 1999.

TYRAN, C. K.; GEORGE, J.F. Improving Software Inspections with Group Process Support. *Communications of The ACM*, n. 9, vol.45, p. 87-92, 2002.

VALENTI, S.; PANTI, M.; CUCCHIARELLI, A. Overcoming Communication Obstacles in User-Analyst Interaction for Functional Requirements Elicitation. *ACM SIGSOFT Software Engineering Notes*, n. 1, vol. 23, p. 50-55, 1998.

WONG, Y.K. Use of Software Inspection Inputs in Practice. *Communications of the ACM*, n. 8, vol. 35, p. 725-726, 2002.

ZAVE, P. Classification of Research Efforts in Requirements Engineering. *ACM Computing Surveys*, n. 4, vol. 29, p. 315-321, 1997.

APÊNDICE

APÊNDICE A – DOCUMENTOS DE INSPEÇÃO

A.1 DRS – PCMSO

Documento de Requisitos de Software

PROGRAMA DE CONTROLE MÉDICO E SAÚDE OCUPACIONAL - PCMSO

Conteúdo

1. Introdução

- 1.1 Objetivo**
- 1.2 Escopo**
- 1.3 Definições, Siglas e Abreviações**
- 1.4 Visão Geral**

2. Descrição Geral do Sistema

- 2.1 Perspectiva do Produto**
- 2.2 Funções do Produto**
- 2.3 Características do Usuário**
- 2.4 Limitações e Suposições**

3. Requisitos Específicos

3.1 Requisitos Funcionais

- 3.1.1 Requisitos para Acessar o Sistema
- 3.1.2 Requisitos para Cadastrar uma Nova Empresa
- 3.1.3 Requisitos para Cadastrar um Novo Funcionário
- 3.1.4 Requisitos para Cadastrar um Novo Médico
- 3.1.5 Requisitos para Cadastrar Riscos
- 3.1.6 Requisitos para Cadastrar Exame-Risco
- 3.1.7 Requisitos para Cadastrar Doenças - CID
- 3.1.8 Requisitos para Identificar e Cadastrar Riscos da Empresa
- 3.1.9 Requisitos para Identificar e Cadastrar Riscos do funcionário
- 3.1.10 Requisitos para Cadastrar e Atualizar Doenças Pré-Existentes
- 3.1.11 Requisitos para Definir Exames do funcionário
- 3.1.12 Requisitos para Cadastrar e Atualizar Histórico do funcionário
- 3.1.13 Requisitos para Acompanhar a saúde do funcionário
- 3.1.14 Requisitos para Providenciar Medidas Adequadas
- 3.1.15 Requisitos para Controlar Acidentes de Trabalho
- 3.1.16 Requisitos para Gerar o Total Geral

3.2 Requisitos de Interface Externa

- 3.2.1 Interfaces do Usuário
- 3.2.2 Interfaces de Hardware

3.3 Requisitos de Performance

3.4 Outros Requisitos

Documento de Requisitos de Software

1. Introdução

1.1 Objetivo

Esse documento apresenta a especificação de requisitos para um sistema de informação denominado como Programa de Controle Médico e Saúde Ocupacional – PCMSO, utilizado na área de medicina do trabalho por clínicas ocupacionais para auxiliar no controle de exames e riscos dos funcionários nas empresas existentes. Esses requisitos representam o entendimento do sistema, obtidos por meio de entrevistas e de consultas fornecidas pelo responsável do setor da área médica. Os objetivos do documento de requisitos são:

- Demonstrar para o responsável do setor da área médica que o analista de sistemas entendeu corretamente os requisitos do sistema PCMSO;
- Obter um retorno do responsável pela área no que se refere ao entendimento do sistema e, em particular, servir como base para a contratação dos serviços de automação na área de PCMSO;
- E, finalmente, servir como um guia durante o projeto do sistema.

1.2 Escopo

Esse documento contém, principalmente, os requisitos necessários para automatizar as operações dos exames realizados e o cadastro de doenças pré-existentes no quadro de funcionários da empresa e seus respectivos relatórios de avaliações. Assim, os requisitos para os outros aspectos de operação do PCMSO, como por exemplo, interação com laboratórios e fiscais do ministério do trabalho, estão, na maior parte, fora do escopo deste documento. Mas o documento descreve as principais entidades externas que deverão interagir com o sistema de PCMSO.

1.3 Definições, Siglas e Abreviações

- Funcionário. É uma pessoa cadastrada no sistema que irá realizar os exames pré-estabelecidos.
- Empresa. É aquela na qual o funcionário presta serviços.
- Riscos. Refere aos riscos que o funcionário está exposto em suas atividades profissionais e o grau de risco é de 1 – 4.
- Exames. Exames periódicos realizados pelo funcionário de acordo com os riscos e com as doenças pré-existentes.
- CID. Cadastro Internacional de Doenças (Padrão Internacional).
- Médico. O profissional fornece o diagnóstico do funcionário e emite os relatórios de avaliações.
- Doenças pré-existentes. Doenças que o funcionário já possui.
- Técnico de Segurança do Trabalho ou Engenheiro de Segurança. É responsável em avaliar a exposição aos riscos.
- CAT. Comunicação de Acidente de Trabalho.
- LTCAT. Laudo Técnico de Condições Ambientais do Trabalho.
- PPRA. Programa de Prevenção de Riscos Ambientais.
- Cartão de identificação. É um cartão que cada funcionário poderá receber ao cadastrar-se no sistema. Esse cartão possui um código de barras com o código do funcionário, que será lido através de uma leitora de cartão magnético.

1.4 Visão Geral

O restante desse documento é dividido em duas seções. A seção 2 contém uma visão geral e informal do sistema de informação. A seção 3 apresenta os requisitos funcionais e os requisitos de performance do sistema.

2. Descrição Geral do Sistema

2.1 Perspectiva do Produto

O sistema PCMSO irá automatizar as operações do dia-a-dia da clínica. O sistema irá permitir o cadastro dos dados referentes às empresas, aos funcionários, aos riscos e aos exames, e o controle dos exames e do histórico do funcionário a serem realizados de maneira rápida,

precisa e conveniente, atualizando automaticamente as movimentações do funcionário. As melhorias para o futuro são:

- As avaliações de funcionários sobre os lançamentos dos resultados dos exames de audiometria.
- Emissão de documentos legais.
- Estatísticas, tais como calcular e emitir um relatório sobre o total geral discriminando o total de serviços prestados e o total de dívidas pendentes das empresas, mensalmente.
- Avaliar a portabilidade para outras plataformas.
- Entrada de dados via *Web*.

O sistema PCMSO irá operar em um microcomputador PC, que a clínica possui. O sistema poderá interagir com leitora de cartão magnético e fotos digitalizadas, para facilitar as consultas, e com impressoras para fornecer os relatórios para as empresas. Todas as informações a respeito dos funcionários, médicos, exames, empresas, riscos e em relação à clínica serão armazenadas no computador.

2.2 Funções do Produto

Primeiramente o Engenheiro do Trabalho visita a empresa para efetuar a avaliação do ambiente e riscos que o funcionário está exposto. O Engenheiro produz dois documentos referentes aos riscos e ao ambiente que são LTCAT e PPRA. Baseados nesses documentos e nos riscos levantados, o médico define quais são os exames a serem realizados de acordo com a função/funcionário e o período de repetição desses exames. Com esses documentos as empresas associam-se a uma clínica ocupacional e encaminham seus funcionários para a avaliação médica.

Antes de acessar as operações do sistema PCMSO o usuário deverá fornecer o nome do usuário e a senha a serem validados na tela de acesso, permitindo ao usuário o acesso às operações do sistema. Na tela principal do sistema o usuário tem as seguintes opções: Cadastro, Movimentação, Relatórios e Sair. Na opção de cadastro o usuário tem as operações:

- PCMSO. Denominado como um Programa de Controle Médico e Saúde Ocupacional para gerenciar as avaliações e controlar os riscos que o funcionário está exposto.
- Cadastrar a empresa, informando todos os seus dados, cadastrar e identificar os riscos que a empresa oferece ao funcionário de acordo com os documentos LTCAT e PPRA;
- Cadastrar o funcionário informando os seus dados pessoais, cadastrar e identificar os riscos que o funcionário está exposto na empresa de acordo com a sua função/cargo, cadastrar e atualizar as doenças pré-existentes do funcionário de acordo com o CID, definir os exames do funcionário a serem realizados a partir dos riscos, das doenças pré-existentes e das lesões causadas pela função/setor, cadastrar e atualizar o histórico do funcionário a partir das consultas realizadas, dos exames e das avaliações médicas.
- Cadastrar o médico informando seus dados para prestar seus serviços a clínica.
- Cadastrar os riscos existentes informando a descrição e o tipo do risco com padrão de 1-4.
- Cadastrar exame-risco tem como função cadastrar os exames a serem realizados de acordo com o risco existente.
- Cadastrar doenças de acordo com o padrão internacional (CID).

Na opção de movimentação o objetivo principal do sistema, além de gerenciar as avaliações (acompanhar saúde do funcionário) é controlar se as exposições aos riscos estão gerando algum tipo de lesão ou doença no funcionário, providenciando as medidas adequadas e também controlar os acidentes de trabalho dos funcionários com a emissão do CAT pela empresa para a clínica ocupacional. O sistema irá mensalmente calcular e emitir um relatório sobre o total geral discriminando o total de serviços prestados e o total de dívidas pendentes das empresas que possuem.

Na opção de relatórios o sistema possui o relatório de PCMSO enviado no final de cada mês para as empresas com a data de consulta de cada funcionário e com as avaliações e os

exames realizados e o relatório de pendências da movimentação enviado mensalmente para as empresas relatando os funcionários com consultas e exames atrasados.

2.3 Características do Usuário

O sistema será utilizado pelos profissionais da área médica, pelas secretárias, pelas fonoaudiólogas, pelos auxiliares administrativos e indiretamente pelos funcionários.

O médico deve estar familiarizado com as capacidades do sistema, inclusive para fornecer as atualizações das movimentações. Os funcionários não precisam ter conhecimento do sistema somente terão de trazer seus cartões de identificação.

