

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito: 02.12.2003

Assinatura: *Ana Paula Jampaio Ferreira*

**Técnicas de leitura de especificação de
requisitos de software: estudos empíricos e
gerência de conhecimento em ambientes
acadêmico e industrial**

Erika Nina Höhn

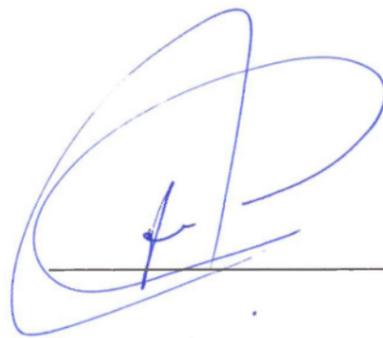
Orientador: Prof. Dr. José Carlos Maldonado

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências de Computação e Matemática Computacional.

USP – São Carlos
Dezembro/2003

A Comissão Julgadora:

Prof. Dr. José Carlos Maldonado



Profa. Dra. Kathia Marçal de Oliveira



Profa. Dra. Rosângela Aparecida Delosso Penteado



Agradecimentos

Agradeço a Deus, por tudo!!!

À minha família, em especial ao meu pai Ivo Anselmo (in memoriam) e à minha mãe Maria do Socorro, que sempre foram educadores, incentivadores, financiadores, amigos e exemplos para mim! Mamãe, obrigada pelo apoio incondicional em tudo! Aos meus irmãos Ivo Jr. e Hans Joseph, por todo o apoio, mesmo à distância! E à toda família Nina!

Ao meu orientador, Prof. Dr. José Carlos Maldonado, pela orientação, confiança, incentivo e paciência durante esse trabalho. Sua experiência, espírito empreendedor, estímulo pela pesquisa e, principalmente, a crença na minha capacidade em todos os momentos foram essenciais ao longo desse trabalho.

À Profa. Dra. Sandra Fabbri, pela colaboração e parceria nos trabalhos, pela amizade, apoio e acolhida familiar.

Ao CPqD, representado aqui por André Villas-Boas, Cláudia Tambascia, Priscilla Pagliuso e Miriam Ellen de Freitas, que pelo convênio firmado com o LABES/ICMC possibilitou a experiência de transferência tecnológica na indústria.

Ao Prof. Dr. Dorival Leão, pela atenção e orientação nos estudos estatísticos dos experimentos.

À Luciana Martímiano, pela colaboração sempre disponível e amizade sempre presente.

A Alexandre Muniz e Mário Meireles, que, desde a UFMA, foram uns dos primeiros a acreditarem e incentivarem o meu mestrado, além de serem amigos sempre presentes.

A todos do LABES e àqueles considerados amigos do LABES, pelas horas de descontração, alegria e incentivo. Um grupo onde sempre tem tudo isso e muita produção! Mesmo com todos os eventos, a pesquisa nunca pára! É um grupo que comemora e compartilha os resultados e celebra a amizade! Tentando nomear, mesmo arriscando esquecer alguém, aos amigos Adenilso, Tati, André Luís, Auri, Simone, Débora, Maris, Andrea, Tânia, Ellen, Marco, Elisa, Thaise, Mateus, Rodrigo Funabashi, Rogério Garcia, Lu, Richard, Reginaldo, Mayb, Osnete, Edilson, Valter, Juliana, Kleber, Naninha, Nilson, Bira, André Rocha, Camila, Otávio, Sandro KLB, Alessandra, Fabiano. E a todos amigos de todos os outros laboratórios!

À Juliana Moraes, pelas orientações bibliográficas.
A todos funcionários, professores e alunos que participaram e colaboraram de alguma
maneira na realização deste trabalho.
Ao CNPq, pelo financiamento ao Projeto Readers.

Resumo

Este trabalho relata replicações de experimentos e uma experiência de transferência de tecnologia desenvolvida no âmbito acadêmico para o contexto industrial. No escopo deste trabalho foram conduzidas duas replicações do experimento para avaliar a efetividade e eficiência da técnica de leitura PBR, sendo uma realizada em ambiente acadêmico e a outra em ambiente industrial, e foi iniciada a transferência tecnológica da técnica PBR para uma empresa, utilizando o processo de experimentação como base em pacotes de laboratório. Os principais resultados deste estudo são sintetizados e analisados sob a ótica do modelo de compartilhamento do conhecimento no contexto de experimentação (EKSM - *Experimentation Knowledge Sharing Model*) e do paradigma de melhoria da experimentação (EIP - *Experimentation Improvement Paradigm*). Em função dessa experiência observa-se que a experimentação traz contribuições na perspectiva da cooperação entre a indústria e a academia, possibilitando a validação de novas tecnologias, a evolução de pacotes de laboratório e a obtenção de resultados de projetos de uso real.

Abstract

This work reports replications of experiments and an experience of technology transfer from the academic to the industrial environment. Two replications were carried out to evaluate the effectiveness and efficiency of the PBR reading technique: one was applied in the academic environment and the other in the industrial environment. The technology transfer of the PBR technique to the industry used the experimentation process based on lab packages. The main results of these experiences were summarized and analyzed from the EKSM (Experimentation Knowledge Sharing Model) and EIP (Experimentation Improvement Paradigm) view points. These results provide evidences that the experimentation contributes in the cooperation between industry and academy and also makes possible to validate new technologies, evolve lab packages, and obtain results from real applications.

Sumário

1	Introdução	1
1.1	Motivação	3
1.2	Objetivo	5
1.3	Organização do Trabalho	6
2	Revisão de Documentos de Especificação de Requisitos de Software	7
2.1	Considerações Iniciais	7
2.2	Especificação de Requisitos de Software	8
2.2.1	Propriedades de Qualidade de Especificação de Requisitos	11
2.3	Tipos de Defeitos em Especificação de Requisitos de Software	13
2.4	Revisão de Software	16
2.4.1	O Processo de Inspeção	17
2.4.2	Técnicas de Leitura	21
2.5	Considerações Finais	30
3	Experimentação em Engenharia de Software	31
3.1	Considerações Iniciais	31
3.2	Estudos Empíricos	32
3.3	Processo de Experimentação	34
3.3.1	Definição	36
3.3.2	Planejamento	36
3.3.3	Operação	39
3.3.4	Análise e Interpretação	39
3.3.5	Apresentação e Empacotamento	39
3.4	Replicação de Experimentos	40
3.5	Experimento Utilizando a Técnica PBR	42
3.6	Replicações do Experimento	52
3.6.1	Replicação 1 (R1)	53

3.6.2	Replicação 2 (R2)	55
3.7	Iniciativas de Automação do Processo de Experimentação	56
3.8	Compartilhamento do Conhecimento e Melhoria da Experimentação	59
3.8.1	Compartilhamento do Conhecimento	59
3.8.1.1	Compartilhamento do Conhecimento no Contexto de Experimentação em Engenharia de Software	60
3.8.2	Paradigma de Melhoria da Qualidade (<i>QIP</i>)	61
3.8.2.1	Paradigma de Melhoria da Experimentação (<i>EIP</i>)	65
3.9	Considerações Finais	67
4	Replicações nos Ambientes Acadêmico e Industrial	69
4.1	Considerações Iniciais	69
4.2	Terceira e Quarta Replicações - R3 e R4	69
4.2.1	Replicação em Ambiente Acadêmico - R3	70
4.2.1.1	Objetivos do estudo	70
4.2.1.2	Participantes	71
4.2.1.3	Materiais	71
4.2.1.4	Procedimento	71
4.2.1.5	Resultados	73
4.2.1.6	Riscos à validade	78
4.2.2	Replicação em Ambiente Industrial - R4	79
4.2.2.1	Objetivos do estudo	79
4.2.2.2	Participantes	80
4.2.2.3	Materiais	80
4.2.2.4	Procedimento	80
4.2.2.5	Resultados	81
4.2.2.6	Riscos à validade	85
4.2.2.7	Evolução do pacote	86
4.3	Melhoria da Experimentação Durante R3 e R4	87
4.3.1	Terceiro Ciclo de Aprendizado - R3	87
4.3.2	Quarto Ciclo de Aprendizado - R4	90
4.4	Considerações Finais	93
5	Transferência da Técnica PBR para Indústria	95
5.1	Considerações Iniciais	95
5.2	Transferência de Tecnologia Utilizando Experimentação	96
5.3	Descrição dos Projetos Pilotos	100
5.3.1	Projeto Piloto 1 - P1	100
5.3.2	Projeto Piloto 2 - P2	104
5.3.3	Resultados Comparativos dos Projetos Pilotos 1 e 2	106
5.4	Étapas da Transferência das Técnicas Conforme o Paradigma EIP	111

5.4.1	Compartilhamento do Conhecimento no Processo de Transferência Tecnológica	111
5.4.2	Primeiro Ciclo de Aprendizado	113
5.4.3	Segundo Ciclo de Aprendizado	117
5.5	Síntese da Experiência Adquirida	119
5.6	Considerações Finais	121
6	Conclusões e Trabalhos Futuros	123
6.1	Trabalhos Futuros	124
	Referências Bibliográficas	127
A	Documento de Especificação de Requisitos OPER	135
B	Documento de Especificação de Requisitos OPER com Defeitos Marcados	147
C	Lista de Defeitos e Falsos Positivos do Documento OPER	159
D	Histórico da Lista de Defeitos	165
E	Caracterização dos Defeitos	207
F	Vínculo dos Defeitos com as Questões	215

Lista de Figuras

1.1	Integração Indústria & Academia	4
2.1	Inspeção Aumenta Produtividade Detectando Defeitos Quando seu Custo de Correção é Menor (Treinamento PBR - Forrest Shull (Basili et al., 1998))	20
2.2	Deteção de Defeitos Durante as Fases de Desenvolvimento - Sem Inspeções (Treinamento PBR - Forrest Shull (Basili et al., 1998))	21
2.3	Benefício da Inspeção: Melhoria da Qualidade desde Fases Iniciais (Treinamento PBR - Forrest Shull (Basili et al., 1998))	22
2.4	Conduzindo uma Inspeção (Treinamento PBR (Basili et al., 1998))	23
2.5	Composição do Cenário (adaptado do Treinamento PBR (Basili et al., 1998))	24
2.6	Famílias de Técnicas de Leitura (Treinamento PBR (Basili et al., 1998; Maldonado et al., 2004?b))	25
2.7	Documento de Requisitos e Seus Diversos Usuários (Treinamento PBR (Basili et al., 1998))	28
2.8	Cobertura <i>Ad Hoc</i> e Cobertura PBR (Treinamento PBR - Forrest Shull (Basili et al., 1998))	29
3.1	Visão Geral do Processo Experimental (Wohlin et al., 2000)	40
3.2	Processos de Experimentação e Empacotamento (Amaral, 2003)	44
3.3	Resultados Obtidos Pelos Grupos (Basili et al., 1996)	54
3.4	Resultados Obtidos Pelos Participantes (Basili et al., 1996)	55
3.5	Cobertura Obtida pela PBR no Documento Genérico ATM (Basili et al., 1996)	56
3.6	Cobertura Obtida pela PBR no Documento Específico NASA-A (Basili et al., 1996)	56
3.7	Modelo de Compartilhamento do Conhecimento (Shull et al., 2004)	65
3.8	Modelo de Compartilhamento do Conhecimento na Engenharia de Software Experimental (EKSM - <i>Experimentation Knowledge Sharing Model</i>) (Mendonça et al., 2004?)	66

3.9	Ciclos de Realimentação (<i>Feedback</i>) - QIP (Kinnula, 2001)	68
3.10	Seis Etapas do QIP (Basili et al., 1994)	70
3.11	Ciclos do FIP (Mendonça et al., 2004?)	71
4.1	Replicações USP/UFSCar (adaptado de (Mendonça et al., 2004?))	76
4.2	PBR - Efetividade versus Experiência	81
4.3	ATM/PG - Defeitos Encontrados pelas Técnicas	82
4.4	ATM: (a) Defeitos Encontrados pelas Perspectivas; (b) Defeitos Agrupados pela Perspectiva com Melhor Desempenho	84
4.5	PG: (a) Defeitos Encontrados pelas Perspectivas; (b) Defeitos Agrupados pela Perspectiva com Melhor Desempenho	85
4.6	PBR - Efetividade versus Experiência	89
4.7	ATM/PG - Defeitos Encontrados pelas Técnicas	90
4.8	ATM: (a) Defeitos Encontrados pelas Perspectivas; (b) Defeitos Agrupados pela Perspectiva com Melhor Desempenho	92
4.9	PG: (a) Defeitos Encontrados pelas Perspectivas; (b) Defeitos Agrupados pela Perspectiva com Melhor Desempenho	92
4.10	Desempenho de R1, R2 e R3 para o Documento ATM	95
4.11	Desempenho de R1, R2 e R3 para o Documento PG	95
4.12	Desempenho das Quatro Replicações para o Documento ATM	97
4.13	Desempenho das Quatro Replicações para o Documento PG	97
5.1	Elaboração da Lista de Defeitos Históricos	105
5.2	Defeitos Encontrados por Perspectiva - Projeto Piloto 1 e 2	115
5.3	Compartilhamento do Conhecimento no Processo de Transferência Tecno- lógica (adaptado de (Mendonça et al., 2004?))	117
5.4	Ciclo de Aprendizagem Durante Processo de Transferência (adaptado de (Mendonça et al., 2004?))	119

Lista de Tabelas

2.1	Tipos de Defeitos (Basili et al., 1998)	18
3.1	Projeto Experimental (Basili et al., 1996)	48
3.2	Projeto Experimental R1	59
4.1	Projeto Experimental R3	77
4.2	ANOVA em Relação à Efetividade	80
4.3	ANOVA em Relação à Eficiência	80
4.4	Resultado das Técnicas por Documento de Requisitos	81
4.5	Percentagem de Defeitos e Ocorrência de Defeitos por Tipo de Defeito	83
4.6	Percentagem Média dos Defeitos Encontrados e Razão de Defeitos Observados	83
4.7	Projeto Experimental R4	86
4.8	ANOVA em Relação à Efetividade	88
4.9	ANOVA em Relação à Eficiência	88
4.10	Resultado das Técnicas por Documento de Requisitos	89
4.11	Percentagem de Defeitos e Ocorrência de Defeitos por Tipo de Defeito	90
4.12	Percentagem Média dos Defeitos Encontrados e Razão de Defeitos Observados	91
5.1	Taxonomia do Pacote de Experimentação	105
5.2	Documentos Utilizados no P1	107
5.3	Projeto Piloto 1	108
5.4	Resultados ATM - Projeto Piloto 1	109
5.5	Resultados OPER - Projeto Piloto 1	110
5.6	Projeto Piloto 2	111
5.7	Resultados ATM - Projeto Piloto 2	112
5.8	Resultados OPER - Projeto Piloto 2	112
5.9	Comparativo dos Resultados ATM - Projeto Piloto 1 e 2	113
5.10	Comparativo dos Resultados ATM - R1, R2, R3 e R4	113
5.11	Comparativo dos Resultados OPER - Projeto Piloto 1 e 2	114

5.12	Composição da Lista de Defeitos após Projetos Pilotos	114
5.13	Ocorrência de Defeitos nos Projetos Pilotos	116
5.14	Total de Defeitos Diferentes Detectados nos Projetos Pilotos	116

Introdução

Em busca de alto nível de qualidade para seus produtos finais, as organizações se preocupam com a melhoria do processo de desenvolvimento desses produtos, uma vez que a qualidade do processo de desenvolvimento afeta diretamente a qualidade dos produtos finais. Para obter melhorias no processo é necessário compreender o processo existente e promover mudanças para melhorar a qualidade do produto e reduzir custos e tempo de desenvolvimento (Rocha et al., 2001; Sommerville, 2003).

Visando à melhoria de processos e produtos, as organizações optam pela implantação de modelos como o CMM (*Capability Maturity Model*), desenvolvido pelo SEI (Software Engineering Institute) e o projeto SPICE (*Software Process Improvement and Capability dEtermination* - futura norma ISO/EIC 15504, para avaliação de processos de software) e a certificação por normas, como a ISO 9000 para gerência da qualidade (mais especificamente as normas ISO 9001, ISO 9002 e ISO 9003).

Uma das exigências desses modelos e normas, como no caso do CMM, é a prática de atividades de VV&T (Verificação, Validação e Teste) ao longo de todo o processo. Para atender a essa exigência é necessária a adoção de métodos, técnicas e ferramentas para verificação, validação e teste que visam a minimizar a ocorrência de defeitos e riscos associados. O objetivo da verificação é assegurar que o software esteja sendo implementado

corretamente, enquanto que a validação visa a assegurar que o software que está sendo desenvolvido está de acordo com os requisitos do usuário. A atividade de teste de software consiste na análise dinâmica do software (execução do produto, sendo em código ou em especificação executável) (Rocha et al., 2001). Este trabalho de mestrado está restrito à análise estática (não há execução do produto) para verificação de documentos de especificação de requisitos que utilizam linguagem natural.