2.4 Limitações e Suposições

- Suposições Iniciais
 - A empresa irá melhorar o hardware do computador assim que for necessário, incluindo a compra de uma leitora de cartão magnético e de um disco rígido, e se necessário e/ou opcionalmente poderá trabalhar com fotos digitalizadas.
- Outras Suposições e Limitações
 - Todos os funcionários, médicos, exames, riscos e empresas devem ser registrados no sistema;
 - Os funcionários devem ter um cartão de identificação da empresa pertencente;
 - Os funcionários são caracterizados pelos seguintes atributos: endereço e telefone;
 - O hardware do computador deve manter a data correta;
 - A empresa deve enviar o relatório do quadro de função/funcionário e os documentos LTCAT e PPRA;
 - A empresa deve informar os acidentes de trabalho com os funcionários enviando o documento CAT;
 - As empresas devem ter um cartão de crédito válido, ou estarem aptos a fazer um depósito, em dinheiro ou cheque mensalmente dos serviços prestados;
 - O pagamento será aceito em dinheiro, cheque ou cartão de crédito;
 - É pedido as empresas que liquidem suas dívidas. O pagamento deve ser feito integralmente, de uma só vez;
 - Periodicamente no final de cada mês será enviado para as empresas o relatório de agendamento de cada funcionário com as avaliações a serem realizadas;
 - A empresa deverá confirmar com a clínica ocupacional o agendamento realizado nas datas pré-determinadas;
 - As empresas são caracterizadas pelos seguintes atributos: endereço, telefone e CNPJ;
 - No contrato entre a empresa e a clínica, cada funcionário terá direito a um número fixo de consultas mensais, mais os retornos necessários de acordo com a avaliação médica, não podendo exceder em 3 consultas/mês;
 - O relatório do total de serviços deve ser enviado às empresas com uma semana de antecedência ao dia de pagamento.

3. Requisitos Específicos

A seguir será apresentada a lista dos requisitos funcionais que o sistema deve satisfazer. Cada requisito é caracterizado através dos seguintes itens:

- Descrição: breve descrição da funcionalidade do sistema.
- Entrada: uma descrição das entradas recebidas pelo software.
- Processamento: uma descrição do que o sistema de software deve realizar com a entrada recebida.
- Saída: uma descrição da resposta/novo estado do sistema de software.

3.1 Requisitos Funcionais

As secretárias não necessitam de nenhum conhecimento específico em relação a computadores, mas precisará familiarizar-se com os processos de cadastro e movimentação.

3.1.1 Requisitos para Acessar o Sistema

Requisito Funcional 1

- **Descrição**
A secretária e o auxiliar administrativo devem gerenciar a parte de cadastro do sistema, como: incluir, pesquisar, excluir e alterar funcionários, empresas, riscos, exames e outros, assim como imprimir os diversos relatórios. O usuário deve entrar com o nome do usuário e uma senha para ter acesso ao sistema.
- **Entrada**
Nome do usuário e senha do usuário.
- **Processamento**
O sistema valida o nome e a senha e possibilita o acesso às funções. Caso o usuário ou a senha digitados não seja reconhecido pelo sistema, este enviará uma mensagem de erro ao usuário.
- **Saída**
Permissão de acesso às funções se o nome e a senha forem válidos ou mensagem de acesso não permitido.

Requisito Funcional 2

- **Descrição**
Após o acesso ao sistema de PCMSO o seguinte menu principal é exibido. Do menu principal o usuário pode optar por uma das seguintes opções:
 1. Cadastro
 2. Movimentação
 3. Relatório
 4. Sair
- **Entrada**
O usuário escolhe a opção desejada.
- **Processamento**
O item selecionado é processado.
- **Saída**
Exibe no monitor o sub menu para o item selecionado.

Requisito Funcional 3

- **Descrição**
No sub menu dos itens cadastro e movimentação o usuário pode selecionar uma das opções: inclusão, exclusão, alteração ou pesquisa dos dados do sistema.
- **Entrada**
O usuário fornece os campos (atributos) necessários para realizar a operação.
- **Processamento**
Executa a operação desejada como: inclusão, exclusão, alteração ou pesquisa dos dados do sistema. Na operação de exclusão será enviada uma mensagem de confirmação para o usuário da operação requerida. Os dados serão salvos no banco de dados de acordo com o formato apropriado, caso contrário será enviada uma mensagem ao usuário. Será exibido ao usuário uma mensagem das restrições sobre as operações de exclusão.
- **Saída**
Mostra os dados atualizados de acordo com a operação requerida.

Requisito Funcional 4

- Descrição
No sub menu no item de relatório o usuário pode imprimir os seguintes relatórios existentes.
- Entrada
O usuário seleciona qual tipo de informação quer ter. Ele pode escolher uma das seguintes opções:
 1. Relatório de PCMSO.
 2. Relatório de Pendências da Movimentação.
- Processamento
Obtenção de todos os dados das informações requeridas e a impressão deles.
- Saída
Dados impressos.

3.1.2 Requisitos para Cadastrar uma Nova Empresa

Requisito Funcional 5

- Descrição
Antes de o funcionário realizar os exames é importante cadastrar todas as informações referentes à empresa onde trabalha, gerando o código da empresa.
- Entrada
A empresa é caracterizada com os seguintes atributos: CNAE fantasia, endereço, bairro, cidade, UF, CEP, telefone, CNPJ, Insc. Estad., tp. Atividade, tipo de risco, nome do funcionário.
- Processamento
Gera o código da empresa e cria um registro da empresa. Será enviado ao usuário uma mensagem dos campos obrigatórios a serem preenchidos. O campo tipo de risco deve ser do tipo numérico e segue um padrão de 1 - 4. Caso o usuário entre com alguma informação incorreta, uma mensagem de erro será enviada a ele pedindo que digite o dado novamente.
- Saída
Um registro da empresa é produzido. O funcionário pode ser cadastrado.

3.1.3 Requisitos para Cadastrar um Novo Funcionário

Requisito Funcional 6

- Descrição
Um novo empregado da empresa é contratado. A secretária entra com todas as informações necessárias, e gera o código do funcionário. Se a empresa optar pelo cartão de identificação este será entregue ao funcionário.
- Entrada
A secretária fornece as seguintes informações: nome, data de nascimento, função/setor, endereço e telefone.
- Processamento
O sistema gera o código do funcionário e cria um registro para o funcionário. O campo data de nascimento deve ser numérico e segue um padrão. Será enviado ao usuário uma mensagem dos campos obrigatórios a serem preenchidos. Caso o usuário entre com alguma informação incorreta, uma mensagem de erro será enviada a ele pedindo que digite o dado novamente.
- Saída
Um registro do funcionário é produzido. O funcionário pode utilizar os serviços para ser avaliado.

3.1.4 Requisitos para Cadastrar um Novo Médico

Requisito Funcional 7

- Descrição

A secretária entra com todas as informações necessárias, e gera o código do médico para prestar seus serviços a clínica.

- Entrada
A secretária fornece as seguintes informações: nome, especialização.
- Processamento
O sistema gera o código do médico e cria um novo registro. Será enviado ao usuário uma mensagem dos campos obrigatórios a serem preenchidos. Caso o usuário entre com alguma informação incorreta, uma mensagem de erro será enviada a ele pedindo que digite o dado novamente.
- Saída
Um registro do médico é produzido. O médico pode prestar os serviços para a clínica.

3.1.5 Requisitos para Cadastrar Riscos

Requisito Funcional 8

- Descrição
A secretária deve cadastrar os riscos existentes.
- Entrada
O risco é caracterizado com os seguintes atributos: descrição, tipo de risco.
- Processamento
Gera o código de riscos e cria um registro de riscos. O tipo de risco deve ser do tipo numérico e segue um padrão de 1 - 4. Será enviado ao usuário uma mensagem dos campos obrigatórios a serem preenchidos. Caso o usuário entre com alguma informação incorreta, uma mensagem de erro será enviada a ele pedindo que digite o dado novamente.
- Saída
Um registro de riscos é produzido.

3.1.6 Requisitos para Cadastrar Exame-Risco

Requisito Funcional 9

- Descrição
A secretária deve cadastrar os tipos de exames a serem realizados de acordo com o risco existente.
- Entrada
O exame-risco é caracterizado com os seguintes atributos: descrição do exame, código do risco, descrição do risco, código função/setor e descrição função/setor. Será enviado ao usuário uma mensagem dos campos obrigatórios a serem preenchidos. Caso o usuário entre com alguma informação incorreta, uma mensagem de erro será enviada a ele pedindo que digite o dado novamente.
- Processamento
Gera o código do exame-risco e cria um novo registro.
- Saída
Um registro de exame-risco é produzido.

3.1.7 Requisitos para Cadastrar Doenças (CID)

Requisito Funcional 10

- Descrição
A secretária deve cadastrar as doenças existentes de acordo com o CID.
- Entrada
A doença é caracterizada com os seguintes atributos: descrição.
- Processamento
Gera o código da doença e cria um registro de doenças. Será enviado ao usuário uma mensagem dos campos obrigatórios a serem preenchidos. Caso o usuário entre com alguma informação incorreta, uma mensagem de erro será enviada a ele pedindo que digite o dado novamente.

- Saída
Um registro de doenças é produzido.

3.1.8 Requisitos para Identificar e Cadastrar Riscos da Empresa

Requisito Funcional 11

- Descrição
Fornecer o código da empresa e o código da função/setor existente na empresa para gerar o relatório de riscos com as respectivas atividades exercidas pelos funcionários.
- Entrada
O usuário fornece as informações: código da empresa e o código da função/ setor.
- Processamento
Associar os riscos de acordo com a função/setor existentes da empresa. Será enviado ao usuário uma mensagem dos campos obrigatórios a serem preenchidos. Caso o usuário entre com alguma informação incorreta, uma mensagem de erro será enviada a ele pedindo que digite o dado novamente.
- Saída
Relatório de riscos da empresa.

3.1.9 Requisitos para Identificar e Cadastrar Riscos do funcionário

Requisito Funcional 12

- Descrição
Identificar as atividades realizadas pelo funcionário de acordo com a função, levantando os riscos com as respectivas atividades exercidas pelo funcionário, fornecendo as informações: código do funcionário, código da empresa e o código da função/ setor.
- Entrada
O usuário fornece as informações: código do funcionário, nome da empresa e o código da função/ setor.
- Processamento
Associar a função/setor com os riscos que este oferece ao funcionário. Será enviado ao usuário uma mensagem dos campos obrigatórios a serem preenchidos. Caso o usuário entre com alguma informação incorreta, uma mensagem de erro será enviada a ele pedindo que digite o dado novamente.
- Saída
Exibir riscos do funcionário.

3.1.10 Requisitos para Cadastrar e Atualizar doenças pré-existentes

Requisito Funcional 13

- Descrição
Catalogar as doenças que o funcionário já possui. (doenças pré-existentes) de acordo com o CID, com as seguintes informações: código do funcionário, código da função/ setor, data do exame e código da doença.
- Entrada
O usuário fornece as informações: código do funcionário, código da função/ setor, data do exame e código da doença.
- Processamento
Associar o funcionário com as doenças pré-existentes determinadas pelo CID. O campo data do exame deve ser numérico e segue um padrão. Será enviado ao usuário uma mensagem dos campos obrigatórios a serem preenchidos. Caso o usuário entre com alguma informação incorreta, uma mensagem de erro será enviada a ele pedindo que digite o dado novamente.
- Saída
Exibir relação de doenças pré-existentes.

3.1.11 Requisitos para Definir Exames do funcionário

Requisito Funcional 14

- Descrição
Os exames necessários a serem realizados pelo funcionário e a data de realização serão identificados a partir dos riscos e das doenças pré-existentes causadas pela função/setor.
- Entrada
O usuário fornece as informações: código do funcionário.
- Processamento
Identificar os exames necessários a serem realizados pelo funcionário e período para repetição dos mesmos de acordo com os riscos e com as doenças pré-existentes causadas pela função/setor. Será enviado ao usuário uma mensagem dos campos obrigatórios a serem preenchidos. Caso o usuário entre com alguma informação incorreta, uma mensagem de erro será enviada a ele pedindo que digite o dado novamente. O campo data de realização deve ser numérico e segue um padrão.
- Saída
Lista de exames a serem realizados e data de realização.

3.1.12 Requisitos para Cadastrar e Atualizar Histórico do funcionário

Requisito Funcional 15

- Descrição
O funcionário tem seu cartão de identificação. O código do funcionário é lido através da leitora de cartão magnético, para que a secretária recupere o registro do histórico do funcionário.
- Entrada
O número de identificação do funcionário é obtido através da leitora de cartão magnético.
- Processamento
Procura o registro do histórico do funcionário. Será enviado ao usuário uma mensagem dos campos obrigatórios a serem preenchidos. Caso o usuário entre com alguma informação incorreta, uma mensagem de erro será enviada a ele pedindo que digite o dado novamente.
- Saída
Mostra o registro do histórico do funcionário.

Requisito Funcional 16

- Descrição
O funcionário não possui cartão de identificação. O nome do funcionário é digitado, através do teclado, para a recuperação do registro do histórico do funcionário.
- Entrada
O código do funcionário é digitado através do teclado.
- Processamento
Procura o registro do histórico do funcionário. Será enviado ao usuário uma mensagem dos campos obrigatórios a serem preenchidos. Caso o usuário entre com alguma informação incorreta, uma mensagem de erro será enviada a ele pedindo que digite o dado novamente.
- Saída
Mostra o registro do histórico do funcionário.

Requisito Funcional 17

- Descrição
Recuperar a data da última consulta, os exames realizados e a avaliação do funcionário para encaminhar ao médico.
- Entrada

- Data da última consulta.
- **Processamento**
O registro da movimentação da última consulta é recuperado. Será enviado ao usuário uma mensagem dos campos obrigatórios a serem preenchidos. Caso o usuário entre com alguma informação incorreta, uma mensagem de erro será enviada a ele pedindo que digite o dado novamente. O campo data da última consulta deve ser numérico e segue um padrão.
- **Saída**
Mostra o registro da movimentação da última consulta, com os exames realizados e a avaliação médica para encaminhá-lo ao médico.

Requisito Funcional 18

- **Descrição**
O número máximo de consultas a serem realizadas mensalmente por funcionário é de três (3) consultas de acordo com o que foi definido no contrato.
- **Entrada**
Código do funcionário e data da consulta.
- **Processamento**
O sistema verifica o número de consultas realizadas pelo funcionário num determinado mês. Se o número de consultas realizadas no mês for cinco (5), a consulta será rejeitada e será enviada uma mensagem ao usuário para não ultrapassar o limite definido no contrato. Será enviado ao usuário uma mensagem dos campos obrigatórios a serem preenchidos. Caso o usuário entre com alguma informação incorreta, uma mensagem de erro será enviada a ele pedindo que digite o dado novamente. O campo data da última consulta deve ser numérico e segue um padrão.
- **Saída**
Uma mensagem de erro é exibida.

3.1.13 Requisitos para Acompanhar a saúde do funcionário

Requisito Funcional 19

- **Descrição**
A partir dos exames realizados avaliar a saúde do funcionário, informando: código do funcionário, código do exame realizado e resultado do exame.
- **Entrada**
O usuário fornece as informações: código do funcionário, código do exame realizado e resultado do exame.
- **Processamento**
Identificar se os resultados dos exames estão alterados e se esta alteração está progredindo. Será enviado ao usuário uma mensagem dos campos obrigatórios a serem preenchidos. Caso o usuário entre com alguma informação incorreta, uma mensagem de erro será enviada a ele pedindo que digite o dado novamente.
- **Saída**
Situação da saúde do funcionário.

3.1.14 Requisitos para Providenciar Medidas Adequadas

Requisito Funcional 20

- **Descrição**
Avaliar a situação do funcionário e determinar as providências a serem tomadas, fornecendo: código do funcionário e a situação da saúde.
- **Entrada**
O usuário fornece as informações: código do funcionário e a situação da saúde.
- **Processamento**
Passar na perícia médica para determinar as providências.(Ex: mudança de cargo, licença de saúde, etc.). Será enviado ao usuário uma mensagem dos campos obrigatórios a serem preenchidos. Caso o usuário entre com alguma informação

incorreta, uma mensagem de erro será enviada a ele pedindo que digite o dado novamente.

- Saída
Providências a serem tomadas e encaminhamento para a engenharia de segurança da empresa para que está tome as ações preventivas.

3.1.15 Requisitos para Controlar Acidentes de Trabalho

Requisito Funcional 21

- Descrição
Realizar um levantamento dos acidentes de trabalho ocorridos com o funcionário nas empresas que prestou serviços com as informações: código do funcionário, número da CAT emitida, data do acidente e código da empresa.
- Entrada
O usuário fornece as informações: código do funcionário, número da CAT emitida, data do acidente e código da empresa.
- Processamento
Identificar os acidentes de trabalho do funcionário e o motivo dos mesmos. Será enviado ao usuário uma mensagem dos campos obrigatórios a serem preenchidos. Caso o usuário entre com alguma informação incorreta, uma mensagem de erro será enviada a ele pedindo que digite o dado novamente. O campo data do acidente deve ser numérico e segue um padrão.
- Saída
Lista de acidentes de trabalho.

3.1.16 Requisitos para Gerar o Total Geral

Requisito Funcional 22

- Descrição
No final do mês o usuário digita o código da empresa e o mês para o fechamento e o sistema recupera as datas das últimas consultas e calcula o total de serviços prestados no mês do fechamento e as dívidas passadas das empresas para gerar o total geral, emitindo um relatório do total geral e dos serviços prestados para as empresas.
- Entrada
A tecla "enter" é pressionada no teclado depois que o usuário digitou o código da empresa e o mês para o fechamento.
- Processamento
O sistema calcula o total de serviços prestados no mês do fechamento. O sistema verifica a existência de dívidas passadas e calcula o total geral que é a soma das dívidas passadas e os serviços prestados no mês do fechamento. Será enviado ao usuário uma mensagem dos campos obrigatórios a serem preenchidos. Caso o usuário entre com alguma informação incorreta, uma mensagem de erro será enviada a ele pedindo que digite o dado novamente. O campo mês para o fechamento deve ser numérico e segue um padrão.
- Saída
Relatório do total geral, dos serviços prestados e das dívidas passadas.

Requisito Funcional 23

- Descrição
A secretária recebe o dinheiro da empresa, ou seja, o valor referente ao total geral e dá entrada da quantia recebida no sistema e a data do pagamento. O sistema registra o valor pago.
- Entrada
Data do pagamento e a quantia de dinheiro são fornecidas ao sistema pela secretária, através do teclado.
- Processamento

O sistema registra o valor pago. Será enviado ao usuário uma mensagem dos campos obrigatórios a serem preenchidos. Caso o usuário entre com alguma informação incorreta, uma mensagem de erro será enviada a ele pedindo que digite o dado novamente. O campo data do pagamento deve ser numérico e segue um padrão.

- Saída
Mostra o valor pago.

Requisito Funcional 24

- Descrição
Quando a secretária seleciona a opção “fechar-movimentação” (definida no sistema), o arquivo de serviços prestados para as empresas é atualizado e a movimentação de serviços é finalizada e atualizada.
- Entrada
A secretária pressiona a tecla “fechar-movimentação”.
- Processamento
Atualização do arquivo de serviços prestados. Fecha e atualiza a movimentação dos serviços.
- Saída
O arquivo de serviços prestados é atualizado. A movimentação de serviços é fechada e atualizada.

Requisito Funcional 25

- Descrição
Depois que a movimentação de serviços é concluída pelo sistema, a transação da movimentação de serviços é armazenada e impressa.
- Entrada
Sistema conclui a movimentação de serviços corrente.
- Processamento
Armazena a transação da movimentação de serviços e imprime um formulário que o responsável da empresa deve assinar. Retorna ao estado inicial.
- Saída
Formulário impresso. O menu inicial é mostrado. A assinatura do responsável é conferida.