Existe a dificuldade em avaliar, escolher, possivelmente adaptar e, então, implantar novos métodos e técnicas em uma organização. A disponibilidade de dados resultantes da experiência com esses métodos e técnicas e a possibilidade de avaliá-los antes de introduzi-los na organização são essenciais para a tomada de decisão. Estudos empíricos fornecem subsídios para avaliar processos, métodos e técnicas, para obter informação objetiva e quantificável visando a prever o impacto das mudanças e também para construir uma base histórica (Rocha et al., 2001; Wohlin et al., 2000).

Pacotes de laboratório são instrumentos que viabilizam a replicação desses estudos e a aplicação de experimentos como mecanismo de transferência de tecnologias. Os pacotes de laboratório descrevem o experimento em termos específicos e fornecem infraestrutura experimental para a replicação. Eles estabelecem uma base para confirmar ou rejeitar os resultados, complementando o experimento original e modelando o objeto de estudo para o contexto experimental específico (Maldonado et al., 2004a).

O propósito da experimentação em transferência tecnológica é duplo: primeiro, antes de introduzir uma nova tecnologia a experimentação ajuda a persuadir os componentes da organização (desde a alta gerência - quanto ao investimento - quanto aos técnicos que irão utilizá-la) e ajuda a adaptar a tecnologia empacotada para as necessidades da organização alvo; e segundo, durante o uso da nova tecnologia, a experimentação ajuda a alterá-la promovendo a otimização de seus efeitos e ajuda a reforçar o uso contínuo, obtendo ganhos contínuos com isso (Rombach, 1999). O processo da transferência de tecnologia utilizando pacotes de laboratório envolve, em princípio, quatro grandes etapas: (i) Instanciação do pacote de laboratório no domínio alvo da organização; (ii) Formação de replicadores; (iii) Disseminação na organização; (iv) Disseminação e troca de experiência com a comunidade.

A transferência de tecnologia consiste em um processo de aquisição, entendimento, absorção e aplicação de uma tecnologia ou de um processo tecnológico (Cysne, 1995). O problema é que o ato da transferência requer que o conhecimento esteja sistematizado e

codificado e requer que o entendimento de como fazer algo (conhecimento tácito) se torne explícito para que possa ser comunicado a outros (Devinney, 1997). São necessários ciclos de compartilhamento do conhecimento para ocorrer a transferência tecnológica (Devinney, 1997; Nonaka e Takeuchi, 1997).

Muitas vezes a transferência de tecnologia é dificultada porque os processos de software geralmente não são incorporados no produto (Zelkowitz, 1996). Tendo a solução desse problema como um de seus objetivos Basili (1985) propôs um paradigma de melhoria da qualidade (QIP - *Quality Improvement Paradigm*) para difusão de inovações em uma organização. O modelo de melhoria do processo envolve compreensão da tecnologia, avaliação de sua aplicabilidade em um novo ambiente e empacotamento para uso geral (Zelkowitz, 1996). Com base do QIP, Mendonça et al. (2004?) propuseram o paradigma de melhoria da experimentação (EIP - *Experimentation Improvement Paradigm*) visando a gerenciar o conhecimento e melhorar o processo de replicações de experimentos controlados.

1.1 Motivação

A cooperação e a integração entre indústria e academia é fundamental no desenvolvimento e na avaliação de novas tecnologias. Deve consistir em uma relação de mútua colaboração, na qual a academia atua como agente de transferência tecnológica e de pesquisas de novas tecnologias. Por sua vez, a indústria atua como agente de validação, aprimoramento e evolução dessas tecnologias e geração de dados durante projetos reais. Os dados de experimentos acadêmicos e de experimentos e projetos industriais são importantes para alimentar a base de dados de iniciativas em estudos empíricos, como a rede ISERN (ISERN, 2003) e o Projeto CeBase (CeBASE, 2003), que se propõem a dar suporte aos pesquisadores e a definir e desenvolver novas técnicas (Figura 1.1).

A ISERN (*International Software Engineering Research Network*) é uma comunidade que, através da engenharia de software experimental, observa e experimenta as tecnologias em uso, visando a entender seus pontos fracos e fortes, a adaptá-las para os objetivos e características de projetos particulares e a empacotá-las junto com a experiência adquirida aumentando seu potencial de reutilização em outros projetos.

O CeBASE (*NSF Center for Empirically Based Software Engineering*) é uma cooperação entre universidades norte-americanas (*University of Maryland/Fraunhofer Center, University of Southern Califórnia, University of Nebraska e Mississippi State University*),

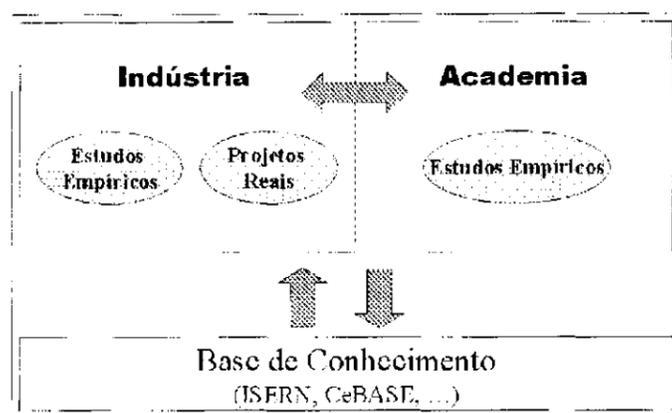


Figura 1.1: Integração Indústria & Academia

com a participação do ICMC-USP como colaborador. O CeBASE foi organizado para dar suporte a organizações de software nas respostas a questões chaves, tais como: definir qual é o melhor modelo de processo de ciclo de vida para um projeto particular; estabelecer a proporção apropriada de esforço entre inspeção e teste em um contexto específico; identificar quais são os benefícios, se houver, para comprar componentes de software disponíveis ao invés de desenvolvê-los. CeBASE reúne modelos empíricos visando a fornecer diretrizes validadas para os modelos e técnicas selecionados, a recomendar áreas para pesquisa e a fornecer apoio à educação em engenharia de software.

Com interesse no desenvolvimento de técnicas para atividades de VV&T e em estudos empíricos foi proposto o projeto Readers (*NSF-CNPq Readers Project: A Collaborative Research to Develop, Validate and Package Reading Techniques for Software Defect Detection*) - é um projeto de pesquisa colaborativa iniciado em 1999 que tem a cooperação de pesquisadores brasileiros (do ICMC-USP, UFSCar, UFRJ e UNIFACS) com apoio do CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico), e americanos (membros do Grupo de Engenharia de Software Experimental - ESEG, da Universidade de Maryland e do Centro de Pesquisa Aplicada Fraunhofer), com apoio do NSF - *National Science Foundation* (Readers, 2003). Nesse projeto, os pesquisadores agregam seus conhecimentos e experiências em leitura e teste de software assim como em estudos empíricos para desenvolver técnicas para analisar documentos de software com o objetivo de detecção de defeitos. O objetivo é contribuir para a definição de uma família de tecnologias de análise de software que devem ser validadas empiricamente em

experimentos controlados e acondicionadas em pacotes de laboratório de engenharia de software adaptáveis e reutilizáveis. As principais metas deste projeto são:

1. Criar tecnologias para apoiar a detecção de defeitos em diferentes tipos de documentos de software e desenvolver métodos empíricos que possam contribuir para melhorar essas tecnologias para diferentes ambientes e contextos culturais.
2. Desenvolver métodos de avaliação que possam comparar diversas tecnologias de detecção de defeitos em contextos industriais e acadêmicos.
3. Expandir a base tecnológica de análise de software criando novos pacotes de laboratórios que possam ser reutilizados e adaptados por outros pesquisadores baseando-se em seus ambientes e necessidades culturais.

A necessidade da indústria por técnicas de VV&T e as evidências obtidas em experimentos com técnicas de leitura (nesse caso particular, a técnica PBR - *Perspective Based Reading*), de que essas técnicas auxiliam e tornam mais produtivas as atividades de inspeção e proporcionam uma cobertura mais ampla do documento de requisitos (Shull et al., 2000), motivou a instanciação do pacote de experimentação PBR.

1.2 Objetivo

Um dos objetivos deste trabalho é apresentar os resultados de duas replicações do experimento com o pacote PBR (R3 e R4) realizadas no contexto do projeto *Readers* e analisá-las segundo o EIP (Mendonça et al., 2004?) e o modelo de compartilhamento do conhecimento para experimentação (EKSM - *Experimentation Knowledge Sharing Model*) (Shull et al., 2004). Com base nesses estudos, propôs-se a transferência tecnológica para a indústria. Outro objetivo deste trabalho apresenta a experiência de instanciação do pacote PBR para o CPqD. A instanciação do pacote objetivou a transferência tecnológica da técnica PBR para o contexto industrial, visando a dar apoio à melhoria do processo de software da indústria e também a contribuir para a validação dessa técnicas, já que estariam sendo utilizadas em um contexto diferente do acadêmico. Dentre as quatro grandes etapas de um processo de instanciação mencionadas anteriormente, apenas a de instanciação do pacote de laboratório no domínio alvo da organização é relatada neste trabalho.

Dentre as metas do projeto *Readers* citadas na Seção 1.1, este trabalho está relacionado principalmente com a meta 3, ou seja, expandir a base tecnológica de análise de software criando novos pacotes de laboratórios que possam ser reutilizados e adaptados por outros pesquisadores baseando-se em seus ambientes e necessidades culturais.

1.3 Organização do Trabalho

Neste capítulo foram apresentados o contexto no qual este trabalho está inserido, a motivação para realizá-lo e os objetivos a serem atingidos.

No Capítulo 2 apresenta-se uma revisão bibliográfica sobre aspectos de qualidade em especificação de requisitos, sobre o processo de inspeção para revisão de produtos de software e sobre técnicas de leitura que auxiliam na detecção de defeitos, abordando-se com maior destaque a técnica de leitura PBR (*Perspective-Based Reading*) utilizada ao longo deste trabalho.

No Capítulo 3 apresentam-se o processo de experimentação, ilustrado com um experimento para avaliação da efetividade da técnica PBR na detecção de defeitos em documentos de especificação de requisitos, um paradigma de melhoria desse processo (EIP) e um modelo de compartilhamento de conhecimento na Engenharia de Software experimental (EKSM).

No Capítulo 4 apresentam-se duas replicações do experimento com a técnica PBR conduzidas no âmbito do projeto *Readers*, e analisando-as conforme o paradigma de melhoria da experimentação - EIP.

No Capítulo 5 apresenta-se a experiência de transferência tecnológica da técnica PBR para o CPqD realizada através do processo de experimentação, analisando-se conforme o paradigma EIP e o modelo de compartilhamento do conhecimento.

No Capítulo 6 apresentam-se as conclusões obtidas deste trabalho e propostas de continuidade.

Revisão de Documentos de Especificação de Requisitos de Software

2.1 Considerações Iniciais

Neste capítulo faz-se uma revisão bibliográfica de especificação de requisitos, seus objetivos e propriedades, abordando aspectos de qualidade de requisitos (Seção 2.2). Na Seção 2.3 abordam-se defeitos cometidos em especificação de requisitos e algumas taxonomias. Na Seção 2.4 é apresentada a atividade de Revisão de Software, que é uma forma de verificação de produtos de software, visando à detecção de defeitos, buscando assegurar a qualidade do produto. Na Seção 2.4.1 apresenta-se o processo de inspeção, uma técnica de revisão de software. Na Seção 2.4.2 abordam-se famílias de técnicas de leitura, com ênfase na técnica PBR (*Perspective-Based Reading*), que auxiliam na fase de detecção de defeitos durante uma inspeção.

2.2 Especificação de Requisitos de Software

Um requisito bem formado é uma declaração de funcionalidade (ou capacidade) do sistema, que pode ser validado, que é definido para resolver um problema do cliente ou atingir um objetivo do sistema e que é qualificado por condições mensuráveis e limitado por restrições (IEEESTD1233, 1998).

Ainda segundo o padrão IEEESTD1233 *Guide for Developing System Requirements Specifications* (IEEESTD1233, 1998), uma especificação de requisitos de software é uma descrição do que os clientes do sistema esperam que esse faça para eles, o ambiente esperado do sistema, o perfil de uso do sistema, seus parâmetros de desempenho e sua qualidade e efetividade esperadas.

Na fase de especificação de requisitos é levantada, junto ao cliente, a descrição detalhada do software desejado, incluindo as necessidades que devem ser atendidas e as limitações e restrições a que o produto estará sujeito. Todas essas informações devem ser registradas em um documento de especificação de requisitos de software. Esse documento, além de servir como um acordo entre o desenvolvedor e o cliente nessa fase inicial, estabelecendo itens para que ambos tenham a mesma idéia do produto a ser construído, será a base para todas as fases subsequentes do desenvolvimento. Ele estará sujeito a mudanças durante todas as etapas do projeto, devendo sempre ser atualizado, por ser um referencial.

Segundo von Mayrhauser (1990), a especificação deve fornecer, tanto ao desenvolvedor quanto ao cliente, uma descrição completa e precisa da solução computadorizada proposta, descrevendo aspectos funcionais e de qualidade. A parte funcional da especificação descreve as entradas, saídas e funções do sistema, como também o estado de mudanças do sistema com as diferentes entradas. Os aspectos de qualidade da especificação, que estão nos requisitos não funcionais, detalham como as várias exigências de qualidade se manifestarão no sistema e como isso aparecerá ao usuário (von Mayrhauser, 1990). É imprescindível que se construam especificações com alto nível de qualidade e bem estruturadas. A especificação de requisitos tem três objetivos principais (Brackett, 1990):

- i) Alcançar acordo, sobre os requisitos, entre desenvolvedores de software, clientes e usuários finais;
- ii) Fornecer a base para o projeto de software e fases posteriores;

iii) Fornecer suporte à verificação e validação.

O padrão IEEE (IEEESTD1233, 1998) para desenvolvimento de especificação de requisitos de software estabelece que uma especificação de requisitos escrita adequadamente beneficia todas as fases subsequentes do ciclo de vida de diversos modos. A especificação de requisitos documenta o conjunto completo de capacidades do sistema e fornece os seguintes benefícios:

- a) segurança para o cliente de que a equipe de desenvolvedores entende as suas necessidades e reage positivamente a elas;
- b) oportunidade inicial para *feedback* bidirecional entre o cliente e a equipe de desenvolvedores;
- c) um meio para o cliente e a equipe de desenvolvedores identificarem problemas enquanto é relativamente barato corrigi-los;
- d) base para a qualificação do sistema estabelecer se esse atende às necessidades do cliente;
- e) proteção para a equipe de desenvolvedores, fornecendo uma linha de referência para capacidades do sistema e uma base de determinação de quando a construção do sistema está completa;
- f) suporte para o planejamento do projeto e esforços de desenvolvimento;
- g) possibilidade para avaliação dos efeitos das inevitáveis mudanças de requisitos; e
- h) aumento da proteção contra desentendimentos entre clientes e desenvolvedores durante o trabalho de desenvolvimento.

Sommerville (2003) sugere uma estrutura para um documento de requisitos de software, baseada no padrão IEEE (IEEESTD830, 1998), que deve ser separada em capítulos, constando:

Prefácio: Deve definir o público a que se destina o documento e descrever seu histórico da versão, incluindo uma lógica para a criação de uma nova versão e um sumário das mudanças feitas em cada versão.

Introdução: Deve descrever as necessidades que o sistema tem que atender, ou seja, os objetivos para os quais ele será construído. Deve descrever brevemente suas funções e explicar como irá interagir com outros sistemas. Deve descrever como o sistema se ajusta no contexto do negócio em que está inserido.

Glossário: Deve definir os termos técnicos usados no documento, tornando-se fundamental para estabelecer um entendimento entre os diversos participantes do projeto. Nenhuma suposição deve ser feita sobre a experiência ou especialidade dos leitores.

Definição de requisitos do usuário: Deve descrever os serviços fornecidos para o usuário e os requisitos não funcionais do sistema. Padrões de produtos e de processos a serem seguidos devem ser especificados.

Arquitetura de sistemas: Deve apresentar uma visão geral de alto nível da arquitetura de sistema prevista, mostrando a distribuição de funções por meio de módulos de sistemas. Os componentes de arquitetura que estão sendo reutilizados devem ser destacados.