Requisito Funcional 26

- Descrição
Se existir dívida pendente, está poderá ser paga neste momento; a secretária seleciona a opção “fechar-movimentação”, que deverá atualizar o arquivo de serviços prestados e das dívidas passadas para gerar o total geral. Fecha e atualiza a movimentação dos serviços.
- Entrada
A secretária pressiona a tecla “fechar-movimentação”.
- Processamento
Atualização do arquivo de serviços prestados e das dívidas passadas para gerar o total geral. Fecha e atualiza a movimentação dos serviços. Vai para o estado inicial.
- Saída
O arquivo de serviços prestados e das dívidas passadas é atualizado. A movimentação de serviços prestados é fechada e atualizada.

3.2 Requisitos de Interface Externa

3.2.1 Interfaces do Usuário

As telas do sistema de PCMSO consistem em interfaces visuais e amigáveis com o usuário. As figuras abaixo representam alguns dos protótipos das telas do sistema. A Figura 3.1 representa a tela de acesso ao sistema, e o usuário deverá informar seu nome (usuário) e a sua senha (login). Acessado o sistema o usuário terá permissão para realizar seu trabalho em todas as operações como: cadastrar novo funcionário, cadastrar nova empresa, atualizar e consultar as movimentações. A Figura 3.2 representa a tela de menu principal, oferecendo todas as opções do sistema a serem selecionadas pelo usuário. A Figura 3.3 apresenta a tela de cadastro de empresas, informando todos os dados da empresa. A Figura 3.4 apresenta a tela de cadastro de funcionários, informando todos os dados pessoais do funcionário. A Figura 3.5 apresenta a tela de cadastro de doenças pré-existentes dos funcionários. A Figura 3.6 apresenta a tela de cadastro de riscos da empresa e do funcionário. A Figura 3.7 apresenta a tela de cadastro de exame/risco, ou seja, os exames a serem realizados de acordo com os riscos. A Figura 3.8 apresenta a tela de movimentação. A Figura 3.9 apresenta o relatório de PCMSO. A Figura 3.10 apresenta o relatório de pendências/movimentações.

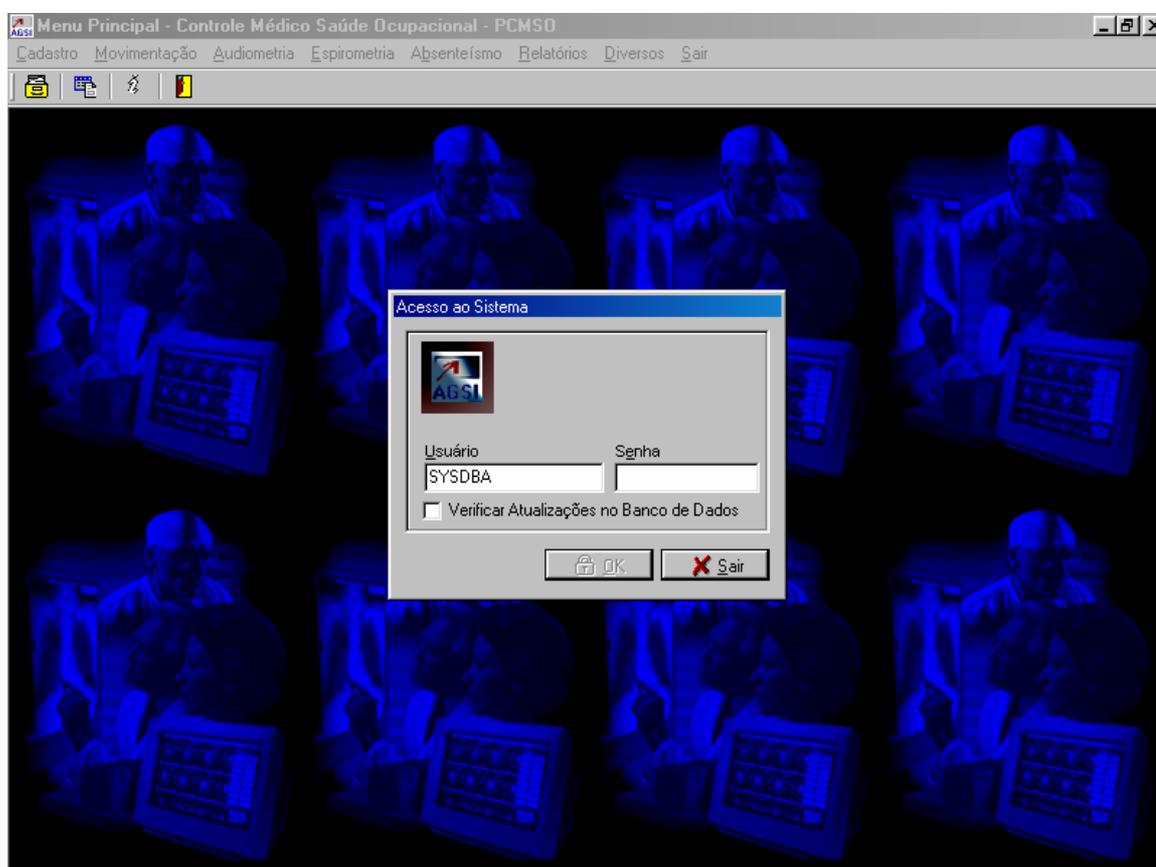


Figura 3.1 Acesso ao sistema

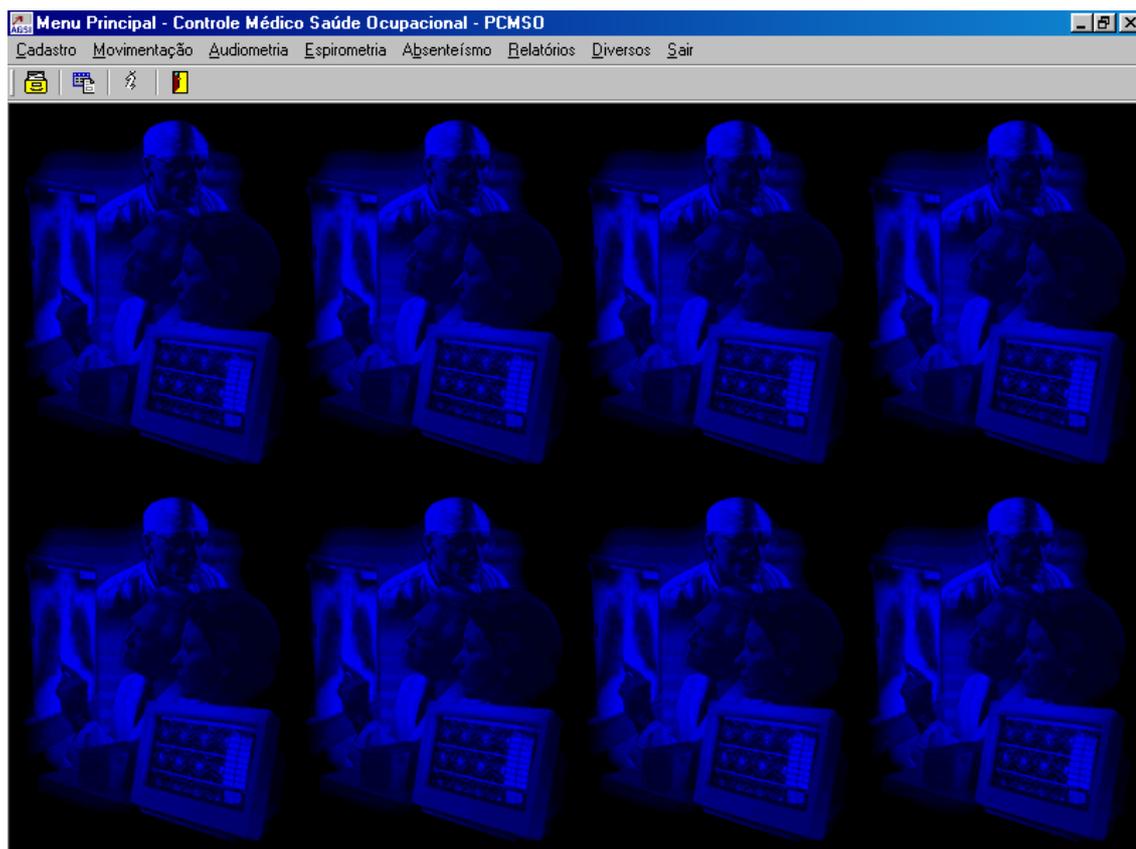


Figura 3.2 Menu principal

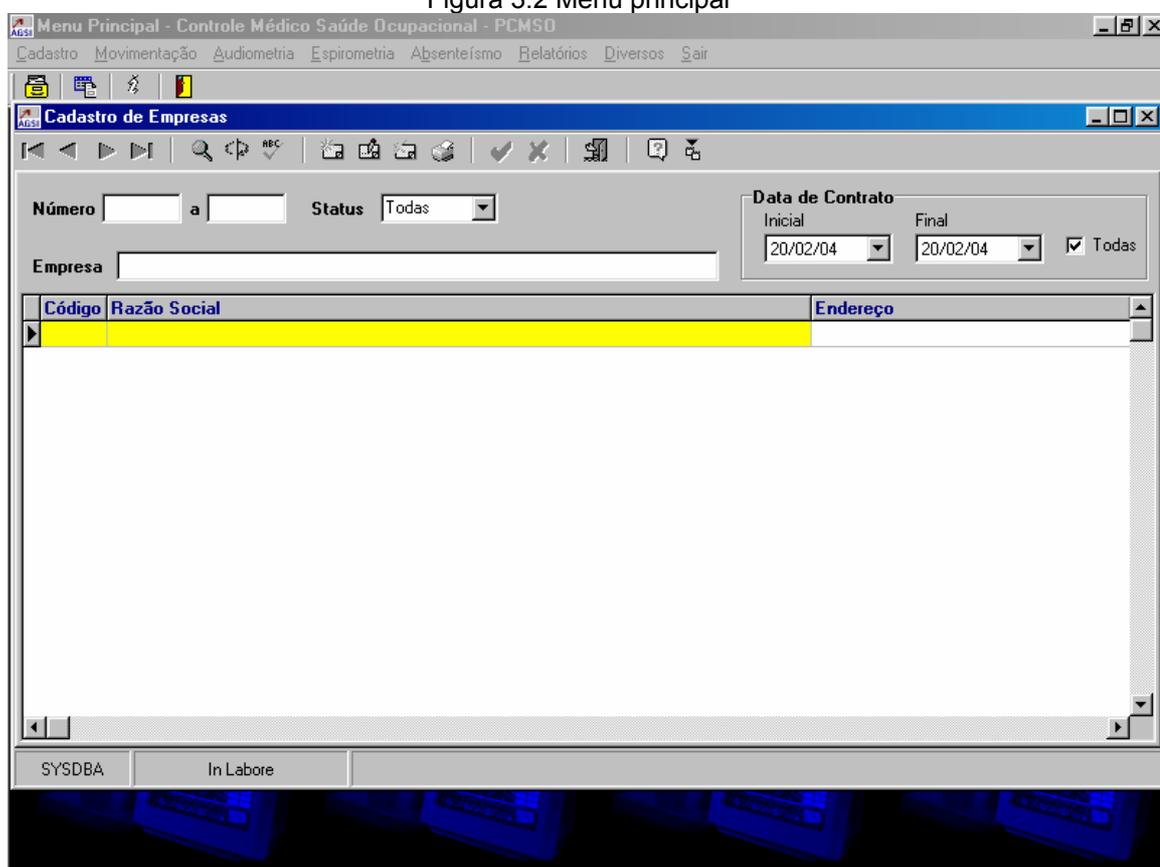


Figura 3.3 Cadastro de empresas

Menu Principal - Controle Médico Saúde Ocupacional - PCMSO - [Cadastro de Funcionários]