Especificação de requisitos do sistema: Deve descrever os *requisitos funcionais e não funcionais* com mais detalhes. Se for necessário, outros detalhes podem também ser adicionados aos requisitos não funcionais; por exemplo, podem ser definidas interfaces com outros sistemas. Os requisitos funcionais estão ligados diretamente à funcionalidade do sistema (Rocha et al., 2001), descrevendo serviços ou funções (Sommerville, 2003), enquanto que os requisitos não-funcionais expressam as propriedades e restrições (como segurança, tempo de resposta exigido) ou qualidades específicas que o sistema deve ter.

Modelos de sistema: Deve apresentar um ou mais modelos do sistema, mostrando o relacionamento entre os componentes do sistema e o sistema e seu ambiente.

Evolução do sistema: Deve descrever as suposições fundamentais em que o sistema está baseado e as mudanças previstas, devidas à evolução de hardware, mudança nas necessidades do usuário, entre outros.

Apêndices: Deve fornecer detalhes e informações específicas relacionadas à aplicação que está sendo desenvolvida, por exemplo, descrições de hardware e bases de dados.

Índice: O documento pode conter vários índices, como índice normal alfabético, índice de diagramas, índice de funções, entre outros.

Existem outras propostas de documentos de requisitos como, por exemplo, o *Volere* - Modelo de Especificação de Requisitos - apresentado por Robertson e Robertson (2001). O *Volere* foi proposto para ser usado como uma base para documentação de especificações. Esse modelo fornece seções para todos tipos de requisitos apropriados para os sistemas de software atuais e é parte do Método de Requisitos *Volere*, uma coleção de serviços voltados ao levantamento e especificação de requisitos.

Deve-se assegurar que os requisitos identificados resultarão em uma solução possível e utilizável para o problema inteiro. Ou seja, como todo produto no processo de desenvolvimento de software, a especificação deve ser verificada e validada (von Mayrhauser, 1990). O objetivo da validação de requisitos é assegurar que documentos de requisitos contêm requisitos *efetivos* e que esses requisitos sejam *todos* os requisitos conhecidos no momento em que os documentos de requisitos foram construídos. Por outro lado, o objetivo da verificação de requisitos é garantir a qualidade dos requisitos de acordo com as propriedades de qualidade desejadas. Algumas dessas propriedades de qualidade relacionam-se à semântica dos requisitos e outras com aspectos sintáticos, estruturais ou pragmáticos dos requisitos (Durán et al., 2001).

Abordar qualidade nos requisitos de software, como requisito não-funcional, integra aspectos de qualidade na definição do próprio software (Rocha et al., 2001). Na seção seguinte apresentam-se algumas propriedades de qualidade de especificação de requisitos.

2.2.1 Propriedades de Qualidade de Especificação de Requisitos

Uma especificação de requisitos de software deve descrever um produto que atenda às necessidades e expectativas do usuário, desde que seja viável o seu desenvolvimento. E deve, também, servir de suporte a todas as fases do ciclo de vida de desenvolvimento do software (Inthurn, 2001).

O padrão IEEE *Recommended Practice for Software Requirements Specifications* recomenda que uma especificação de requisitos seja (IEEESTD830, 1998) :

- a) **Correta:** Uma especificação de requisitos está correta se, e somente se, todo requisito declarado nela for atendido no software.

- b) **Não ambígua:** Uma especificação de requisitos não é ambígua se, e somente se, todo requisito declarado nela tiver apenas uma interpretação. No mínimo, isto requer que cada característica do produto final seja descrito usando um termo simples e único.
- c) **Completa:** Uma especificação de requisitos é completa se, e somente se, incluir os seguintes elementos:
- Todos os requisitos significantes, se relacionados com funcionalidade, desempenho, restrições do projeto, atributos, ou interfaces externas. Alguns requisitos externos impostos pela especificação do sistema devem ser reconhecidos e tratados.
 - Definição das respostas dadas pelo software para todas as classes realizáveis de dados de entrada em classes realizáveis de situações. É importante especificar as respostas, tanto para valores válidos, quanto para inválidos.
 - Legendas e referências completas para todas as figuras, tabelas e diagramas e definição de todos os termos e unidades de medida.
- d) **Consistente:** Refere-se à consistência interna. Uma especificação de requisitos está internamente consistente se, e somente se, nenhum subconjunto de requisitos individuais descrito estiver em conflito, ou seja, se uma especificação de requisitos não está de acordo com algum documento de nível superior, então ela não está consistente. Características especificadas de objetos do mundo real não podem entrar em conflito: não pode haver conflito lógico ou temporal entre duas ações especificadas: se dois ou mais requisitos descrevem o mesmo objeto do mundo real, então devem usar a mesma terminologia.
- e) **Verificável:** Uma especificação de requisitos é verificável se, e somente se, todo requisito declarado nela for verificável. Um requisito é verificável se, e somente se, existe algum processo custo-benefício finito, com que uma pessoa, ou máquina, possa verificar que o produto de software atende o requisito.
- f) **Modificável:** Uma especificação de requisitos é modificável se, e somente se, sua estrutura e estilo forem tais que mudanças de requisito possam ser feitas facilmente, completamente e mantendo a estrutura e estilo consistentes.

g) **Rastreável:** Uma especificação de requisitos é rastreável se a origem de cada um dos seus requisitos for clara e se facilitar a referência de cada requisito no desenvolvimento e documentos futuros.

O padrão IEEE (IEEESTD1233, 1998) para desenvolvimento de especificação de requisitos de sistema acrescenta a propriedade de **abstração** para os requisitos, ou seja, cada requisito deve ser independente de implementação.

Inthum (2001) apresenta algumas outras propriedades desejáveis, além das citadas anteriormente:

Necessidade: Devem constar apenas itens imprescindíveis, pois requisitos que estabelecem restrições desnecessárias ou demanda de custo/tempo ao sistema comprometem o desenvolvimento.

Mensurabilidade: Esta é uma característica crítica para a fase de testes. Um requisito que não pode ser medido, não pode ser testado especificamente.

Uniformidade de terminologia: Deve haver um padrão nos termos usados ao longo de toda especificação.

Uniformidade no nível de abstração: Todos os aspectos devem estar especificados com o mesmo grau de detalhe, em cada nível de abstração.

Modularidade: Todos os itens referentes a um assunto devem estar agrupados em um mesmo módulo.

Explícita: Não podem haver hipóteses, restrições e/ou considerações implícitas na especificação.

2.3 Tipos de Defeitos em Especificação de Requisitos de Software

Todas as etapas de projeto e desenvolvimento de software são suscetíveis à introdução de defeitos inerentes à atuação humana. Segundo Deutsch (1982):

“O desenvolvimento de sistemas de software envolve uma série de atividades de produção em que as oportunidades de injeção de falhas humanas são enormes. Defeitos podem começar a acontecer logo no começo do processo, onde os objetivos do sistema de software podem estar errôneos ou imperfeitamente especificados, como também durante as fases de projeto e desenvolvimento posteriores, quando esses objetivos estão automatizados.[...] Por causa da incapacidade que os seres humanos têm de executar e comunicar com perfeição, o desenvolvimento de software é acompanhado por uma atividade de garantia de qualidade.”

Basili et al. (1999), baseando-se na terminologia padrão IEEE (*IEEE Software Engineering Standards*, IEEE CS Press, 1987), definiram defeito como sendo um erro no processo do pensamento humano, feito enquanto tenta-se entender determinada informação, resolver problemas ou usar métodos e ferramentas. No contexto de especificação de requisitos de software, um defeito é uma concepção básica equivocada das necessidades atuais de um usuário ou cliente. Defeito é qualquer propriedade de qualidade do produto que não é atendida, devendo-se evitar focar na corretitude como a primeira e única propriedade de qualidade (Treinamento PBR (Basili et al., 1998)).

Boehm (Boehm (1981) **apud** Eberlein e do Prado Leite (2002); van Lamsweerde (2000); Sallis et al. (1995)) indicou que o custo para remediar um defeito de software sobe exponencialmente com o número de fases que passam antes do defeito ser finalmente descoberto. Um defeito de requisitos não descoberto até a codificação pode levar 10 vezes o esforço para corrigi-lo; se descoberto no teste de aceitação, o custo pode ser 50 vezes maior, e se descoberto durante a operação pode ser 200 vezes o custo da fase de requisitos para corrigi-lo. De modo semelhante, defeitos de projeto não descobertos até o teste de aceitação podem levar 10 vezes o esforço para corrigi-los na fase de projeto.

As causas de defeitos (Smith e Wood, 1989) em especificações de requisitos são devidas a:

1. requisitos incorretos (ex.: o modelo não representa adequadamente a situação real);
2. requisitos inconsistentes ou incompatíveis (ex.: conflito de informações);
3. requisitos ambíguos ou ilógicos;

1. requisitos omissos (ex.: ausência de procedimento para tratamento de entradas inválidas).

Um fator a ser considerado, quanto à ocorrência de defeitos, é a linguagem utilizada na especificação de requisitos. As linguagens de especificação podem ir do formal, como notação matemática, à linguagem natural, como uma narrativa em inglês ou português (von Mayrhauser, 1990). As especificações que utilizam linguagem podem ter problemas, devido à ambigüidade inerente da linguagem natural, e enganos, porque não existe uma terminologia padrão de computação (Sommerville, 1995).

Os defeitos podem ser classificados em defeitos de criação e defeitos de omissão. O defeito de criação caracteriza-se quando existe a informação, mas ela está errada. No defeito de omissão, uma informação necessária ao sistema é omitida do produto de software.

Tanto a omissão de funcionalidade quanto a incorporação de funcionalidade desnecessária tendem a ter conseqüências caras. O desenvolvimento de funções supérfluas requer o esforço do pessoal de desenvolvimento e envolve o uso de recursos. Além disso, pode causar a introdução de defeitos em outras funções (devido às inconsistências) e pode resultar, subseqüentemente, em testes extensos e desnecessários. Sallis et al. (1995) comentam que, em geral, achar funções erroneamente incluídas em uma especificação é mais difícil que identificar as que foram omitidas, principalmente porque o usuário tende a notar o que está faltando mais do que o que é incluído. Nos Capítulos 3 e 4 apresentam-se resultados dos experimentos realizados utilizando técnicas de leitura para inspeção de documentos de especificação de requisitos, nos quais, em alguns casos, os defeitos de omissão foram os mais detectados e em outros os defeitos de criação, indicando uma possível influência do domínio do documento utilizado na inspeção.

Na Tabela 2.1 são exemplificados alguns tipos de defeitos que podem ser cometidos em documentos especificação de requisitos.

Não existe uma única taxonomia de defeitos. Segundo Rocha et al. (2001), as mais conhecidas são:

1. Taxonomia de Beizer (1990)
2. Taxonomia da IBM (*Orthogonal Defect Classification*) (1992)
3. Taxonomia de Basili e Perricone (1994)
4. Taxonomia de DeMillo e Mathur (1995)

Tabela 2.1: Tipos de Defeitos (Basili et al., 1998)

Tipo	Descrição
Omissão	Informação necessária sobre o sistema que tem sido omitida do documento de requisitos.
Ambigüidade	Informação no documento de requisitos é ambígua, isto é, várias interpretações podem ser derivadas.
Inconsistência	Informação em uma das partes do documento de requisitos está inconsistente com informação contida em outra parte.
Fato Incorreto	Alguma informação contida no documento de requisitos afirma um fato que pode não ser verdade sobre as condições especificadas nos requisitos gerais ou no conhecimento de domínio geral.
Informação Estranha	Uma informação é fornecida, mas não é necessária ou não é utilizada.
Diversos	Outros defeitos.

2.4 Revisão de Software

A meta primária de desenvolvimento de software é gerar sistemas que satisfaçam às necessidades do usuário. Entretanto, os vários produtos associados com o desenvolvimento de software (por exemplo, documentos de especificação de requisitos, código e planos de teste) freqüentemente requerem revisões e modificações durante todo o ciclo de vida de desenvolvimento.

Para manter a incidência mínima de defeitos é crucial que esses produtos estejam sujeitos a procedimentos de garantia de qualidade. Garantia de qualidade objetiva assegurar que padrões, previamente estabelecidos, sejam alcançados antes de qualquer produto ser usado como base para a próxima etapa do desenvolvimento (Sallis et al., 1995).

Segundo Sallis et al. (1995): “Até mesmo nas primeiras fases, a garantia de qualidade é uma atividade vital. Os participantes do software - os clientes, usuários, desenvolvedores e gerentes de projeto - todos têm expectativas diferentes do produto e da qualidade do processo. Revisões, *walkthroughs* e inspeções durante as fases de análise e de projeto ajudam assegurar que um entendimento comum de requisitos de qualidade é alcançado entre todas as partes.”

Muitas das atividades da garantia da qualidade de software (GQS) são atividades de Verificação e Validação (VV&T) e incluem revisões técnicas formais, auditorias de

configuração e qualidade, monitoração do desempenho, simulação, estudo da viabilidade, revisão da documentação, revisão dos bancos de dados, análise de algoritmos, teste de desenvolvimento, teste de qualificação e teste de instalação (Pressman, 2002).

Em termos de requisitos, uma revisão é um processo manual, que envolve múltiplos leitores do grupo de clientes e do grupo de desenvolvedores conferindo o documento de requisitos, procurando anomalias e omissões. A revisão geralmente tem grande êxito na descoberta de defeitos nas especificações de requisitos (Sommerville, 2003).

Uma forma de revisão é a inspeção, que é um método de análise estática para verificar propriedades de qualidade de produtos de software, sendo a mais rigorosa forma de revisão (Pressman, 2002). Inspeção de software tem sido utilizada na engenharia de software como um dos métodos mais eficientes e efetivos para melhorar a qualidade dos produtos de software. Como ela pode ser executada no fim de cada fase de desenvolvimento e, considerando-se também que os defeitos são geralmente encontrados próximos ao ponto em que eles foram introduzidos, os custos de re-trabalho podem ser reduzidos consideravelmente.

Eberlein e do Prado Leite (2002) defendem que a adição da atividade de verificação, executada através de inspeções junto com validação, beneficiaria os métodos ágeis, melhorando a qualidade dos processos ágeis, que usualmente contam apenas com a validação.

Considerando que o custo para corrigir um defeito aumenta rapidamente com os progressos do processo de desenvolvimento, como é mostrado na Figura 2.1, a detecção de defeitos por uso de inspeções é promissora (Andriole, 1986).

Na Figura 2.2 é apresentado o percentual de defeitos encontrados em cada fase de desenvolvimento sem a utilização de inspeções. Pode-se observar que uma grande quantidade de defeitos passa de uma fase para outra, sendo muitas vezes descobertos mais na fase seguinte do que na que foram introduzidos. Quando se utiliza inspeção desde as fases iniciais do desenvolvimento do software, os defeitos são descobertos em maior número na fase em que foram introduzidos, como é mostrado na Figura 2.3.

2.4.1 O Processo de Inspeção

A inspeção foi desenvolvida por Michael Fagan em 1976 (Smith e Wood, 1989). O processo de inspeção, como definido por Fagan (Fagan (1976) *apud* Shan-Jarvis e Crandall (1997)), envolve os seguintes passos:

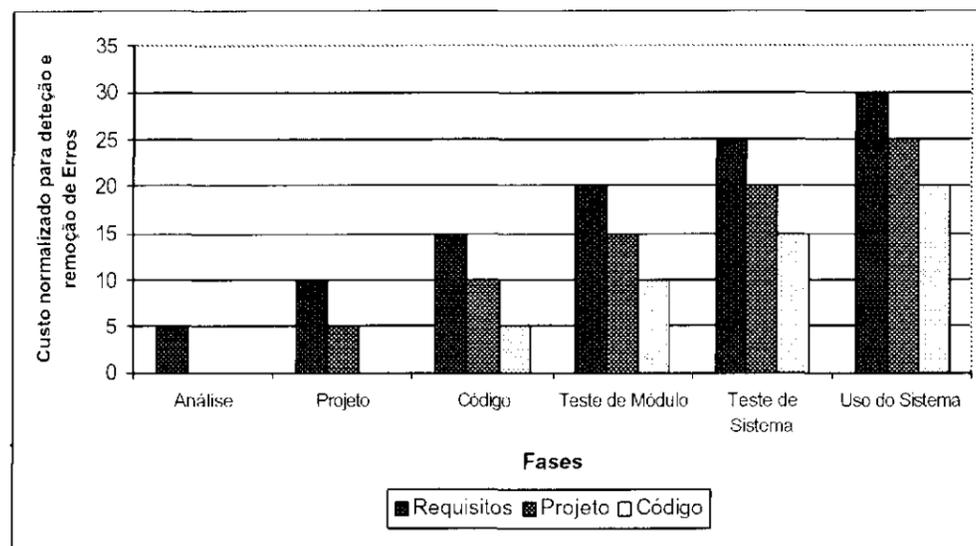


Figura 2.1: Inspeção Aumenta Produtividade Detectando Defeitos Quando seu Custo de Correção é Menor (Treinamento PBR - Forrest Shull (Basili et al., 1998))

1. **Planejamento:** identificar os participantes, preparar materiais e programar o treinamento;
2. **Treinamento:** treinar os participantes;
3. **Preparação:** preparação individual para as inspeções;
4. **Inspeção:** execução da inspeção para identificar defeitos;
5. **Correção:** corrigir os defeitos registrados;
6. **Follow-up:** assegurar que todos os defeitos estão corrigidos.

O processo de inspeção de software pode ser agrupado em quatro passos consecutivos (Figura 2.4): planejamento, detecção de defeitos, reunião (coletar defeitos) e atividades de pós-inspeção (correção de defeitos). No passo de planejamento, cada membro da equipe de inspeção familiariza-se com o material e com a técnica adotada. No passo de detecção, eles revêm individualmente o produto para encontrar defeitos. Na reunião, os membros da equipe discutem os defeitos das revisões individuais, revêm brevemente o produto para encontrar outros defeitos e preparam um relatório de defeitos encontrados pela equipe.

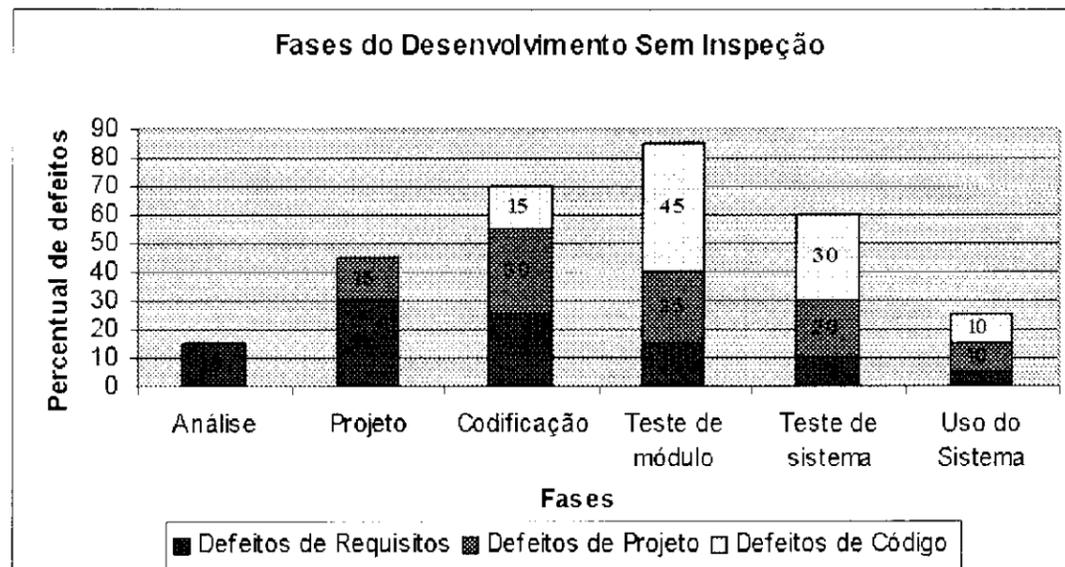


Figura 2.2: Detecção de Defeitos Durante as Fases de Desenvolvimento - Sem Inspeções (Treinamento PBR - Forrest Shull (Basili et al., 1998))

Nas atividades de pós-inspeção, o autor do produto corrige os defeitos contidos no relatório de defeitos da equipe e as correções são então revistas em uma nova inspeção (Basili et al., 1996).

Embora cada uma dessas atividades seja importante para uma inspeção bem sucedida, a chave de uma inspeção é a atividade de detecção de defeitos. Durante essa atividade, os revisores lêem documentos de software e avaliam se eles satisfazem os requisitos de qualidade, tais como exatidão, consistência, testabilidade e manutenibilidade.

Os benefícios qualitativos atribuíveis ao uso de inspeções são relevantes, como os seguintes (Andriole, 1986; Basili et al., 1998):

- programas ficam menos complexos;
- subprogramas são escritos em um estilo consistente e obedecem padrões estabelecidos;
- o desenvolvimento de sistemas se torna altamente transparente;
- a estimativa e o planejamento tornam-se mais confiáveis;

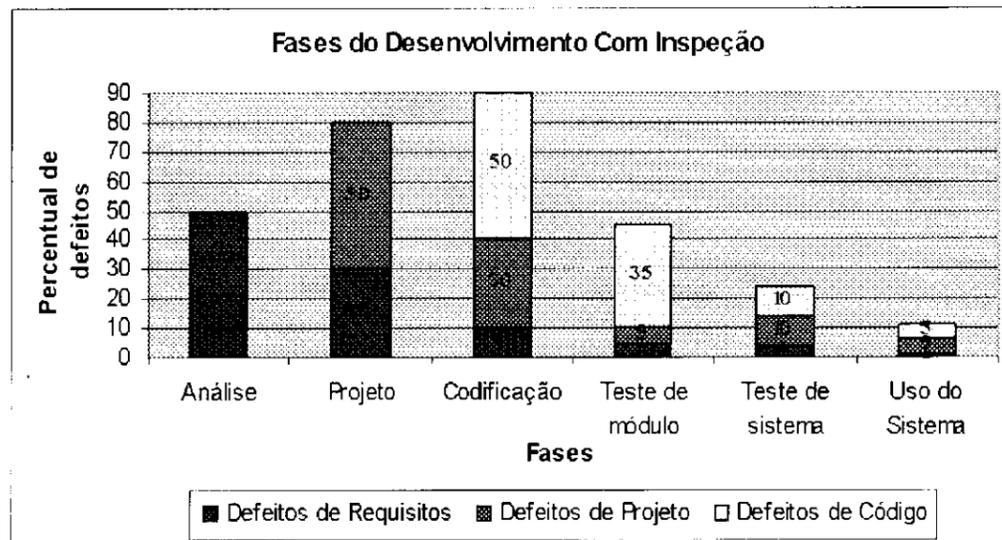


Figura 2.3: Benefício da Inspeção: Melhoria da Qualidade desde Fases Iniciais (Treinamento PBR - Forrest Shull (Basili et al., 1998))

- aumenta o aprendizado e experiência de todos os indivíduos envolvidos no processo de inspeção;
- convence os membros da equipe de desenvolvimento a desenvolverem produtos mais compreensíveis e bem documentados, facilitando-se também a manutenção;
- aumenta a satisfação do usuário;
- permite registro de dados de defeitos; e
- pode ser usada para certificação da qualidade do produto.

As inspeções podem ser executadas durante todas as fases do desenvolvimento e, para a maioria dos tipos de aplicações, em projetos grandes ou pequenos.

Seu custo principal é dos recursos humanos envolvidos. E esses custos se tornam irrisórios quando comparados com o ganho na prevenção de defeitos maiores em fases posteriores.

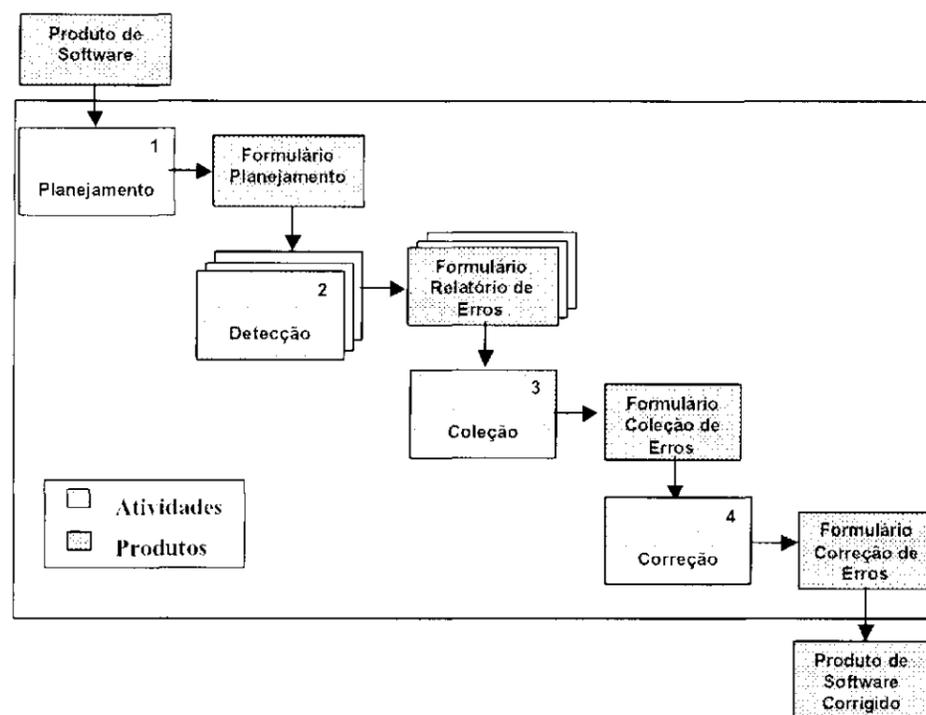


Figura 2.4: Conduzindo uma Inspeção (Treinamento PBR (Basili et al., 1998))

2.4.2 Técnicas de Leitura

Implementar uma inspeção requer uma compreensão precisa das suas tarefas. A fase de descoberta de defeitos em uma inspeção é a mais significativa. Os revisores devem ler um documento de software (por exemplo, especificação de requisitos) e podem aplicar alguma técnica (formal ou informal) para auxiliar a descoberta de defeitos. Profissionais da área de desenvolvimento de software aprendem como escrever documentos de software, mas quando exercem o papel de revisores possuem pouca perícia para ler (Shull et al., 2000).

Em geral, a maioria das implementações de inspeção usa abordagem *ad-hoc* durante a deteção de defeitos. Por não fornecer orientação explícita aos revisores de como proceder, ou o quê especificamente procurar durante a atividade de leitura, os revisores que utilizam abordagem *ad hoc* seguem sua própria intuição, conhecimento e experiência para tentarem identificar defeitos em um documento de software.

Em contrapartida, uma técnica de leitura fornece um conjunto definido de instruções que explicam como ler um documento de software e o quê um revisor deve procurar (Basili

et al., 1996). Várias técnicas têm sido estudadas para auxiliarem os revisores na leitura de produtos de software (como documentos de requisitos, projeto, código).

Uma técnica é a leitura baseada em *Checklist*, que oferece como apoio uma lista de perguntas, a qual os revisores devem responder enquanto lêem um documento de software. A finalidade das perguntas é fornecer alguma direção na busca por defeitos.

Outra técnica é a leitura baseada em cenários, que tem sido proposta para solucionar algumas das deficiências da *ad hoc* e *Checklist*. A idéia básica dessa técnica é o uso de cenários, que descrevem **como** fazer para descobrir uma informação em um documento de software, assim como **o quê** aquela informação representa.

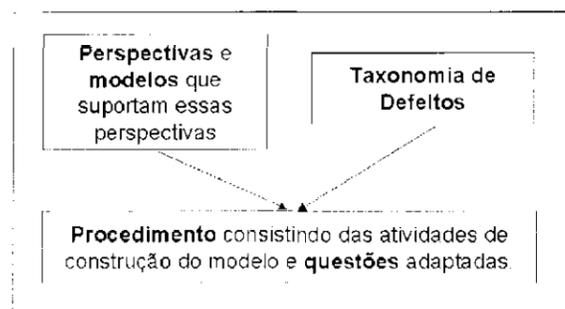


Figura 2.5: Composição do Cenário (adaptado do Treinamento PBR (Basili et al., 1998))

As técnicas que utilizam cenários designam responsabilidades distintas e específicas para cada revisor, além de orientação de como proceder através dos cenários. Um cenário consiste de um conjunto detalhado de atividades que devem ser executadas pelo revisor enquanto ele estiver lendo o produto de software e de um conjunto de perguntas relacionadas às atividades, que ajudam os revisores a descobrirem defeitos (Figura 2.5) (Basili et al., 1996).

Na Figura 2.6 são apresentadas diversas famílias de técnicas de leitura divididas em dois propósitos gerais, o de análise de produtos de software e o de construção. Dentro de cada uma dessas linhas as técnicas foram desenvolvidas para vários propósitos específicos. Cada família, e conseqüentemente cada técnica, está associada com um produto de software particular (por exemplo, código, interface do usuário) e uma notação (por exemplo, código fonte). A família de técnicas PBR utilizada neste trabalho foi desenvolvida com o objetivo de análise para a detecção de defeitos em requisitos que utilizam a linguagem

natural inglês e suas técnicas são baseadas nas perspectivas do desenvolvedor, do testador e do usuário.

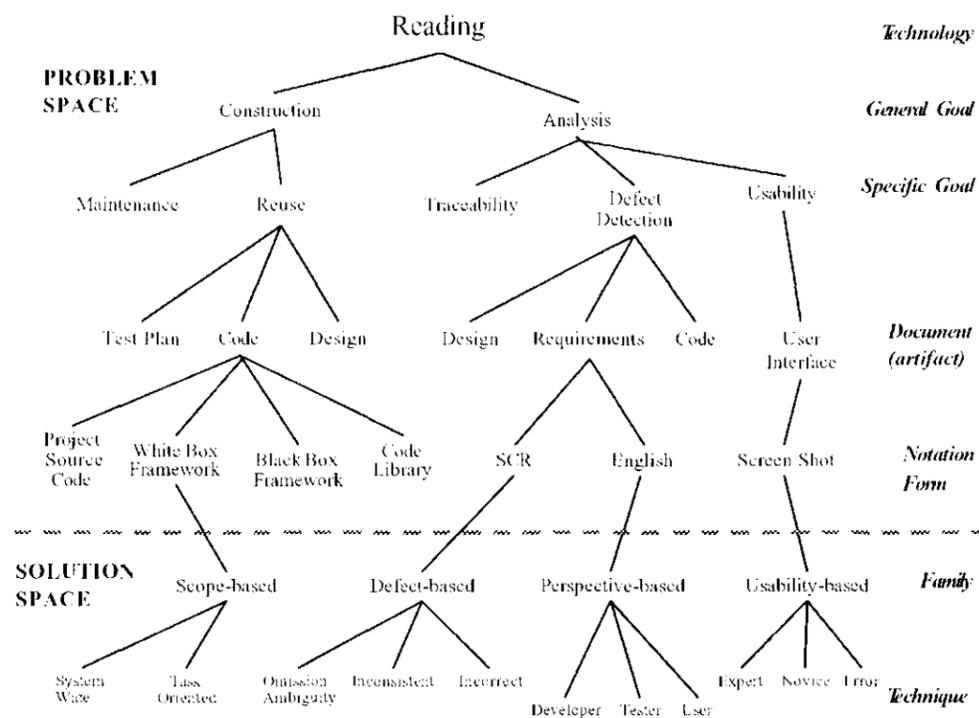


Figura 2.6: Famílias de Técnicas de Leitura (Treinamento PBR (Basili et al., 1998; Maldonado et al., 2004?b))

A seguir, apresentam-se algumas técnicas de leitura baseadas em cenário, com maior ênfase à técnica de leitura baseada em perspectiva, devido à sua relevância neste trabalho.

- **Leitura Baseada em Defeito (*Defect-Based Reading - DBR*)**

A DBR é focalizada nos diferentes tipos de defeitos. Um conjunto de procedimentos é fornecido para guiar na detecção de classes particulares de defeitos. O revisor deve criar um modelo para uma classe específica de defeitos (por exemplo, falta de funcionalidade) e então responder uma lista de perguntas. Cada revisor segue um cenário correspondente a uma classe de defeito específica e depois a equipe combina os cenários (Basili et al., 1999; Porter et al., 1995).

- **Leitura Baseada em *Checklist* (*Checklist-Based Reading* - CBR)**

A técnica utiliza questionários, que os revisores devem responder sobre o documento que está sendo examinado. As perguntas são baseadas nos tipos de defeitos mais comumente encontrados na organização.

Embora a técnica baseada em *checklist* ofereça mais apoio para a leitura que a *ad-hoc*, ela ainda não atende alguns fatores. Um desses é o fato de que um *checklist* é freqüentemente baseado em informações de defeitos anteriores. Se não há informações disponíveis, então reutilizam-se de outras organizações, podendo ocorrer casos em que alguns tipos de defeitos, ou até classes de defeitos inteiras que não tenham sido previamente detectadas, não serão contemplados no *checklist*. Outro fator é a responsabilidade do revisor tanto na abordagem *ad hoc* quanto na técnica *Checklist*, pois todos os revisores procuram por todos os tipos de defeitos, e um mesmo documento não é reavaliado por outros revisores, dependendo assim unicamente da revisão individual (Basili et al., 1999).