Cadastro Movimentação Audiometria Espirometria Absenteísmo Relatórios Diversos Sair

Número a Funcionário Status Todos

Número a Empresa Status Todos

Data Admissão 01/01/20 a 01/01/20 Função Setor

Cód. Emp.	Empresa	Cód. Func.	Funcionário	Setor

SYSDBA In Labore

Figura 3.4 Cadastro de funcionários

Menu Principal - Controle Médico Saúde Ocupacional - PCMSO

Cadastro Movimentação Audiometria Espirometria Absenteísmo Relatórios Diversos Sair

Cadastro de Doenças Pré-Existentes nos funcionários

Número a Usuário

Cód. Usuario	Usuário	Cód. Cid	Descrição do CID	Data da

SYSDBA In Labore

Figura 3.5 Cadastro de doenças pré-existentes

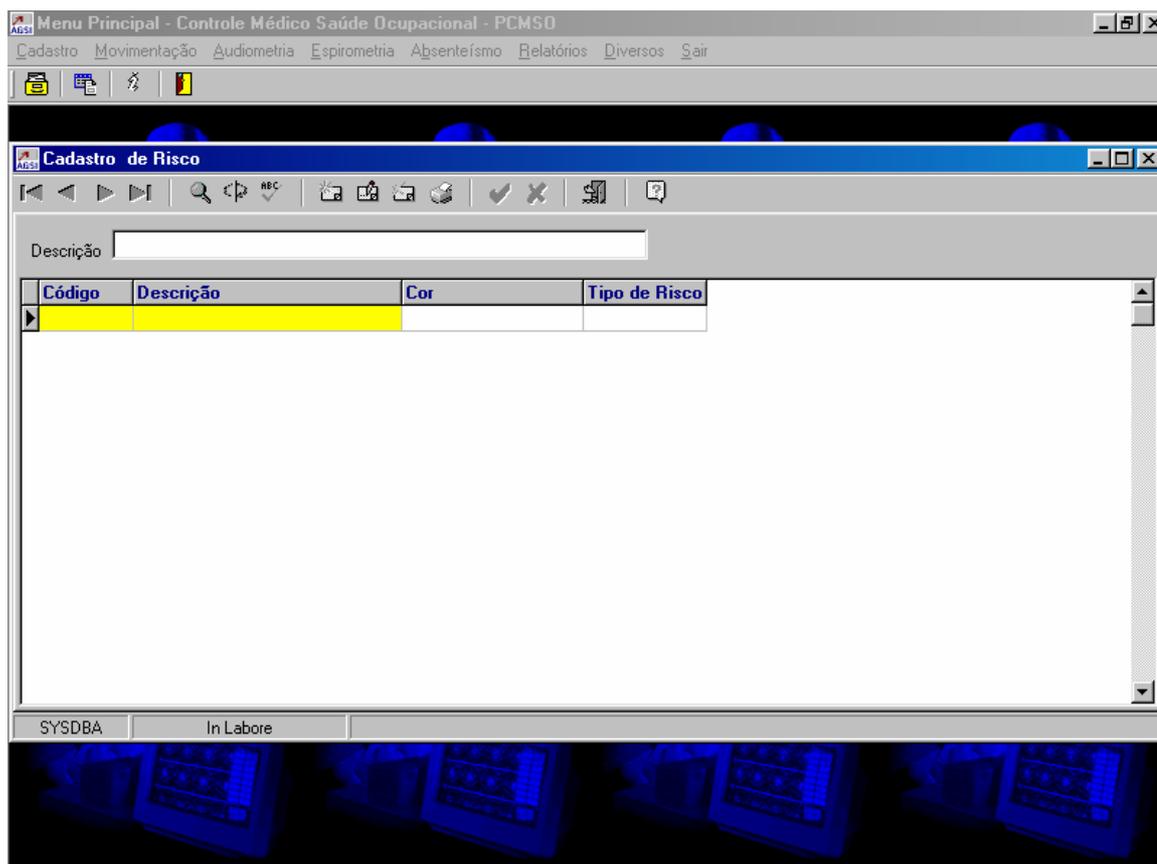


Figura 3.6 Cadastro de riscos

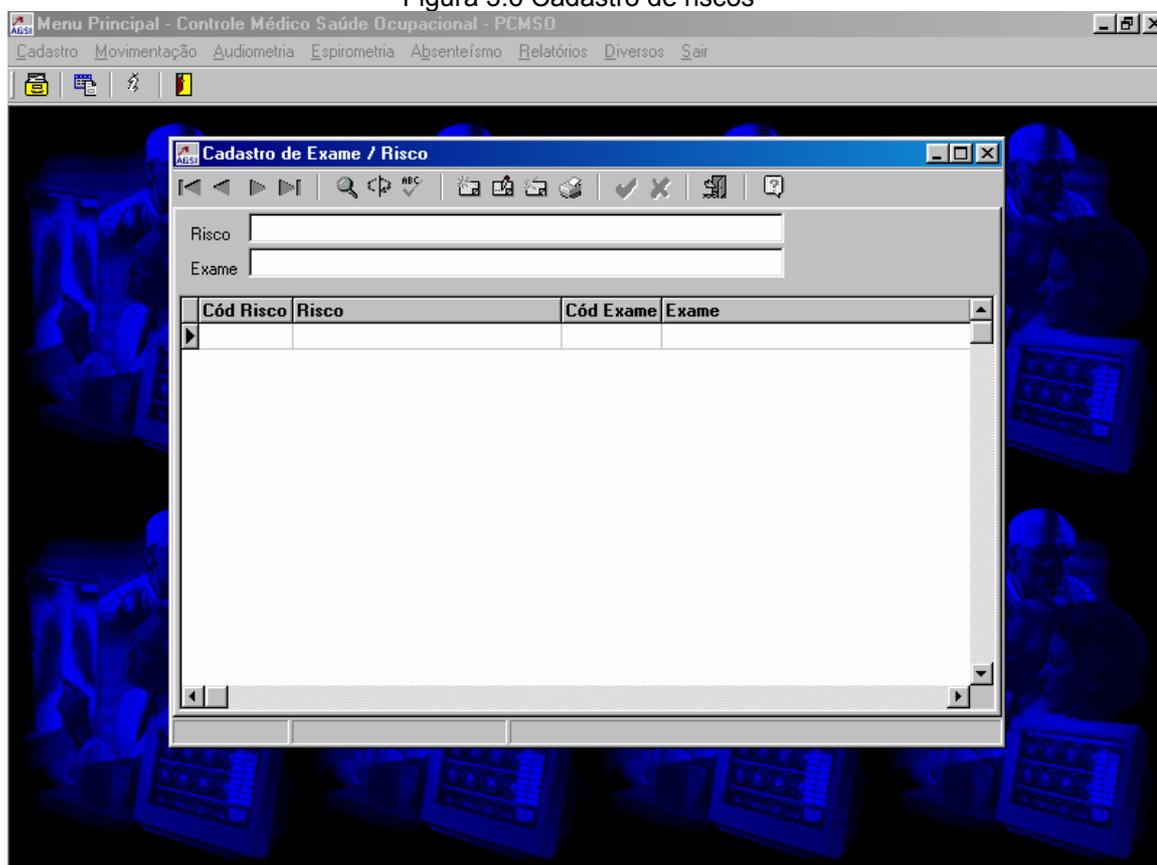


Figura 3.7 Cadastro de exame/risco

Menu Principal - Controle Médico Saúde Ocupacional - PCMSO

Cadastro Movimentação Audiometria Espirometria Absenteísmo Relatórios Diversos Sair

Movimentação

Número a Usuário

Número a Empresa

Tipo Exame Exame

Data de Atendimento 01/01/20 a 01/01/20 Avaliação

Status Usuário
 Todos 1 - Inativos
 0 - Ativos 2 - Afastados

Código	Empresa	Cód. Func.	Funcionário	Dt.Ate

SYSDBA In Labore

Figura 3.8 Movimentação

Menu Principal - Controle Médico Saúde Ocupacional - PCMSO

Cadastro Movimentação Audiometria Espirometria Absenteísmo Relatórios Diversos Sair