- **Leitura Baseada em Uso (*Use-Based Reading* - UBR)**

Essa técnica utiliza casos de uso para guiar os revisores na inspeção. A UBR dirige os revisores para focalizarem nas partes do software que são mais importantes para os usuários (Basili et al., 1999).

- **Leitura Baseada em Escopo (*Scope-Based Reading* - SBR)**

Consiste em duas técnicas de leitura que foram desenvolvidas para aprendizado de uma estrutura orientada a objeto, visando à reutilização na construção de um novo sistema, usando cobertura parcial de projeto e código orientado a objeto (Basili et al., 1999).

- **Técnicas de Leitura Orientadas a Objetos (*Object-Oriented Reading Techniques* - OORT)**

Uma família de técnicas para leitura de projeto orientado a objetos (*Object-Oriented Reading Techniques* - OORT) foi desenvolvida com o propósito de fornecer um procedimento para inspeção de diagramas e documentos de projetos orientados a objetos. Essas leituras são divididas em duas categorias: leituras horizontais e verticais. A leitura horizontal verifica a eficiência e a consistência dos diagramas de

projeto. A leitura vertical verifica a consistência entre os artefatos de projeto e os requisitos do sistema (Rocha et al., 2001; Travassos et al., 1999).

No âmbito do projeto Readers estão sendo executados trabalhos para explorar a técnica OORT nos artefatos de projeto gerados ao longo do processo de construção dos estudos de casos. Replicações utilizando pacotes de laboratório estão sendo executadas pelo grupo de Engenharia de Software Experimental, da COPPE/UFRJ (COPPE/UFRJ, 2003; Travassos et al., 1999).

Um outro trabalho define uma estratégia de inspeção para um processo particular de desenvolvimento de software OO, denominado ProDeS/UML (Processo de Desenvolvimento de Software baseado na UML). Essa estratégia é composta por um conjunto de técnicas de leitura, denominado OORTs/ProDeS, as quais avaliam os documentos UML gerados durante as fases de Engenharia de Requisitos, Análise e Projeto que compõem esse processo (Marucci, 2002).

- **Leitura Baseada em Perspectiva (*Perspective-Based Reading-PBR*)**

A técnica PBR foi criada por pesquisadores no Grupo de Engenharia de Software Experimental da Universidade de Maryland (EUA) objetivando sistematizar e dar mais eficácia à tarefa de revisar documentos de requisitos em linguagem natural (Shull et al., 2000). A PBR utiliza cenários de leitura baseados em perspectivas. Um cenário apóia a verificação de um documento sob uma perspectiva particular.

A técnica PBR fornece um conjunto de procedimentos visando a ajudar desenvolvedores a lidarem com problemas em inspeção de documentos de requisitos de software. Cada revisor exerce um papel específico baseado no uso que ele faz daquele documento.

Muitas pessoas diferentes usam um documento de requisitos para dar suporte às suas tarefas durante o ciclo de vida de desenvolvimento do software, cada uma com suas próprias necessidades. Por exemplo, em relação a um documento de especificação de requisitos (Figura 2.7):

1. um projetista cria um projeto do sistema usando como base os requisitos;
2. um testador produz um plano de teste para garantir que o software está de acordo com os requisitos funcionais e de desempenho; e

3. para um usuário, os requisitos devem capturar adequadamente a funcionalidade necessária ao sistema final.

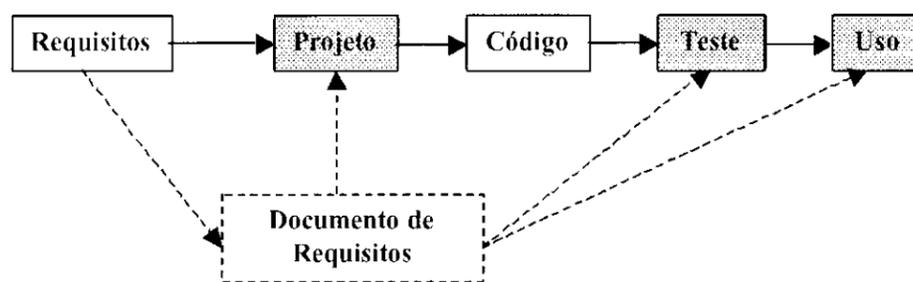


Figura 2.7: Documento de Requisitos e Seus Diversos Usuários (Treinamento PBR (Basili et al., 1998))

Esses diversos usos sugerem perspectivas para revisão do documento de requisitos. Dependendo do ambiente em que a técnica PBR é aplicada, pode-se achar um conjunto diferente de perspectivas mais aplicáveis.

Assim, em uma inspeção com PBR, cada revisor em uma equipe assume a perspectiva de um usuário específico do documento. O revisor cria uma versão de alto nível do produto que normalmente é produzido pelo usuário que ele representa, ou seja, um modelo, uma abstração do documento que está sendo revisado (modelo subjacente), para aumentar sua compreensão. Das perspectivas do projetista, testador e usuário, os produtos pertinentes seriam um projeto de sistema de alto nível, um plano de teste do sistema e um conjunto de casos de usos enumerando as funcionalidades descritas, respectivamente.

O uso de diferentes perspectivas caracteriza a técnica como (Shull et al., 2000):

1. **Sistemática:** a identificação de diferentes usos para os requisitos possibilita que os revisores tenham um procedimento definido para efetuar a verificação.
2. **Focalizada:** a técnica PBR ajuda os revisores a se concentrarem mais efetivamente em certos tipos de defeitos, em vez de procurar todos possíveis tipos. Isso evita esforço duplicado e sobreposição de responsabilidades entre os membros da equipe. A união das perspectivas fornece uma cobertura extensa do documento (Figura 2.8).

3. **Orientada à meta e adaptável:** Em cada organização, as perspectivas podem ser mudadas para refletir as metas específicas da inspeção. A PBR não é prescrita estritamente em um conjunto definitivo de procedimentos.
4. **Transferível via treinamento:** Como a técnica PBR trabalha com procedimentos definidos, e não apenas com a própria experiência do revisor, podem-se treinar novos revisores nos procedimentos.

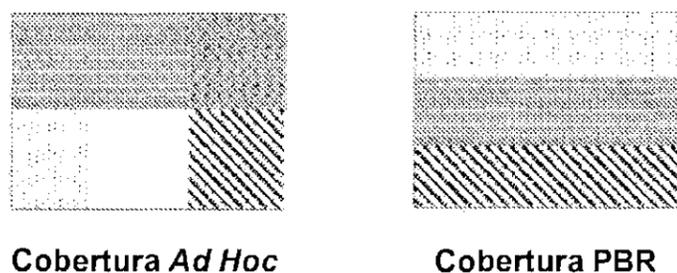


Figura 2.8: Cobertura *Ad Hoc* e Cobertura PBR (Treinamento PBR - Forrest Shull (Basili et al., 1998))

A técnica PBR ajuda os revisores a responderem duas importantes questões sobre os requisitos que eles estão inspecionando: “Que informações desses requisitos devem ser verificadas? Como identificar defeitos nessas informações?” (Shull et al., 2000).

Com as abstrações dos requisitos criadas, os revisores ainda precisam determinar que defeitos podem existir. A técnica PBR fornece uma lista de questões, desenvolvidas para cada tipo de defeito, que os revisores devem tentar responder. Os requisitos que não fornecem informação suficiente são relacionados em um relatório com uma justificativa do problema e classificados de acordo com a taxonomia de defeitos adotada.

A classificação dos defeitos não é ortogonal, pois um determinado defeito pode ser atribuído a mais de uma categoria e pode depender da interpretação do revisor. Cada organização pode acrescentar outras categorias, dependendo de suas necessidades (Shull et al., 2000).

A seguir são exemplificados os procedimentos de leitura para as três perspectivas da PBR, apresentando-se parte dos cenários do testador, do projetista e do usuário (Basili et al., 1996):

Perspectiva baseada no teste:

Para cada requisito, fazer um teste ou conjunto de testes que permitirão assegurar que a implementação satisfaz o requisito. Usar a abordagem de teste e critério de teste usuais para fazer o conjunto de testes. Enquanto estiver montando o conjunto de testes, para cada requisito, fazer as seguintes perguntas:

1. Há todas as informações necessárias para identificar o item que está sendo testado e seu critério de teste? Casos de teste razoáveis podem ser feitos para cada item, baseado no critério?
2. Há outro requisito para o qual poderia ser elaborado um caso de teste similar, mas que geraria um resultado contraditório?
3. Você pode assegurar que o teste resultará no valor correto nas unidades corretas?
4. Há outras interpretações desse requisito que o implementador possa fazer, baseado no modo como o requisito está definido? Isto afetaria o teste feito?
5. O requisito faz sentido com o que você conhece sobre a aplicação e com o que está especificado na descrição geral?

Perspectiva baseada no projeto

Gerar um projeto do sistema a partir do qual possa ser implementado. Utilizar abordagem própria e técnica padrões de projeto e incorporar todos os objetos de dados, estruturas de dados e funções necessárias. Enquanto o projeto estiver sendo gerado, fazer a si mesmo as seguintes perguntas:

1. Todos os objetos necessários (dados, estrutura de dados, funções) estão definidos?
2. Todas as interfaces estão especificadas e consistentes?
3. Todos os tipos de dados podem ser definidos? (exemplo: precisão e unidade requerida estão especificadas?)
4. Toda informação necessária para fazer o projeto está disponível? Todas as condições envolvendo todos os objetos estão especificadas? (exemplo: alguma especificação de requisitos/funcional está faltando?)

5. Existe algum ponto que não está claro para você, sobre o que você deveria fazer, porque a especificação de requisitos/funcional não está bem clara ou não está consistente?
6. A especificação de requisitos/funcional faz sentido a partir do que você conhece sobre a aplicação, ou do que está especificado na descrição geral/introdução?

Perspectiva baseada no uso

Definir o conjunto de funções que um usuário do sistema deveria estar apto a executar. Definir um conjunto de objetos de entrada necessário para executar cada função e o conjunto de objetos de saída que é gerado por cada função. Isto pode ser visto como descrever todos os cenários operacionais ou subconjuntos de cenários operacionais que o sistema deveria executar. Iniciar com os cenários mais convencionais e prosseguir para os cenários menos comuns ou de condições especiais. Enquanto os cenários estiverem sendo gerados, fazer a si mesmo as seguintes perguntas:

1. Todas as funções necessárias para escrever o cenário operacional estão descritas na especificação de requisitos/funcional? (exemplo: todas as capacidades listadas na descrição geral estão especificadas?)
2. Todas as condições iniciais para começar um cenário operacional estão claras e corretas?
3. As interfaces entre as funções estão bem definidas e compatíveis? (exemplo: fazer as entradas de uma função ligadas para as saídas da função anterior)
4. Você pode entrar em um estado do sistema que deve ser evitado? (exemplo: por razões de cuidado e segurança)
5. Alguma porção de um cenário operacional pode dar uma resposta diferente dependendo de como uma especificação de requisitos/funcional é interpretada?
6. A especificação de requisitos/funcional faz sentido a partir do que você conhece sobre a aplicação ou do que está especificado na descrição geral/introdução?

Uma questão fundamental é como gerar bons cenários. Segundo Basili et al. (1999), existem várias razões que dificultam a geração dos cenários, são elas:

- Existem vários níveis de abstração em um documento de especificação de software;

- Não existe um padrão para documentos de especificação. Um documento de especificação de requisitos pode conter: breve esboço, definições de entrada/saída, requisitos funcionais, funções dos algoritmos e outros aspectos que são importantes para o sistema;
- Vários métodos e técnicas podem ser utilizados dentro de cada perspectiva.

O experimento piloto, base para outros experimentos realizados para avaliação da técnica PBR, e as replicações conduzidas no âmbito do projeto *Readers* são descritos nos Capítulos 3 e 4.

2.5 Considerações Finais

Neste capítulo foi discutida a relevância do documento de especificação de requisitos no desenvolvimento do software. Devido à necessidade de se garantir a qualidade desse documento, vários padrões para sua elaboração foram propostos na literatura sendo que alguns deles e suas propriedades de qualidade foram brevemente comentados.

Assim, dada a importância desse documento, torna-se indispensável que atividades de VV&T sejam consideradas. Apresentou-se o processo de inspeção de software, que é um método de análise estática que auxilia na verificação de propriedades de qualidade de produtos de software. As características e vantagens e algumas técnicas de leitura que apóiam a realização desse processo no caso de avaliação do documento de requisitos também foram apresentadas. Deu-se ênfase à técnica de leitura PBR, a qual corresponde à técnica utilizada nos experimentos que serão mencionados nos capítulos seguintes e que foi alvo de transferência para a indústria. Essa técnica, ao contrário de uma inspeção *ad-hoc*, em que cada inspetor utiliza sua experiência para avaliar o documento de requisitos, é uma técnica sistemática e operacional que propõe algumas diretrizes para sua aplicação.

No Capítulo 3 serão apresentados o processo de experimentação utilizado nos estudos que têm avaliado a técnica de leitura PBR, um paradigma de melhoria desse processo, um modelo de compartilhamento de conhecimento associado ao processo experimental e os principais resultados de dois desses experimentos.

Experimentação em Engenharia de Software: Especificação de Requisitos de Software

3.1 Considerações Iniciais

Estudos empíricos são importantes em engenharia de software na obtenção de resultados objetivos e significativos em relação à compreensão, controle e avaliação de técnicas ou processos de desenvolvimento de software. Neste capítulo, na Seção 3.2, abordam-se tipos de estudos empíricos utilizados na engenharia de software. Na Seção 3.3 detalha-se o processo de experimentação e suas fases. Na Seção 3.4 apresentam-se os tipos de replicações de experimentos e suas características. Na Seção 3.5 exemplifica-se o processo de experimentação na avaliação da efetividade da técnica de leitura PBR (apresentada na Seção 2.4 do Capítulo 2) na detecção de defeitos em documentos de especificação de requisitos, quando comparada com outras técnicas. Na Seção 3.6 abordam-se replicações desse experimento, realizadas na Universidade de São Paulo e na Universidade Federal de São Carlos, como parte do projeto de pesquisa colaborativa, *Readers Project*.

3.2 Estudos Empíricos

“Um estudo empírico é um ato ou operação com a finalidade de descobrir algo desconhecido ou de provar uma hipótese, envolvendo um investigador coletando dados e executando análise para determinar o que os dados significam.”
(Basili et al., 1999)

Estudos empíricos são importantes em engenharia de software quando se busca obter resultados objetivos e significativos em relação à compreensão, controle, prognóstico e melhoria nos processos de desenvolvimento de software (Wohlin et al., 2000).

Os seguintes tópicos são necessários para obter êxito no desenvolvimento de software (Wohlin et al., 2000):

1. Compreensão do processo e produto de software;
2. Definição de processo de desenvolvimento e qualidade do produto;
3. Avaliação de sucessos e falhas;
4. Realimentação de informação para controle do projeto;
5. Aprendizagem com a experiência;
6. Empacotamento e reutilização da experiência relevante.

Para atender tais necessidades pode-se necessitar obter dados através de estudos empíricos. Segundo Wohlin et al. (2000), os tipos de estudos empíricos em engenharia de software incluem:

a. Execução de pesquisa de opinião (do inglês *survey*)

Uma pesquisa de opinião é geralmente uma investigação executada em retrospecto, quando, por exemplo, uma ferramenta ou técnica já está em uso durante algum tempo. Os meios básicos para obtenção de dados qualitativos ou quantitativos são entrevistas ou questionários, que são feitos por meio de amostragem, representando a população a ser estudada. Os resultados são analisados e generalizados para a

população da qual se extraiu a amostra. Numa pesquisa de opinião não se possui controle da execução ou das medidas.

As pesquisas de opinião são efetuadas quando se tem por objetivo estar apto a afirmar algo sobre uma população - pesquisa descritiva; quando se pretende sustentar explicações sobre uma população - pesquisa explicativa; ou como pré-estudo para uma investigação mais completa - pesquisa de opinião explorativa.

b. Estudos de caso

Estudos de caso são usados para monitorar projetos, atividades ou tarefas, geralmente objetivando observar um atributo específico ou estabelecer relações entre atributos diferentes.

O investigador coleciona informações detalhadas durante um período contínuo de tempo. Durante a execução de um estudo de caso, pode ser aplicada uma variedade de procedimentos de coleta de dados diferentes.

O nível de controle é menor em um estudo de caso, por ser um estudo observacional, do que em um experimento, que é um estudo controlado.

c. Montagem de experimentos controlados

Experimentos são executados quando um controle da situação é necessário, manipulando a execução do estudo diretamente e sistematicamente.

Em um experimento, selecionam-se objetos, representando uma variedade de características (que serão as variáveis do experimento) e projeta-se a pesquisa para medir mais de um valor para cada característica, os quais serão comparados depois.

Experimentos são apropriados para investigar diferentes aspectos, como: confirmar teorias; testar concepções de pessoas; explorar relações; avaliar a precisão de modelos; validar medidas, etc. Além disso, podem ser efetuados em ambiente universitário antes de se fazer um estudo em indústria, baixando assim o custo e os riscos (Linkman e Rombach (1997) *apud* Wohlin et al. (2000)).

Já diversos fatores que ajudam a decidir entre executar um estudo de caso ou um experimento controlado. O fator central é o nível de controle necessário para um experimento. Quando se tem um nível de controle alto sobre as variáveis, que podem afetar a veracidade das hipóteses, então deve-se considerar a execução de um experimento. Se não

existe esse controle, é preferível a execução de um estudo de caso. Mesmo sendo possível controlar as variáveis, deve-se levar em consideração, na escolha do tipo de estudo empírico a ser utilizado, o nível de dificuldade, custo e risco para fazer isso. Um outro aspecto a ser considerado é a viabilidade de replicar a situação básica que está sendo investigada. Se a replicação não é possível, ou se seu custo é proibitivo, então um experimento controlado pode não deve ser recomendado (DESMET, 1994).

3.3 Processo de Experimentação

Quando há uma relação de causa e efeito que se quer analisar, tem-se uma teoria ou pode-se formular uma hipótese. Para avaliar essa teoria ou hipótese pode-se usar um experimento.

Na execução de experimento controlado, estuda-se o resultado da alteração de algumas variáveis de entrada de um processo. Há dois tipos de variáveis de entrada em um experimento: variáveis independentes e dependentes.

Todas as variáveis em um processo de experimentação, que são manipuladas e controladas, são chamadas **variáveis independentes** (ou variáveis de entrada, ou ainda, variáveis de estado). As variáveis nas quais se observa o efeito das mudanças das variáveis independentes são chamadas **variáveis dependentes** (ou variáveis de resposta, ou ainda, variáveis de saída).

Um experimento estuda o efeito de mudar uma ou mais variáveis independentes. Essas variáveis são chamadas **fatores**. As outras variáveis independentes são fixadas durante o experimento, pois de outra forma não se poderia dizer se foi o fator, ou outra variável, que causou o efeito.

Os **tratamentos** são aplicados à combinação de objetos e participantes (pessoas que aplicam o tratamento). Objeto pode ser, por exemplo, um documento que será revisado com diferentes técnicas de inspeção. Tratamento é um valor particular de um fator, podendo ser métodos, técnicas, ferramentas ou outras condições que está se estudando o efeito.

Um experimento consiste em um conjunto de testes (às vezes chamados tentativas) em que cada teste é uma combinação de tratamento, participante e objeto.

Para executar um experimento, vários passos têm que ser efetuados em uma certa ordem, por isso necessita-se de um processo. O processo de experimentação pode ser dividido nas seguintes atividades (Wohlin et al., 2000):

1. **Definição** - Define-se o experimento em termos de problema, objetivo e metas.
2. **Planejamento** - Determina-se o projeto do experimento e a instrumentação. As possíveis ameaças para validação dos resultados do experimento são avaliadas.
3. **Operação** - Coletam-se medidas.
4. **Análise e interpretação** - As medidas coletadas na fase anterior são analisadas e avaliadas.
5. **Apresentação e empacotamento** - Os resultados são apresentados e as informações do experimento são disponibilizadas para futuras replicações.

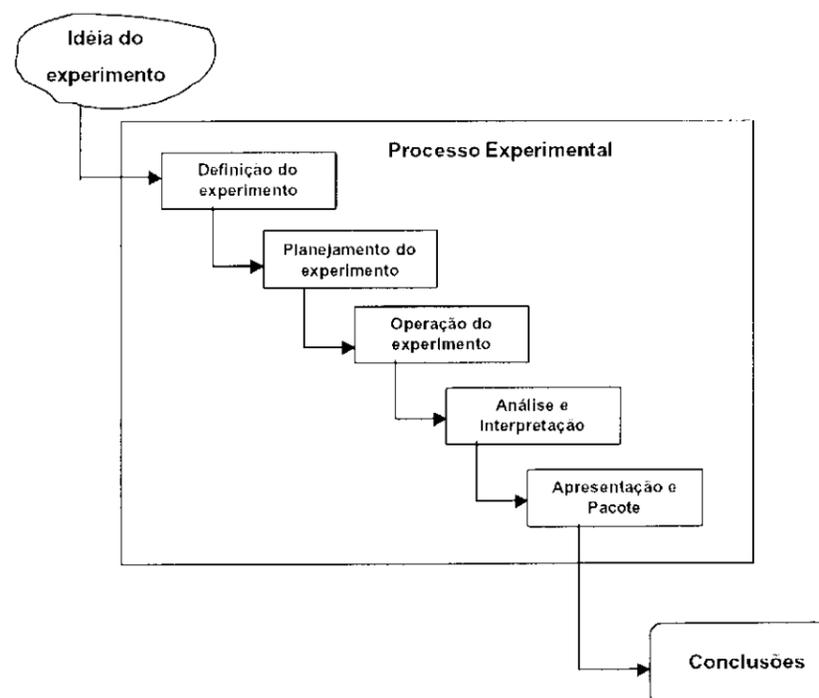


Figura 3.1: Visão Geral do Processo Experimental (Wohlin et al., 2000)

O processo de experimentação é iterativo, podendo ser necessário voltar e refinar uma atividade, antes que seja dada continuidade ao experimento. A ordem das atividades no processo (Figura 3.1) indica, principalmente, a ordem de início das mesmas, não sendo necessariamente obrigatório que uma atividade seja terminada primeiro para que a próxima seja iniciada. A principal exceção é quando a fase de operação do experimento foi iniciada. Nessa fase os participantes já estão influenciados pelo experimento, não sendo mais possível retornar para as fases de definição e planejamento e, quando retomada a fase de operação do processo, utilizar os mesmos participantes.

A seguir, as atividades do processo são comentadas com mais detalhes.

3.3.1 Definição

Para iniciar a primeira fase, a hipótese (possível relacionamento de causa e efeito que se quer analisar) já deve estar estabelecida, definindo-se agora os objetivos e metas. Um modelo de meta deve ser estabelecido. Wohlin et al. (2000) exemplifica:

Analisar <objeto de estudo>

Para o propósito de <propósito>

Com respeito a <enfoque>

Do ponto de vista da <perspectiva>

No contexto <contexto>

O objeto do estudo é a entidade que é analisada. Podem ser produtos, processos, teorias, modelos etc. O propósito consiste na intenção do experimento, como por exemplo, analisar o impacto de duas técnicas diferentes. O enfoque é o efeito estudado, podendo ser custo, eficácia, etc. A perspectiva determina sob qual ponto de vista os resultados serão interpretados. Por fim, o contexto mostra o ambiente no qual o experimento está sendo conduzido, definindo o perfil das pessoas que estão envolvidas e caracterizando os artefatos de software usados.

3.3.2 Planejamento

Enquanto a fase de definição determina **porquê** o experimento é executado, a fase de planejamento diz **como** ele é conduzido.

A fase de planejamento pode ser subdividida em 7 passos:

a) Seleção de contexto

Seleciona o ambiente em que o experimento ocorrerá e o pessoal que participará, usando como base a definição do experimento, feita na fase anterior.

b) Formulação de hipóteses

A hipótese do experimento (ou hipótese alternativa) e a hipótese nula são declaradas. A hipótese nula é a que assume que não há diferença entre os dois tratamentos, com respeito à variável de resposta. A hipótese alternativa apresenta uma posição que há uma diferença significativa entre os dois tratamentos (DESMET, 1994).

c) Seleção de variáveis

Determinam-se as variáveis, tanto as independentes como as dependentes. Determinam-se também a escala de medida e os valores que as variáveis podem receber.

d) Seleção dos participantes

Os participantes que farão parte do experimento são escolhidos e identificados. As características dos participantes devem ser claramente definidas, para que os efeitos ou diferenças entre eles possam ser avaliados em termos dos resultados observados.

e) Projeto do experimento

O projeto do experimento é um plano completo para aplicação de condições experimentais distintas para os participantes, para que se possa determinar como as condições afetam o comportamento ou resultado de alguma atividade (DESMET, 1994).

O experimento é projetado com base nas hipóteses e variáveis selecionadas. O projeto pode ser aleatório, em blocos, balanceado ou alguma combinação desses.

Aleatório: Para garantir validade conclusiva na ocorrência de perturbações não especificadas (Box et al., 1978). Quando se quer impedir que fatores indesejáveis, mas que não se tem conhecimento, sejam obrigatoriamente associados a determinadas combinações e possam influenciar o resultado (Barros Neto et al., 2001).

A aleatoriedade pode ser aplicada na distribuição dos objetos, participantes e na ordem que os testes são executados (Wohlin et al., 2000).

Em blocos: Para eliminar fontes indesejáveis de variabilidade (Box et al., 1978). É empregado quando se tem um fator que provavelmente tem um efeito conhecido e controlável no resultado, mas não se está interessado nesse efeito. É usado para sistematicamente eliminar o efeito indesejado na comparação entre os tratamentos. Em um bloco (ou grupo) o efeito indesejado é o mesmo e pode-se estudar o efeito do tratamento nesse bloco (Wohlin et al., 2000). Então, na hora de definir o planejamento, distribuem-se os fatores de forma a evitar ou minimizar os efeitos indesejáveis (Barros Neto et al., 2001).

Balancedo: Tem-se um projeto balanceado quando os tratamentos são determinados com igual número de participantes. Balanceamento é desejável porque ele tanto simplifica quanto fortalece as análises estatísticas do dado, apesar de não ser imprescindível. (Wohlin et al., 2000)

f) Instrumentação

Este passo prepara para a implementação prática do experimento (objetos, diretrizes e instrumentos de medida).

g) Avaliação de validade

Neste passo a validade do experimento é analisada, levando em consideração os fatores que podem afetá-la.

Riscos à validade são fatores além do controle dos experimentadores, que podem afetar as variáveis dependentes. Assim, riscos podem ser considerados variáveis independentes desconhecidas causando a existência de hipóteses antagônicas não controladas, em adição às hipóteses que estão sendo pesquisadas. Um passo crucial em um experimento é minimizar o impacto desses riscos (Basili et al., 1996).

Se houver uma relação entre o tratamento e o resultado para haver **validade interna**, deve-se certificar que essa é uma relação causal e não resultado de um fator que não tenha sido controlado ou medido. Riscos para validação interna constituem problemas potenciais na interpretação dos dados de um experimento. Se o experimento não tem um mínimo de validação interna, não podem ser feitas inferências válidas com relação ao relacionamento causa-efeito entre as variáveis independentes e dependentes.

A **validade externa** está relacionada com generalização. Se há uma relação causal entre a construção da causa e o efeito, deve-se questionar se o resultado do estudo pode ser generalizado para fora do escopo do estudo.

A **validade de conclusão** refere-se à relação entre o tratamento e o resultado do experimento.

A **validade de construção** refere-se à relação entre teoria e observação. Se a relação entre causa e efeito é causal, tem-se que assegurar que o tratamento reflete a construção da causa e que o resultado reflete a construção do efeito.

3.3.3 Operação

A operação consiste de três passos: **preparação**, **execução** e **validação de dados**. No passo de preparação, o material necessário é preparado, como formulários, software etc., e os participantes são escolhidos. Estes devem ser informados sobre a intenção do experimento e então, deve-se obter o consentimento deles em participar. A execução ocorre quando os participantes executam as tarefas e os dados são coletados. Na validação dos dados, o experimentador deve checar se os dados são razoáveis e se foram coletados corretamente.

3.3.4 Análise e Interpretação

Esta fase é executada utilizando os dados coletados na fase anterior. O primeiro passo na análise é tentar entender os dados usando estatísticas descritivas, que fornecem uma visualização dos mesmos. O segundo passo é reduzir o conjunto de dados para um conjunto válido, ou seja, excluir os dados anormais ou dados falsos. No terceiro passo os dados são analisados pelo teste de hipóteses, no qual as hipóteses do experimento são avaliadas estatisticamente, verificando se é possível rejeitar hipóteses nulas.

3.3.5 Apresentação e Empacotamento

A última atividade do processo de experimentação preocupa-se com a documentação dos resultados, que podem ser feitos através de artigos de pesquisa para publicação, pacotes de laboratório e como parte de uma base de conhecimento.

Uma modificação nesse processo de experimentação foi apresentada por Amaral (2003). Esse trabalho propõe que as atividades de empacotamento sejam realizadas ao longo de todas as fases do processo de experimentação ao invés de ser executada apenas como a última atividade do processo.

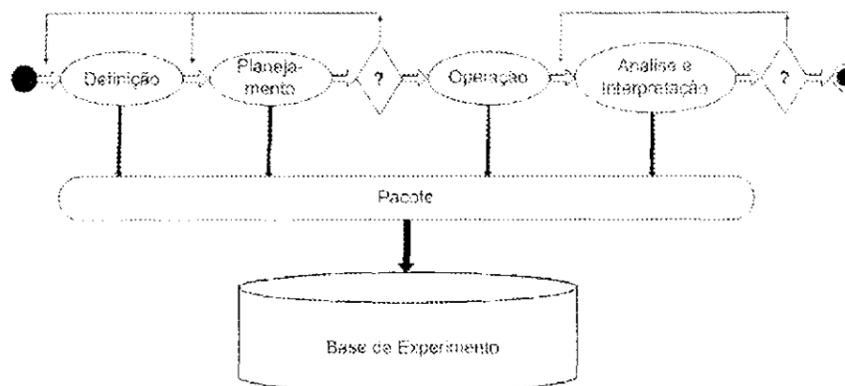


Figura 3.2: Processos de Experimentação e Empacotamento (Amaral, 2003)

Pela Figura 3.2 observa-se que o processo de empacotamento acompanha todas as fases do processo de experimentação, armazenando em um repositório informações de todas as atividades.

3.4 Replicação de Experimentos

“Construir um corpo de conhecimento em engenharia de software requer famílias de experimentos e um conjunto de princípios unificados, que permitem que os resultados sejam combinados e generalizados” (Basili et al., 1999).

Replicações são necessárias para construir esse corpo de conhecimento sobre os resultados, confirmando-os sob outras condições ou mesmo formulando outros questionamentos.

Replicação é importante por diversas razões. Primeiro, replicação fornece uma estimativa de erro experimental que atua como uma base para avaliação das diferenças observadas em uma variável independente. Ou seja, replicação pode ajudar a conhecer o grau de segurança dos resultados do experimento. Segundo, replicação permite estimar o efeito principal de qualquer fator experimental (DESMET, 1994).

Existem vários objetivos para replicações, segundo Basili et al. (1999):

1. **Replicações em que nenhuma hipótese de pesquisa varia**

Não variam as variáveis dependentes ou independentes do experimento original. Podem ser:

- **Replicações estritas:** repetem o experimento original tão precisamente quanto possível. São necessárias para aumentar a confiança na validade da conclusão do experimento.
- **Replicações que variam a execução:** Variam a maneira na qual o experimento é executado. Aumentam a confiança em resultados experimentais testando as mesmas hipóteses de experimentos anteriores, mas alteram detalhes para direcionar determinadas ameaças internas para validade.

2. **Replicações que variam as hipóteses de pesquisa**

Variam atributos do processo, produto e modelos de contexto, mas permanecem no mesmo nível de detalhamento que o experimento original. Podem ser:

- **Replicações que variam as variáveis independentes:** investigam que aspectos do processo são importantes como propriedades intrínsecas, variando o processo e examinando os resultados.
- **Replicações que variam as variáveis dependentes:** podem variar os modos nos quais a efetividade está sendo medida, para entender em que dimensões de uma tarefa, um processo trará melhores resultados.
- **Replicações que variam as variáveis de contexto:** variação de contexto no ambiente no qual a solução é avaliada. Podem identificar fatores ambientais importantes que afetam os resultados do processo sob investigação e assim, ajudar a entender sua validade externa.

3. **Replicações que estendem a teoria**

Ajudam a determinar os limites de efetividade de um processo, fazendo grandes mudanças no processo, produto e/ou modelo de contexto, para verificar se princípios básicos ainda permanecem.