Relatório de PCMSO

Empresas
 Inicial
 Final Todas

Funções
 Inicial
 Final Todas

Imprimir Sair

Figura 3.9 Relatório de PCMSO

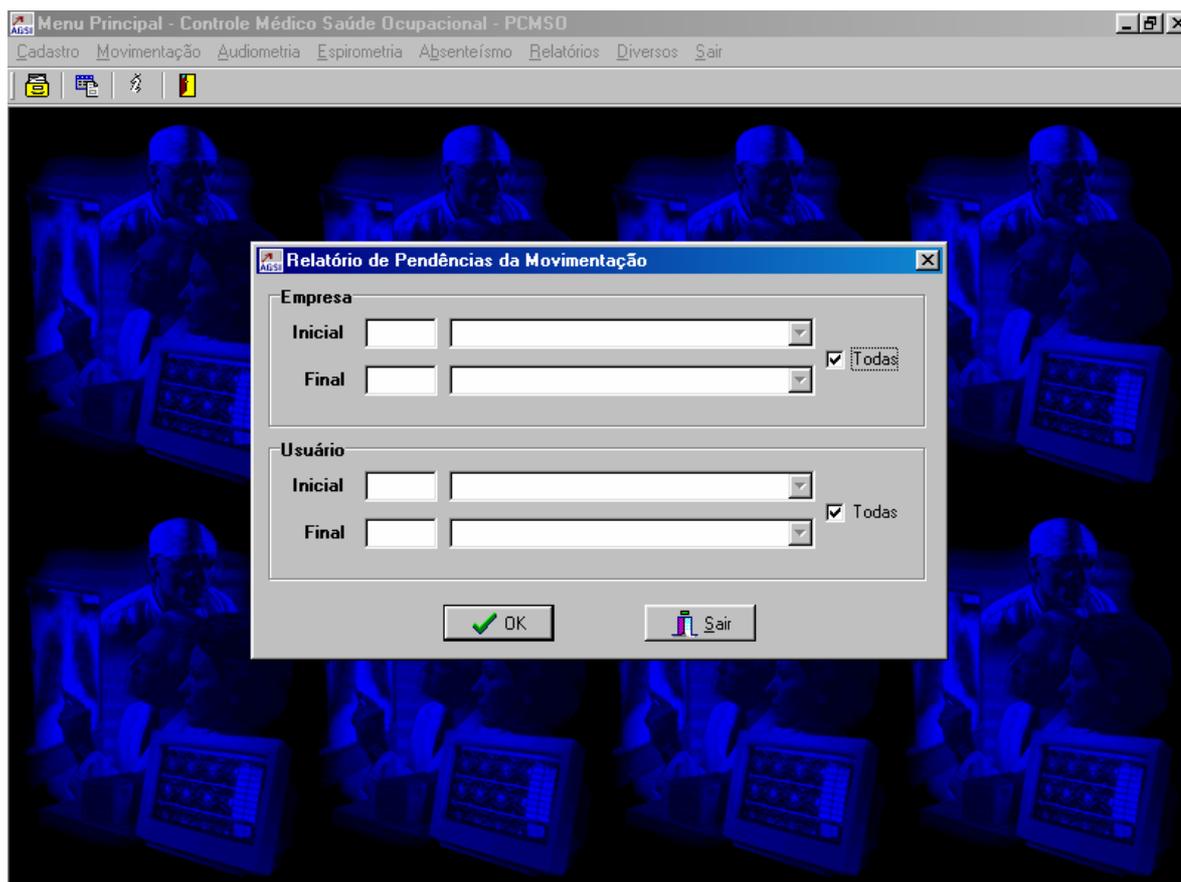


Figura 3.10 Relatório de Pendências da Movimentação

3.2.2 Interfaces de Hardware e de Software

O sistema PCMSO será implementado para funcionar em ambiente PC e sistema operacional *Windows*. A implementação do código fonte será realizada utilizando-se a linguagem de programação visual *Delphi*, versão 8.0. A escolha desta linguagem deve-se ao fato de seu alto grau de desempenho e sua facilidade para a construção de interfaces amigáveis no padrão *Windows*, tendo como principal objetivo facilitar a utilização do sistema pelo usuário, sabendo-se que este padrão é conhecido internacionalmente.

O banco de dados usado para a construção da base de dados do sistema será o *Paradox*, versão 8.0, com a principal finalidade de armazenar todos os dados relativos aos funcionários, as empresas, as movimentações, aos riscos e aos exames.

Os relatórios do sistema serão elaborados com o auxílio do software *Report Smith*, um componente da linguagem *Delphi* 8.0 para construção, visualização e impressão de relatórios, permitindo a construção de relatórios mais elaborados.

Para o usuário o hardware requerido é *Pentium* III, 300 MHz ou *Athlon*, 1.0 Ghz, ambos com 256 MB RAM e com 80 MB de espaço em disco.

3.3 Requisitos de Performance

Requisito de Performance 1

O sistema de software completo será de fácil utilização.

Requisito de Performance 2

Interfaces visuais e amigáveis com o usuário, possibilitando a comunicação com pacote *Office* (*Word/Excel*).

Requisito de Performance 3

O tempo de resposta para as operações de inserção, alteração e exclusão não deve exceder a três segundos.

Requisito de Performance 4

O sistema terá um tempo de resposta rápido. Tipicamente o tempo de resposta para passar o cartão de identificação do funcionário para ser avaliado deverá ser menos de quinze (5) segundos. Na maioria dos casos o tempo de resposta será inferior a um (1) minuto, no caso de pesquisas com a solicitação de datas o tempo de resposta pode alcançar um (1) minuto e com a possível exceção da recuperação dos dados arquivados de um funcionário ou um disco de *backup* o tempo de resposta pode alcançar um (10) minutos.

3.4 Outros Requisitos**Requisito 1**

O sistema poderá funcionar em todos os sistemas computacionais, ou seja, qualquer plataforma empregada na empresa.

Requisito 2

O sistema poderá sofrer mudanças, ou seja, novas funcionalidades e novos módulos podem ser inseridos facilmente.

Requisito 3

O sistema deve ter capacidade para recuperar os dados perdidos da última operação que realizou em caso de falha.

Requisito 4

O sistema deve fornecer facilidades para a realização de backups dos arquivos do sistema.

A.2 INSTRUÇÕES DO ESTUDO

INSTRUÇÕES PARA EXECUÇÃO DO ESTUDO EMPÍRICO

Passo 01 - Estudo: Objetivo do estudo

Você está participando de um estudo empírico baseado em experimentos, apoiado nas etapas do processo de inspeção para avaliar a eficiência das técnicas de leitura: Perspectiva em Projeto, Cenários e *Checklist*.

Passo 02 – Material: Formulários usados no estudo.

Você recebeu os seguintes documentos e formulários:

Instruções para Execução do Estudo Empírico

Documento de Requisitos de Software – PCMSO – Programa de Controle Médico e Saúde Ocupacional

Formulário de Inspeção de Documentos de Software

Formulário de Detecção de Defeitos da Técnica *Checklist* – Grupo 1

Formulário de Detecção de Defeitos da Técnica Cenários – Grupo 2

Formulário de Detecção de Defeitos da Técnica de Perspectiva – Grupo 3

Formulário – Aspectos Gerais do Estudo

Classificação de Defeitos

Passo 03 – Execução: Instruções para realizar o estudo.

1° Verificar a técnica de leitura a ser adotada e o seu respectivo formulário;

2° Ler o Documento de Requisitos de Software - PCMSO;

3° Procurar pelos tipos de defeitos de acordo com o proposto na Classificação de Defeitos individualmente;

4° Anotar os defeitos encontrados no Formulário de Inspeção de Documentos de Software, informando a página, a seção, uma descrição do defeito, o número do requisito e o tipo de defeito.

5° Reunir a equipe, discutir e analisar os defeitos encontrados e anotá-los no Formulário de Inspeção de Documentos de Software, informando a página, a seção, uma descrição do defeito, o número do requisito e o tipo de defeito;

6° Responder o Formulário – Aspectos Gerais do Estudo.

A.3 ANOTAÇÃO DE DEFEITOS

Formulário de Inspeção de Documentos de Software

GRUPO: __ TÉCNICA: _____

Documento Inspecionado: Programa de Controle Médico e Saúde Ocupacional - PCMSO

Horário da Inspeção: Início: _____ Término: _____

Nome do Participante: _____

DEFEITOS ENCONTRADOS					
N°	TIPO DEFEITO	PÁG.	SEÇÃO	N° REQUISITO	DESCRIÇÃO DO DEFEITO

TIPOS DE DEFEITOS	
FO - Funcionalidade Omitida	IA - Informação Ambígua
PO - Performance Omitida	II - Informação Inconsistente
AO - Ambiente Omitido	FI - Funcionalidade Incorreta
IO - Interface Omitida	SE - Seção Errada
O - Outros	

A.4 QUESTIONÁRIO

FORMULÁRIO - ASPECTOS GERAIS DO ESTUDO

Nome:	
Telefone para Contato: () -	
E-mail:	
Profissão:	
Nível de Formação:	Quanto tempo atua na área de SI e outras:
<input type="checkbox"/> Superior <input type="checkbox"/> Mestrando <input type="checkbox"/> Mestre <input type="checkbox"/> Doutorando <input type="checkbox"/> Doutor	<input type="checkbox"/> até 12 meses <input type="checkbox"/> 13 a 24 meses <input type="checkbox"/> 25 a 36 meses <input type="checkbox"/> 37 a 60 meses <input type="checkbox"/> acima de 60 meses
Experiência sobre Inspeção de Software:	Conhecimento sobre as Técnicas de Inspeção:
<input type="checkbox"/> Nenhuma Experiência <input type="checkbox"/> Experiência em Sala de Aula <input type="checkbox"/> Experiência em Empresa <input type="checkbox"/> Experiência em Sala de Aula e Empresa	<input type="checkbox"/> Nenhum Conhecimento <input type="checkbox"/> Conhecimento em Sala de Aula <input type="checkbox"/> Conhecimento em Empresa <input type="checkbox"/> Conhecimento em Sala de Aula e Empresa

A partir das perguntas abaixo, aponte os pontos positivos e as dificuldades da utilização da Técnica de Inspeção, de acordo com a escala de 1 (um) a 5 (cinco), onde 1 (um) representa a menor satisfação e 5 (cinco) a maior satisfação. Comente as perguntas, buscando explicitar os pontos positivos e as dificuldades encontradas. Após procure identificá-las através de sua numeração e elaborar um comentário na área reservada, denominada Comentários Gerais, esta área poderá ser utilizada também para qualquer outro tipo de comentário desejado:

- [1]** Seu conhecimento sobre inspeção foi suficiente para a utilização da Técnica de Leitura?
 1- Insuficiente () 2- Fraco () 3- Regular () 4- Bom ()
 5- Excelente ()
 Comentário: _____
- [2]** Seu conhecimento sobre Documento de Requisitos de Software foi suficiente para a realização da inspeção?
 1- Insuficiente () 2- Fraco () 3- Regular () 4- Bom () 5- Excelente ()
 Comentário: _____
- [3]** A classificação de defeitos utilizada nos formulários de defeitos é de fácil entendimento?
 1- Insuficiente () 2- Fraco () 3- Regular () 4- Bom () 5- Excelente ()
 Comentário: _____
- [4]** A classificação de defeitos utilizada nos formulários de defeitos é completa?