Na seção seguinte exemplifica-se um processo de experimento controlado, que objetiva avaliar a efetividade da técnica de leitura *Perspective-Based Reading* (PBR) na detecção

de defeitos, quando aplicada em documentos de especificação de requisitos, comparando-a com técnicas convencionais de revisão. A técnica PBR foi apresentada no Capítulo 2, na Seção 2.1.2.

3.5 Experimento Utilizando a Técnica PBR em Documentos de Especificação Informal de Requisitos

Nesta seção apresenta-se o experimento que foi base para os experimentos conduzidos no projeto *Readers* para avaliação da técnica PBR. Esse experimento será apresentado seguindo as atividades do processo de experimentação apresentado na Seção 3.3.

Um experimento piloto foi conduzido pela equipe do professor Victor R. Basili (Universidade de Maryland) no Laboratório de Engenharia de Software (SEL - *Software Engineering Laboratory*) da NASA/GSFC (*National Aeronautics and Space Administration/Goddard Space Flight Center*) em novembro de 1994 (Basili et al., 1996). Esse estudo objetivava avaliar a efetividade da técnica de leitura PBR (*Perspective Based Reading*) na detecção de defeitos em documentos de especificação de requisitos, quando comparada com a técnica desenvolvida na NASA SEL. Os participantes desse experimento eram profissionais de desenvolvimento de software que leram um conjunto de documentos de especificação de requisitos, nos quais foram introduzidos defeitos comuns ao ambiente.

1. Definição

Analisar a *técnica PBR*

Para o propósito de *avaliação*

Com respeito a *efetividade*

Do ponto de vista de *revisores de software*

No contexto de *profissionais de desenvolvimento*.

2. Planejamento

- a) **Seleção de contexto:** profissionais desenvolvedores de software do Laboratório de Engenharia de Software (SEL - *Software Engineering Laboratory*) da NASA/GSFC (*National Aeronautics and Space Administration/Goddard Space Flight Center*)

b) Formulação de hipóteses:

H0: a união dos defeitos detectados pelos grupos de participantes que utilizaram perspectivas diferentes, fornece uma maior cobertura do documento do que a união dos defeitos detectados pelos grupos usando a técnica da NASA.

Para estudar a hipótese acima, o experimento destinou-se a responder as seguintes questões:

1. Se fossem dadas unicamente as regras PBR aos grupos de revisores (tal como durante uma sessão de inspeção), seria detectado um maior número de defeitos do que se cada um lesse o documento de modo similar?
2. Se os revisores lessem um documento usando PBR, seria encontrado um número diferente de defeitos do que aqueles detectados caso utilizassem técnicas convencionais?
3. A experiência do revisor na perspectiva (projetista, testador e usuário) influencia o desempenho quando utiliza PBR?

c) Seleção de variáveis: Variáveis independentes: técnicas de leitura (PBR e a pertencente a NASA SEL), perspectivas (projetista, testador e usuário) e documentos de especificação de requisitos (dois de domínio de aplicação da NASA e dois de domínio de aplicação genérico). Variáveis dependentes: conhecimento do participante na língua inglesa e experiência prática nas perspectivas (projetista, testador e usuário).

d) Seleção de participantes: Os participantes foram voluntários, profissionais de desenvolvimento.

e) Restrições e Limitações: Tempo e custo, pois os participantes eram profissionais, sem grande disponibilidade. A quantidade de participantes também teve que ser limitada.

f) Projeto do experimento: As tarefas executadas pelos participantes consistiam em leitura e revisão de documentos de especificação de requisitos e registro dos defeitos identificados em um formulário. O tratamento, que teve o propósito de manipular uma ou mais das variáveis independentes, consistiu em treinar os participantes nas técnicas PBR. A ordem de tarefas e tratamentos para cada grupo de participantes era: fazer uma tarefa com a técnica habitual, então treinar na PBR, seguido pela segunda tarefa utilizando PBR. Todos os documentos

revisados por um participante deviam ser diferentes, para que os resultados não fossem influenciados pelo conhecimento adquirido em leituras prévias. Devido a isso, eles foram separados em dois grupos.

O projeto do experimento foi com tratamento em bloco na técnica, na perspectiva, no documento e na seqüência de leitura, para poder adquirir uma distribuição igual dos valores das variáveis independentes diferentes. Assim ficaram dois grupos de participantes (cada um com 6 participantes), em que cada grupo continha três subgrupos, um para cada perspectiva (Tabela 3.1).

Tabela 3.1: Projeto Experimental (Basili et al., 1996)

Técnica	Grupo 1			Grupo 2			
	Projetista	Testador	Usuário	Projetista	Testador	Usuário	
Usual	Treinamento			Treinamento			Primeiro Dia
	NASA_A			NASA_B			
	Treinamento			Treinamento			
	ATM			PG			
PBR	Teoria PBR						Segundo Dia
	Treinamento			Treinamento			
	PG			ATM			
	Treinamento			Treinamento			
	NASA_B			NASA_A			

g) **Instrumentação:** Documento de requisitos genérico para treinamento: *Video Rental System* (14 páginas, 16 defeitos semeados); Documentos de requisitos genéricos para aplicação: *Automatic Teller Machine - ATM* (17 páginas, 29 defeitos semeados), *Parking Garage - PG* (16 páginas, 27 defeitos semeados). Documento de requisitos do domínio da NASA para treinamento: *NASA sample* (9 páginas, 6 defeitos semeados). Documentos de requisitos do domínio da NASA para aplicação: *Flight dynamics* (NASA_A e NASA_B - 27 páginas cada, 15 defeitos semeados em cada um).

h) **Avaliação de validade:**

I. Análise dos riscos para validade interna

- **História:** Desde que haja um dia de intervalo entre os dois dias do experimento, algumas das melhoras, que parecem ocorrer devido à técnica, poderiam ser atribuídas a outros eventos que podem acontecer entre os

testes. Os participantes foram instruídos a não discutir o experimento e a não fazer nada entre os testes que pudesse causar um efeito não esperado nos resultados. Devido a isso, acredita-se que os programadores profissionais de um grupo não discutiram suas especificações com os do outro grupo. Assim, pode-se considerar esse efeito sem muita significância, mas não pode-se ignorá-lo completamente.

- **Maturação:** Esse efeito ocorre nos participantes em função do tempo, como cansaço ou tédio. Mas também pode ser maturação intelectual, descuidando dos eventos experimentais. Nesse experimento, esse efeito poderia ser devido à aplicação dos testes no final do dia, tendendo a conseguir resultados piores do que se eles fossem executados no horário habitual de trabalho. Foram dadas longas paradas entre os testes para tentar evitar essa tendência. Entretanto, já que a ordenação dos documentos e domínios foi diferente para os dois dias, as diferenças entre os dois dias podem ter sido influenciadas pelos efeitos de maturação. Por meio do projeto experimental nota-se que uma melhora do primeiro para o segundo dia poderia ser ampliada para os documentos genéricos e diminuída para os documentos específicos. Baseando-se nos resultados do experimento, pode-se ver que esse efeito parece plausível. Por outro lado, escolhesse-se a mesma ordem de domínios e documentos nos dois dias, o risco poderia ser pior porque uma melhora de resultados com o uso da técnica PBR em relação ao uso da técnica usual poderia ser completamente confundida com o efeito de maturação.
- **Teste:** Ganhar familiaridade com os testes pode ter efeitos nos resultados subsequentes. Esse risco tem diversos componentes, incluindo tornar-se familiar com as especificações, a técnica ou os procedimentos de teste. Esse efeito pode aumentar os efeitos dos eventos históricos. Efeitos de teste podem anular efeitos de maturação em cada dia. Apesar dos participantes já estarem familiarizados com documentos NASA, tentou-se submeter efeitos não esperados por meio de sessões de treinamento antes de cada teste, nas quais os participantes pudessem familiarizar-se com o tipo particular de documento e técnica. Além disso, os participantes não receberam *feedback* com respeito ao seu

sucesso da detecção de defeitos durante o experimento, assim torna-se difícil para eles descobrirem se aspectos do seu desempenho de fato melhoraram a razão de detecção, ou não. Fora isso, os documentos genéricos são suficientemente diferentes, que torna mínimo o que pode ser aprendido no primeiro documento que possa ser transferido para o segundo.

- **Instrumentação:** Esses efeitos são basicamente devido a diferenças no modo de medir e avaliar a marcação dos defeitos. Foi avaliado, independentemente, por duas pessoas que não sabiam que tratamento eles estavam classificando e então, discutido para resolver alguma diferença consistente.
- **Seleção:** Foi usada definição aleatória de participantes para as perspectivas. Sendo que a PBR assume que revisores na equipe usem as perspectivas com as quais mais eles estão familiarizados, a definição aleatória usada no experimento deixaria, presumidamente, uma subestimação da melhora causada pela PBR.

Outro risco para a validação interna é a possibilidade dos participantes ignorarem a PBR, quando eles estão supostamente usando-a. Há um perigo que os participantes continuem a usar suas técnicas habituais. Isso não precisa ser uma escolha consciente do participante, mas pode simplesmente refletir o fato que pessoas aplicam suas habilidades inconscientemente. O único meio de lidar com esse risco é fornecer sessões de treinamento maiores e algum tipo de controle ou medida de conformidade para a técnica determinada.

II. Análise dos riscos para validade externa

Riscos para validade externa implicam limitações para generalizar os resultados. Esse experimento foi conduzido com desenvolvedores profissionais e com documentos de um contexto industrial, assim esses fatores apresentaram pouco risco para validação externa. Entretanto, o número limitado de pontos de dado é um problema potencial que pode apenas ser superado por replicações do experimento.

Outros riscos para validação externa pertinente ao projeto experimental incluem:

- **Interação de teste e tratamento:** Um pré-teste pode afetar a sensibilidade do participante em relação à variável experimental. Os dois grupos receberam pré-testes e tratamentos similares, assim, esse efeito pode ser preocupante. Não pode ser evitado o fato de que esse é um ambiente experimental e todos os participantes sabem disso. Isto pode afetar os resultados e é uma limitação de qualquer projeto experimental.
- **Interação de seleção e tratamento:** Propensão da seleção pode ter efeitos diferentes devido à interação com o tratamento. O fato de que todos os participantes tenham sido voluntários pode implicar que eles são mais inclinados a esforços orientados a aperfeiçoamento que o desenvolvedor médio - ou pode indicar que eles consideram o experimento uma oportunidade para ter atividades fora do trabalho normal, por dois dias. Assim, os efeitos podem ir nessa direção. Além disso, todos os participantes receberam treinamento na técnica da instituição, uma propriedade que desenvolvedores de outras organizações podem não ter.
- **Disposições Reagentes:** Esses efeitos ocorrem devido ao ambiente experimental. Esse projeto piloto foi realizado no ambiente próprio dos participantes e assim, sua validade poderia ser ampliada para os ambientes análogos. Em um ambiente acadêmico não se pode assumir o mesmo (experimento aplicado em 1995). Entretanto, a mudança do ambiente experimental, ocorrida entre os experimentos aplicados, tem tornado mais fácil se concentrar nas técnicas e testes a serem feitos, separando melhor as técnicas.

3. Operação

O experimento foi conduzido em novembro de 1991, com 12 participantes, durante 2 dias. Em uma segunda-feira cada participante usou a técnica usual para revisar um documento NASA e um genérico e, na quarta-feira, cada participante assumiu uma das três perspectivas da PBR e aplicou nos outros dois documentos nos quais ainda não tinha trabalhado.

Durante a realização desse experimento foram identificados os seguintes problemas potenciais:

- (a) A tentativa de distribuir os participantes de acordo com suas experiências e as perspectivas não foi possível, porque havia mais projetistas do que testadores e usuários. Foi decidido que em todas as replicações posteriores a distribuição seria aleatória.
- (b) Os documentos de domínio específico da NASA eram muito longos para a análise no tempo determinado. Foi decidido por uma revisão e redução dos documentos para futuras replicações do experimento.
- (c) As sessões de revisão eram de 3 horas para cada documento. Como apenas um participante levou mais do que duas horas, foi decidido reduzir o tempo para duas horas nas replicações posteriores.
- (d) As revisões poderiam ser feitas nas próprias salas dos participantes, muitas vezes sofrendo interrupções diversas. Decidiram que outras ocorrências do experimento seriam executadas em ambiente único e reservado, para garantir maior validade dos resultados.
- (e) Foram encontradas algumas inconsistências e ambigüidades na descrição geral do problema nos documentos de especificação de requisitos. Decidiu-se revisá-los para que as sessões de descrição geral das especificações estivessem corretas. Assim, os participantes teriam alguma base para tomar decisões.

Devido a essas mudanças, esse experimento foi chamado de estudo piloto para o projeto experimental. Foi conduzido um segundo experimento em junho de 1995 com 14 participantes. Como um desses participantes não era familiarizado com aplicações de dinâmicas de voo da NASA, foram considerados apenas os outros 13 participantes na análise dos documentos NASA_A e NASA_B.

4. Análise e interpretação

Como as técnicas PBR são específicas e nenhuma delas fornece cobertura do documento inteiro, então assume-se que a combinação delas, em grupos, irá oferecer cobertura completa do documento. Na Figura 3.3 mostra-se a análise por grupos, na qual a técnica PBR obteve resultados significativos apenas nos documentos de domínios genéricos, na primeira execução do experimento. Na segunda execução, os

grupos que utilizaram a PBR obtiveram melhora tanto nos documentos de domínio genérico quanto nos específicos. As razões para isso podem ser atribuídas ao problema encontrado nos documentos de domínio específico, relatado acima, no item b, os quais foram reformulados para a segunda execução, e também ao fato das sessões de treinamento terem sido efetuadas antes de cada revisão de documento.

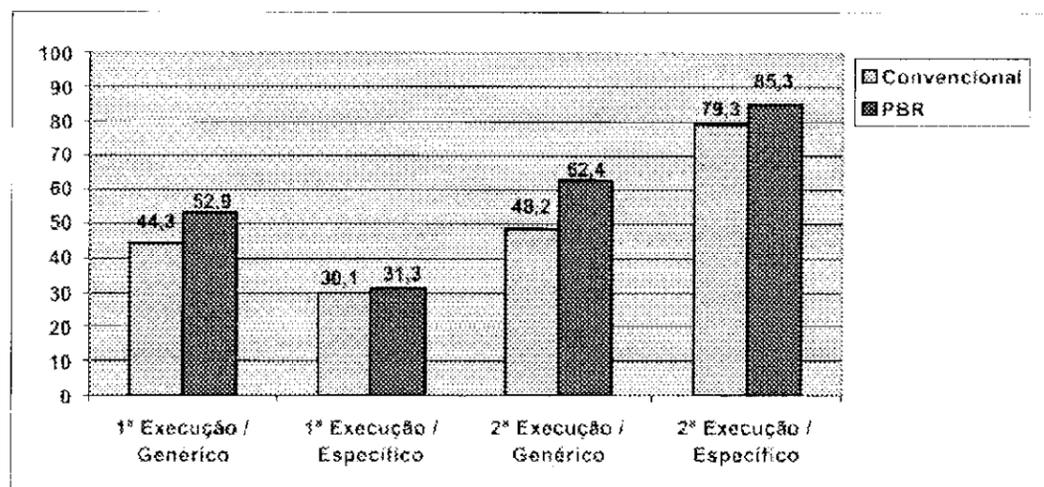


Figura 3.3: Resultados Obtidos Pelos Grupos (Basili et al., 1996)

Quanto ao desempenho individual, foi feita a análise observando-se o percentual de defeitos detectados por participante em relação ao número total de defeitos existentes no documento inspecionado (Figura 3.4). Apesar do objetivo principal ser a investigação dos efeitos da técnica PBR em grupos, avaliou-se também o desempenho individual, comparando o uso da PBR com o da técnica usual. Observou-se que os participantes que utilizaram PBR nos documentos genéricos, durante a segunda execução, tiveram um desempenho significativamente melhor, o que mostra que o uso de técnicas sistemáticas com responsabilidades específicas pode melhorar os percentuais individuais, além dos do grupo.

Outra análise feita visou a determinar se participantes utilizando PBR descobriram classes mais amplas de defeitos e se houve sobreposição em relação às perspectivas e aos defeitos descobertos por elas. Na Figura 3.5 mostra-se essa análise para o documento genérico ATM e na Figura 3.6 para o documento específico NASA_A.