1- Insuficiente () 2- Fraco () 3- Regular () 4- Bom () 5- Excelente ()

Comentário: _____

[5] O tipo de defeitos utilizado nos formulários de defeitos é de fácil entendimento?

1- Insuficiente () 2- Fraco () 3- Regular () 4- Bom () 5- Excelente ()

Comentário: _____

[6] O tipo de defeitos utilizado nos formulários de defeitos é completo?

1- Insuficiente () 2- Fraco () 3- Regular () 4- Bom () 5- Excelente ()

Comentário: _____

[7] O formulário de inspeção de documentos de software utilizado para preenchimento de defeitos é de fácil entendimento?

1- Insuficiente () 2- Fraco () 3- Regular () 4- Bom () 5- Excelente ()

Comentário: _____

[8] O formulário de detecção de defeitos da técnica (Cenários, *Checklist* ou Perspectiva) utilizado para guiar na localização dos defeitos é de fácil entendimento?

1- Insuficiente () 2- Fraco () 3- Regular () 4- Bom () 5- Excelente ()

Comentário: _____

[9] As questões referentes aos formulários das técnicas (Cenários, *Checklist* ou Perspectiva) são suficientes para encontrar os defeitos?

1- Insuficiente () 2- Fraco () 3- Regular () 4- Bom () 5- Excelente ()

Comentário: _____

[10] Os artefatos (formulários e documentos) utilizados na inspeção são suficientes?

1- Insuficiente () 2- Fraco () 3- Regular () 4- Bom () 5- Excelente ()

Comentário: _____

[11] De modo geral, a Técnica de Inspeção é de fácil entendimento?

1- Insuficiente () 2- Fraco () 3- Regular () 4- Bom () 5- Excelente ()

Comentário: _____

[12] A partir do estudo realizado com o documento de requisitos de software, você considera ter adquirido conhecimento sobre a técnica de inspeção?

1- Insuficiente () 2- Fraco () 3- Regular () 4- Bom () 5- Excelente ()

Comentário: _____

[13] Você utilizaria a Técnica de Inspeção em um projeto de desenvolvimento de software no seu ambiente de trabalho?

1- Insuficiente () 2- Fraco () 3- Regular () 4- Bom () 5- Excelente ()

Comentário: _____

Comentários Adicionais:

A.5 RELATÓRIO DE PARTICIPANTES

RELATÓRIO DE PARTICIPANTES		
GRUPO: _____ TÉCNICA: _____		
Nº	Nome do Participante	Função
1		Moderador
2		Relator
3		Inspetor
4		Inspetor
5		Inspetor

APÊNDICE B – TÉCNICAS DE INSPEÇÃO

B.1 TÉCNICA DE INSPEÇÃO: LBCH

Formulário de Detecção de Defeitos da Técnica *Checklist*

Passo 01: Analise as funções descritas no DRS, buscando responder as questões abaixo:

Q1.1 As funções descritas são suficientes para se conhecer os objetivos do sistema?

Q1.2 Todas as entradas para cada função são suficientes para a execução da função requerida?

Q1.3 Eventos indesejáveis são considerados e suas respostas são especificadas?

Q1.4 O estado inicial e os estados especiais (tais como: inicialização do sistema, término anormal, etc.) são considerados?

Classe e Tipo do possível defeito detectado: Omissão/Funcionalidade Omitida (FO)

Passo 02: Analise as características do sistema, buscando responder as questões abaixo:

Q2.1 O sistema pode ser testado, demonstrado, analisado ou inspecionado para mostrar que satisfaz os requisitos?

Q2.2 As características relativas à descrição dos itens de dados, como: tipo, taxa de variação unidade, exatidão, resolução, limites, média e valores críticos são definidos sempre que necessário?

Q2.3 As características relativas à descrição das entradas e saídas de cada função (tais como precisão, média, frequência, volume, etc.) foram definidas?

Classe e Tipo do possível defeito detectado: Omissão/ Performance Omitida (PO)

Passo 03 : Analise as funções do ambiente do sistema, buscando responder as questões abaixo:

Q3.1 A funcionalidade do hardware, software, banco de dados ou fluxo de atividades, relacionadas com o sistema estão descritos?

Classe e Tipo do possível defeito detectado: Omissão/ Ambiente Omitido (AO)

Passo 04 : Analise as interfaces do sistema, buscando responder as questões abaixo:

Q4.1 As entradas e as saídas definidas para cada uma das interfaces são suficientes?

Q4.2 Os requisitos de interface entre o sistema em si e os componentes que estão fora do escopo do sistema foram definidos?

Classe e Tipo do possível defeito detectado: Omissão/ Interface Omitida (IO)

Passo 05: Analise os requisitos do sistema, buscando responder as questões abaixo:

Q5.1 Cada requisito está definido de forma discreta, não ambíguo e testável?

Q5.2 Todas as transições entre modos ou estados do sistema são especificados claramente e corretamente?

Classe e Tipo do possível defeito detectado: Comissão/Informação Ambígua (IA)

Passo 06: Analise os requisitos do sistema, buscando responder as questões abaixo:

Q6.1 Os requisitos funcionais são consistentes entre si?

Q6.2 Os requisitos funcionais são consistentes com a visão geral?

Q6.3 Os requisitos funcionais são consistentes com o ambiente em que o sistema será executado?

Classe e Tipo do possível defeito detectado: Comissão/Informação Inconsistente (II)

Passo 07: Analise as funcionalidades do sistema, buscando responder as questões abaixo:

Q7.1 Todas as funções descritas são necessárias para se alcançar os objetivos do sistema?

Q7.2 Todas as entradas para cada função são necessárias para implementar a função?

Q7.3 As entradas e saídas, para todas as interfaces definidas, são necessárias?

Q7.4 Todas as saídas produzidas, para cada função, são usadas por outras funções ou transferidas através de alguma interface externa?

Classe e Tipo do possível defeito detectado: Comissão/Funcionalidade Incorreta (FI)

Passo 08: Analise os requisitos em suas seções, buscando responder as questões abaixo:
Q8.1 Todos os requisitos, interfaces, restrições, etc, estão listados nas seções apropriadas?
Classe e Tipo do possível defeito detectado: Comissão/Seção Errada (SE)

Passo 09: Defeitos que não se enquadram nos tipos acima.
Classe e Tipo do possível defeito: Outros (O)

B.2 TÉCNICA DE INSPEÇÃO: LBCE

Formulário de Detecção de Defeitos da Técnica Cenários

Passo 01 - Apresentação Geral do documento: Analisar e identificar as questões gerais de apresentação do documento de requisitos.

Q1.1 O documento está de acordo com o padrão exigido?

Q1.2 O documento está livre de erros de layout?

Q1.3 Todos os documentos de referência ou anteriores que o inspetor/revisor irá necessitar para seu trabalho, assim como a especificação de requisitos do sistema está disponível?

Q1.5 Os números dos requisitos do documento estão claros para facilitar a referência de localização específica durante a inspeção?

Classe e Tipo do possível defeito detectado: Outros (O)

Passo 02 - Qualidade de requisitos: Descrever a qualidade que os requisitos devem apresentar no documento de especificação de Requisitos.

Q2.1 Os requisitos estão escritos em uma linguagem simples, possibilitando o completo entendimento?

Q2.2 Todos os requisitos evitam conflitos com outros requisitos?

Q2.3 Os requisitos evitam conflitos com a especificação do projeto?

Q2.4 Os requisitos apresentam nível de detalhe apropriado?

Classe e Tipo do possível defeito detectado: Comissão/ Informação Inconsistente (II)

Passo 03 - Organização e completitude: Descreve o que o DRS deve apresentar com relação à organização e consistência dos requisitos, assim como analisar a completitude destes documentos.

Q3.1 O DRS inclui tudo que o sistema precisa saber sobre hardware, software, banco de dados, etc?

Classe e Tipo do possível defeito detectado: Omissão/ Ambiente Omitido (AO)

Passo 04 – Consistência e Tipo de Dados (CTD)

1- Identificar todos os objetos e dados mencionados na visão geral (por exemplo, componentes de hardware, variáveis de aplicação, termo ou função abreviados):

Q4.1.1 Todos os objetos mencionados na seção visão geral estão listados na seção de interface externa?

Classe e Tipo do possível defeito detectado: Omissão/ Performance Omitida (PO)

2- Para cada objeto e dados que aparecem na seção de interface externa, determinar as seguintes informações:

- Nome dos objetos
- Classe (por exemplo, entrada, saída, variável, termo abreviado, função, etc.)
- Tipo de dado (por exemplo, inteiro, real, booleano, etc.)
- Valores aceitáveis (há restrições impostas, limites de variação, etc., para os valores desse objeto?)
- Valores da falha (o objeto tem um valor específico que indica alguma falha prevista?)
- Unidades ou taxas
- Valor inicial.

Q4.2.1 A especificação do objeto ou componente é consistente com sua descrição na visão geral?

Q4.2.2 Se os objetos representam uma quantidade física, suas unidades são especificadas corretamente?

Q4.2.3 Se os valores dos objetos são computados, pode esta computação gerar um valor não aceitável?

Classe e Tipo do possível defeito detectado: Omissão/ Performance Omitida (PO)

3- Para cada requisito funcional, identificar todas as referências a objetos:

Q4.3.1 Todas as referências a objetos de dados, obedecem as convenções de formatação?

Q4.3.2 Todos os objetos referenciados no requisito funcional, são listados nas seções de entrada ou de saída?

Q4.3.3 O objeto de dados definido pode estar inconsistente com tipo do objeto, valores aceitáveis, valores de falhas, etc.?

Classe e Tipo do possível defeito detectado: Omissão/ Performance Omitida (PO)

Passo 05 – Funcionalidade Incorreta (FI)

1- Para cada requisito funcional, identificar todas as entradas/saídas de objetos de dados:

Q5.1.1 Todos os valores definidos para cada saída são consistentes com a respectiva função?

Q5.1.2 Identificar pelo menos uma função que usa cada saída.

Classe e Tipo do possível defeito detectado: Comissão/ Funcionalidade Incorreta (FI)

2- Para cada requisito funcional, identificar todos os eventos especificados no sistema:

Q5.2.1 A especificação desses eventos é consistente com as interpretações pretendidas por eles?

Classe e Tipo do possível defeito detectado: Comissão/ Funcionalidade Incorreta (FI)

3- Identificar uma invariante para cada modo ou estado do sistema (ou seja, sob que condições o sistema tem que sair ou permanecer em um determinado modo?)

Q5.3.1 As condições iniciais do sistema podem falhar e não satisfazer a(s) invariante(s) do modo inicial?

Q5.3.2 Identificar uma seqüência de eventos que permita que o sistema entre em um modo sem satisfazer a invariante identificada.

Q5.3.3 Identificar uma seqüência de eventos que permita que o sistema entre em um determinado modo, mas nunca o abandone (como em um deadlock).

Classe e Tipo do possível defeito detectado: Comissão/ Funcionalidade Incorreta (FI)

Passo 06 – Funcionalidade Omitida ou Ambigüidades (FOA): Identificar a precisão requerida, tempo de resposta, etc., para cada requisito funcional.

Q6.1 Todas as entradas para cada função são suficientes para a execução da função requerida?

Q6.2 O estado inicial e os estados especiais (tais como: inicialização do sistema, término anormal, etc.) são considerados?

Q6.3 Cada requisito está definido de forma discreta, não ambíguo e testável?

Q6.4 Todas as transições entre modos ou estados do sistema são especificados claramente e corretamente?

Classe e Tipo do possível defeito detectado: Comissão/ Informação Ambígua (IA) ou Omissão/ Funcionalidade Omitida (FO)

Passo 07 – Interface Omitida: Identificar se as entidades externas envolvidas no sistema são mencionadas no documento.

Q7.1 Todas as entidades externas envolvidas no sistema são mencionadas no escopo e estão listadas nos requisitos funcionais.

Classe e Tipo do possível defeito detectado: Omissão/ Interface Omitida (IO)

Passo 08: Seção Errada: Analisar os requisitos descritos em suas seções, de acordo com o modelo adotado:

Q8.1 Todos os requisitos descritos no DRS como: interfaces, restrições, etc, estão listados nas seções apropriadas?

Classe e Tipo do possível defeito detectado: Comissão/Seção Errada (SE)

B.3 TÉCNICA DE INSPEÇÃO: LBPE

Formulário de Detecção de Defeitos da Técnica Perspectiva em Projeto

Assuma que você está criando um projeto desse sistema. Utilize a técnica de análise estruturada e crie um diagrama de fluxo de dados (DFD) nível 0 do sistema de PCMSO definindo as entidades externas envolvidas. Incorpore todos os objetos de dados, estruturas de dados e funções necessárias para o sistema. Enquanto o projeto está sendo gerado faça a si mesmo as seguintes perguntas analisando o DFD gerado para documentar a funcionalidade que os usuários devem ter ao executar o sistema. Siga o procedimento abaixo analisando os aspectos de acordo com o projeto gerado para identificar as falhas no documento:

Passo 01 – Padrão do documento – analisar e identificar as questões gerais de apresentação do documento de requisitos.

Q1.1 O documento está de acordo com o padrão exigido?

Q1.2 O documento está livre de erros de layout?

Q1.3 Todos os documentos de referência ou anteriores que o inspetor/revisor irá necessitar para seu trabalho, assim como a especificação de requisitos do sistema está disponível?

Q1.5 Os números dos requisitos do documento estão claros para facilitar a referência de localização específica durante a inspeção?

Classe e Tipo do possível defeito detectado: Outros (O)

Passo 02 - Interface e Acesso – estes defeitos tratam do modo que o usuário ou outros envolvidos no sistema têm no acesso ao sistema. Ambos lidam com os mecanismos de acesso (a interface) bem como com as restrições ao acesso.

Q2.1 A interface não está descrita ?

Q2.2 Existe acesso adequado ao sistema descrito, bem como pedidos de acesso com tempo apropriado ?

Classe e Tipo do possível defeito detectado: Omissão/ Interface Omitida (IO)

Passo 03 - Dados – estes defeitos tratam dos dados mantidos no sistema. Isto inclui tipos de dados inconsistentes, validação de dados novos e restrições de acesso aos dados.

Q3.1 O dado possui um formato de Entrada/Saída ?

Q3.2 Os dados são conferidos contra restrições ?

Q3.3 Está claro o valor do dado que está sendo conferido ?

Q3.4 O acesso aos dados está limitado por um processo de acesso exclusivo ?

Q3.5 As Informações deletadas no sistema podem ocasionar perdas de informações importantes em outras partes do sistema ?

Q3.6 Está definido que o item do dado pode adicionar/editar/deletar ?

Q3.7 Quando o valor de um dado é atualizado, todos os itens dos dados associados contêm valores corretos ?

Q3.8 Está descrito o tipo de dado usado ?

Q 3.9 Está controlado o tipo específico de conversão ?

Q3.10 Todos os tipos possíveis de dados estão controlados ?

Classe e Tipo do possível defeito detectado: Omissão/Funcionalidade Omitida (FO)

Passo 04 - Estados – estes defeitos tratam do sistema, objetos ou estado dos dados. Enquanto os requisitos estão nivelados, os conceitos dos estados estão sempre presentes.

Q4.1 Está descrita a mudança de estado ?

Q4.2 A veracidade de uma condição põe um objeto, ou um estado particular, num conjunto de condições aceitáveis ?

Q4.3 É suposto que o sistema deverá agir quando algo acontecer, incluindo as condições os erros ?

Q4.5 Existe uma condição de erro controlada com notificação ao usuário ?

Classe e Tipo do possível defeito detectado: Comissão/Funcionalidade Incorreta (FI)

Passo 05 - Terminologia – estes defeitos ocorrem quando a terminologia usada está incorreta

ou inconsistente, gerando confusão para o usuário na documentação.

Q5.1 O termo está definido ?

Q5.2 O termo usado está correto ?

Q5.3 A terminologia é consistente ?

Classe e Tipo do possível defeito detectado: Comissão/Informação Ambígua (IA)

Passo 06 - Aspectos na formatação do documento – estes defeitos tratam de problemas na definição dos requisitos dentro do documento.

Q6.1 Os requisitos estão na seção correta ?

Q6.2 A referência do requisito está correta ?.

Classe e Tipo do possível defeito detectado: Comissão/Seção Errada (SE)

Passo 07 - Outras Inconsistências – estes defeitos tratam de situações em que os requisitos do documento descrevem contradições em duas partes diferentes do documento.

Classe e Tipo do possível defeito detectado: Comissão/Informação Inconsistente (II)

Passo 08 – Desempenho – analisar os requisitos de desempenho do sistema que envolve tempo de resposta, rotinas de backup e batch entre outros.

Q8.1 Existe a ocorrência de tempos exagerados na execução de rotinas de backup e de batch ?

Q8.2 O tempo de resposta ao usuário entre o enter e o resultado na tela é demorado?

Classe e Tipo do possível defeito detectado: Omissão/ Performance Omitida (PO)

Passo 09 – Ambiente Operacional – verificar os defeitos no DRS com relação ao hardware, software e banco de dados necessários para satisfazer os requisitos do sistema.

Q9.1 O DRS descreve as informações sobre o software, o hardware e o banco de dados que o sistema necessita ?

Classe e Tipo do possível defeito detectado: Omissão/ Ambiente Omitido (AO)

APÊNDICE C – ASPECTOS GERAIS DO ESTUDO

C.1 QUESTÕES DO ESTUDO

Escala de Avaliação das Técnicas em %															
Técnicas	Checklist					Cenários					Perspectiva				
Questões	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	20%		80%				40%	40%	20%			20%	60%	20%	
2			60%	40%		20%		60%	20%			40%	60%		
3			40%	60%				20%	80%				60%	40%	
4		20%	40%	40%				60%	40%			20%	40%	40%	
5			80%	20%				20%	80%				40%	60%	
6			40%	60%				60%	40%				40%	60%	
7				60%	40%			40%	40%	20%			40%	20%	40%
8			40%	60%					100%				40%	60%	
9			20%	40%	40%			20%	80%				20%	80%	
10			20%	80%				20%	60%	20%			20%	80%	
11		20%	40%	40%				20%	80%				40%	60%	
12		20%	40%	40%				60%	40%				40%	60%	
13	40%	20%		40%		20%	40%	40%					100%		

C.2 INFORMAÇÕES PESSOAIS

Levantamento da Experiência e dos Dados Profissionais dos Participantes									
Nível de Formação	LBCh	LBCe	LBPe		Quanto tempo atua na área de SI e outras	LBCh	LBCe	LBPe	
Superior			40%		Até 12 meses (12 meses)		20%		
Mestrando	100%	80%	60%		13 a 24 meses (18 meses)	20%		20%	
Mestre		20%			25 a 36 meses (30 meses)			60%	
Doutorando					37 a 60 meses (50 meses)	20%			
Doutor					Acima de 60 meses (90 meses)	60%	80%	20%	
Experiência sobre Inspeção de Software	LBCh	LBCe	LBPe		Conhecimento sobre Técnicas de Inspeção	LBCh	LBCe	LBPe	
Nenhuma Experiência	20%	20%			Nenhum Conhecimento		20%		
Experiência em Sala de Aula	60%	60%	60%		Conhecimento em Sala de Aula	100%	60%	80%	
Experiência em Empresa	20%	20%	20%	Conhecimento em Empresa		20%			
Experiência em Sala de Aula e Empresa			20%	Conhecimento em Sala de Aula e Empresa			20%		