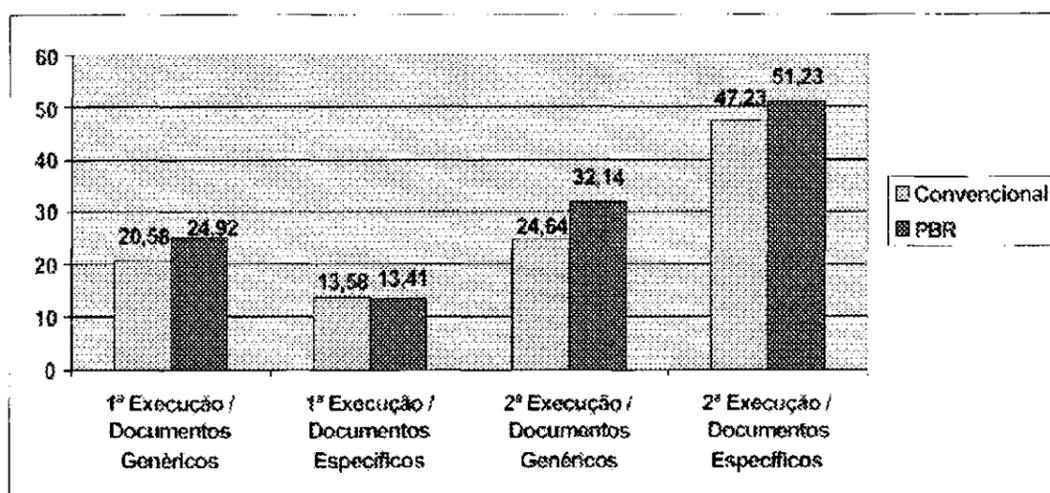


Figura 3.4: Resultados Obtidos Pelos Participantes (Basili et al., 1996)

Em documentos genéricos, foi observado que houve um número maior de defeitos descoberto apenas por uma das perspectivas. Enquanto que nos documentos específicos houve maior sobreposição entre as perspectivas.

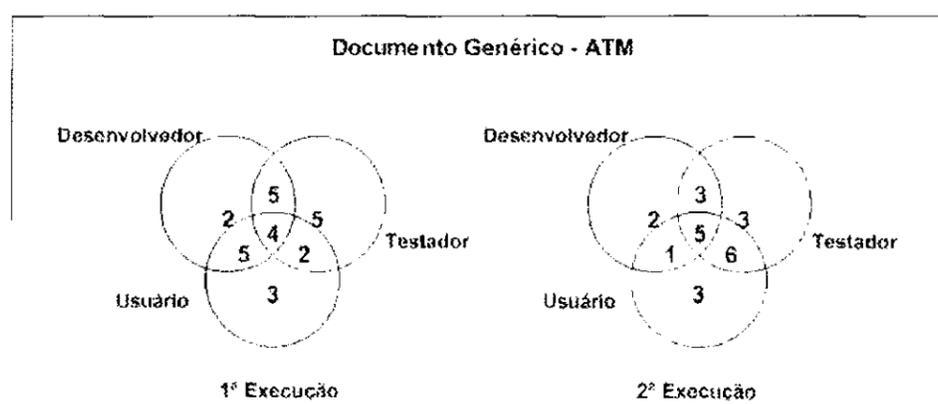


Figura 3.5: Cobertura Obtida pela PBR no Documento Genérico ATM (Basili et al., 1996)

Em documentos genéricos, foi observado que houve um número maior de defeitos descoberto apenas por uma das perspectivas. Enquanto que nos documentos específicos houve maior sobreposição entre as perspectivas.

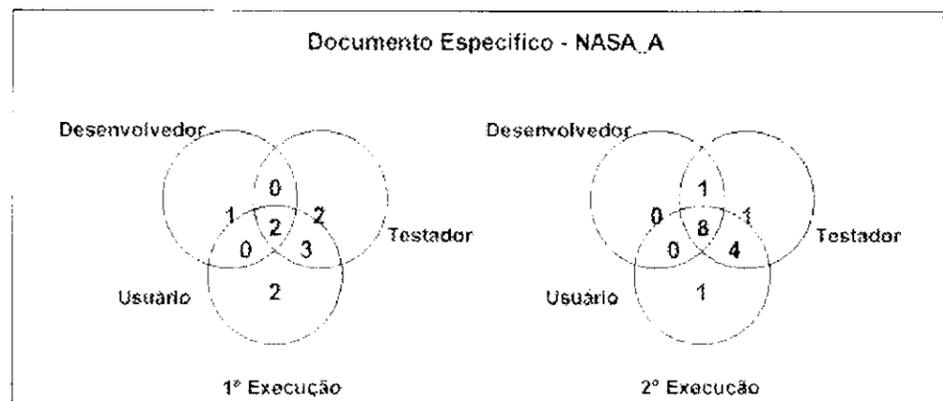


Figura 3.6: Cobertura Obtida pela PBR no Documento Especifico NASA_A (Basili et al., 1996)

5. Apresentação e empacotamento

Os resultados dos experimentos (estudo piloto e execução de 1995) foram publicados em artigo (Basili et al., 1996), constando os riscos à validação encontrados (riscos à validação interna e externa apresentados na seção 3.5) para que outros pesquisadores possam evitá-los quando da replicação desse experimento ou desenvolvimento de outros.

Basili et al. (1996) ressaltam a necessidade de replicar esse experimento em outros ambientes, inclusive em outros países com língua e cultura diferentes, para verificar os efeitos causados por essas diferenças e para obter mais dados quantitativos, para que seja possível entender como controlar adaptações da técnica, descobrindo também características diferentes entre as versões da técnica PBR. Com esse intuito, foi criado um pacote de laboratório com todo o material necessário para dar suporte à replicação por outros pesquisadores. O pacote de laboratório é composto por (Basili et al., 1998):

- Procedimentos para a pessoa que irá conduzir a replicação;
- Formulários iniciais: formulário de concordância (*consent form*), lista de participantes e formulário de pesquisa (*analyst survey*);
- Material de treinamento;
- Material de orientação para os participantes;

- Documentos de requisitos;
- Listas de defeitos dos documentos de requisitos;
- Questionário de opiniões (*Feedback Questionnaire*) e Formulários de Entrevista;
- Procedimentos de análise de dados.

Para Shull et al. (2004), um pacote de laboratório compõe uma infra-estrutura experimental para dar suporte a futuras replicações descrevendo um experimento em termos específicos, fornecendo material para replicação, destacando oportunidades para variação e estabelecendo um contexto para combinação de resultados de diferentes tipos de tratamentos experimentais. Pacotes de laboratório estabelecem uma base para confirmar ou rejeitar os resultados originais, complementando o experimento original e adaptando o objeto de estudo para um contexto experimental específico.

3.6 Replicações do Experimento

O experimento descrito na seção anterior foi replicado na Universidade de Kaiserslautern (Alemanha) utilizando um projeto diferente visando a estudar diretamente os efeitos da PBR nas equipes de revisores (Ciolkowski et al. (1997) *apud* Basili et al. (1999)). Outra replicação foi executada na Universidade de Trondheim (Noruega) utilizando um projeto bem similar ao estudo original, mas com alterações nas técnicas PBR com o objetivo de estudar conformidade do processo (Sorungard (1996) *apud* Basili et al. (1999)). As idéias resultantes desse experimento influenciaram no re-planejamento das técnicas PBR, efetuadas pela Universidade de Maryland, com o objetivo de tratar da conformidade do processo (Shull, 1998). Essa versão das técnicas PBR está sendo aplicada em replicações na Universidade de Bari (Itália), Universidade Drexel (EUA), Universidade de São Paulo (Brasil) e Universidade Lund, Suécia (Basili et al., 1999).

A Universidade de São Paulo (USP), juntamente com a Universidade Federal de São Carlos (UFSCar), vem conduzindo replicações desse experimento através do projeto *NSF-CNPq Readers*. Esse projeto de pesquisa colaborativa visa a desenvolver, validar e empacotar técnicas de leitura para detecção de defeitos de software, tendo também como parceiros a Universidade de Maryland, a COPPE - Universidade Federal do Rio de Janeiro e a Universidade Salvador (UNIFACS).

A seguir comentam-se as duas primeiras replicações desse experimento no contexto do projeto *Readers*.

3.6.1 Replicação 1 (R1)

A primeira replicação do experimento controlado para estudar a técnica PBR, dentro do projeto *Readers*, foi conduzida pela USP/UFSCar, em dezembro de 2000. Foi aplicada com alunos de graduação do Instituto de Ciências da Computação e Estatística da Universidade de São Paulo (ICMC-USP). O objetivo principal do estudo era comparar o uso das técnicas *Checklist* e *Perspective-Based Reading* (PBR) para detecção de defeitos em requisitos de software (Dória, 2001; Maldonado et al., 2004a).

Projeto do experimento:

Os participantes foram 18 estudantes de graduação da disciplina de Engenharia de Software do Curso de Bacharelado em Ciência da Computação do ICMC-USP. Foram utilizados dois documentos de requisitos de domínio genérico: ATM (*Automatic Teller Machine*) e PG (*Parking Garage*). Os participantes foram divididos em dois grupos (Grupo 1 e Grupo 2). Cada grupo foi dividido em 3 subgrupos contendo 3 participantes cada. Cada subgrupo foi relacionado a uma das perspectivas (projeto, teste e uso) da técnica PBR. No primeiro dia, os participantes dos Grupos 1 e 2 aplicaram a técnica *Checklist* para revisar os documentos de requisitos (ATM e PG), respectivamente. No segundo dia, cada participante foi treinado em uma das perspectivas (projeto, teste e uso) da técnica PBR para então iniciar o processo de revisão dos documentos de domínio genérico PG e ATM. Esse projeto experimental garante que cada participante aplique cada técnica somente uma vez e que todas as combinações de documentos de especificação de requisitos e técnicas ocorram. Antes de aplicar as técnicas nesses documentos, os participantes foram treinados usando o documento de requisitos ABC Vídeo (Tabela 3.2).

Basicamente o experimento consistiu dos seguintes passos:

1. Preencher o Formulário de Concordância - *Consent Form* e o Formulário de Levantamento do Perfil - *Analyst Survey Form*;
2. Treinar os participantes nas técnicas;
3. Aplicar as técnicas;

Tabela 3.2: Projeto Experimental R1

	Grupo 1 - 9 participantes			Grupo 2 - 9 participantes			Técnica
Primeiro Dia	Treinamento (ABC Vídeo)			Treinamento (ABC Vídeo)			<i>Checklist</i>
	ATM			PG			
Segundo Dia	Projetista	Testador	Usuário	Projetista	Testador	Usuário	PBR
	3 parti- cipantes	3 parti- cipantes	3 parti- cipantes	3 parti- cipantes	3 parti- cipantes	3 parti- cipantes	
	Teoria PBR						
	Treinamento (ABC Vídeo)			Treinamento (ABC Vídeo)			
	PG			ATM			

4. Coletar dados e analisar os resultados; e
5. Preencher o questionário de opinião - *Feedback Questionnaire*.

Os principais resultados obtidos nessa replicação foram (Maldonado et al., 2004b; Shull et al., 2002):

- considerando ambos documentos, revisores PBR foram mais eficientes que os revisores utilizando a técnica *Checklist*;
- revisores utilizando a técnica PBR foram mais efetivos para o documento ATM e tão efetivos quanto os revisores que aplicaram a técnica *Checklist*, para o documento PG;
- cada perspectiva PBR achou um grupo de defeitos que a outra perspectiva não encontrou, o que enfatiza o aspecto complementar delas; e
- quanto ao aspecto de uniformidade de resultados obtidos pelos revisores de cada técnica, nem a técnica *Checklist*, nem a PBR conduziu para a uniformidade completa de defeitos registrados, mas PBR teve uma maior percentagem de participantes que obtiveram o mesmo alto desempenho (dentro de cada perspectiva).

Uma das maiores contribuições desta replicação foi a criação de uma lista de falsos positivos frequentes. Falsos Positivos são os defeitos reportados pelos revisores, mas que não são defeitos reais. Essa lista ajudará os pesquisadores na análise dos dados e a reconhecer novos defeitos, contribuindo para estender a lista de defeitos reais (Maldonado et al., 2004a).

Um outro ponto importante observado nessa replicação foi que os participantes acharam sete novos defeitos para o documento ATM e dois novos defeitos para o documento PG, em adição aos já documentados nos experimentos anteriores (Maldonado et al., 2004a).

3.6.2 Replicação 2 (R2)

A segunda replicação foi executada em maio de 2001, com alunos de graduação do curso de Ciência da Computação da Universidade Federal de São Carlos. Essa replicação foi similar à primeira: 18 estudantes de graduação da disciplina de Engenharia de Software do Curso de Bacharelado em Ciência da Computação da UFSCar, e consistiu dos seguintes passos:

1. Preencher o Formulário de Concordância - *Consent Form* e o Formulário de Levantamento do Perfil - *Analyst Survey Form*;
2. O experimento seguiu o projeto experimental de R1, sendo dividido em quatro dias com meio período e não dois dias inteiros. No primeiro período, todos os participantes receberam treinamento em técnicas de inspeção e treinaram na técnica *Checklist*. No segundo período, os participantes do Grupo 1 revisaram o documento de requisitos ATM e os do Grupo 2 revisaram o documento de requisitos PG. No terceiro período, cada participante foi treinado em uma das três perspectivas PBR, de acordo com o grupo em que foi designado e, no quarto período, o Grupo 1 revisou PG e o Grupo 2 ATM.
3. Coletar dados e analisar os resultados;
4. Preencher o questionário de opinião - *Feedback Questionnaire*.

Os dados coletados nessas replicações foram analisados separadamente e os resultados estatísticos comparados (Mendonça et al., 2004).

Comparando os resultados de R1 com os resultados obtidos em R2 temos (Maldonado et al., 2004b):

- Em R2, a técnica PBR foi mais efetiva do que a técnica *Checklist*, enquanto que em na R1 isso ocorreu apenas para o documento ATM;

- Em termos de eficiência, a técnica PBR obteve melhores resultados do que a técnica *Checklist* para ambos os documentos, tanto em R1 quanto em R2.

3.7 Iniciativas de Automação do Processo de Experimentação

A condução de um experimento exige o registro, a análise e a interpretação de um grande volume de dados, além de um trabalho minucioso na definição e preparação do mesmo. Para se avaliar e validar uma técnica ou processo, por meio de experimentação, é necessário que se tenham dados suficientes para fornecer o grau de credibilidade necessário para generalizar a conclusão, conhecer os custos e benefícios e consolidar uma base de conhecimento. Isso leva à necessidade de replicações do estudo feito, em condições, domínios e até em cultura, diferentes (Maldonado et al., 2001?a).

O suporte ao processo de experimentação com um ambiente automatizado, facilita a coleta, controle e análise dos dados, viabilizando assim maior número de replicações e maior qualidade e, conseqüentemente, confiabilidade nos dados obtidos. A automatização sistematiza as atividades da experimentação facilitando sua aplicação individual e independente, o que viabiliza e estimula a utilização na indústria.

Em cada fase do experimento, algumas tarefas podem ser automatizadas ou apoiadas por ferramentas, trazendo mais rigor e confiança aos resultados. Podemos exemplificar as seguintes:

Planejamento

- Cadastro dos indivíduos;
- Seleção dos indivíduos para os grupos;
- Identificação e distribuição do material.

Operação

- Manuseio do material;
- Preenchimento de formulários;

- Registro e classificação dos defeitos (localização no documento, classificação na taxonomia, justificativa)
- Monitoramento do tempo total;
- Registro do tempo de detecção dos defeitos;
- Percentual de cobertura do documento.

Análise e Interpretação

- Levantamento dos defeitos registrados pelos indivíduos comparando com a Lista de Defeitos;
- Cálculo da Eficácia e Eficiência de cada indivíduo e dos grupos com cada técnica de leitura.

Algumas iniciativas nesse sentido estão sendo realizadas no âmbito do projeto *Readers*. A ferramenta COTEST (*Code Testing Experiment Support Tool*) foi desenvolvida com o objetivo de apoiar a replicação de experimentos baseados em código fonte (Felizardo, 2003). Estão em desenvolvimento uma ferramenta de apoio à aplicação das Técnicas de Leitura Baseadas em Perspectiva (PBR) (Silva, 2003), uma ferramenta de construção de casos de uso para utilização na técnica de leitura PBR e uma infra-estrutura de suporte ao processo de inspeção de software (ISPIS) (COPPE/UFRJ, 2003). Para integração dessas e de outras ferramentas está em estudo uma arquitetura utilizando XML (Spínola, 2003). Também está em desenvolvimento uma ferramenta propondo apoio automatizado à configuração e aplicação de técnicas de leitura OORTs (Reis, 2003). Além disso, um sistema de apoio ao processo de experimentação está sendo estudado, possibilitando interação com outras ferramentas em diversas fases desse processo (Maltez Jr., 2003).

Podemos citar ainda outros trabalhos visando à automatização do processo de experimentação ou partes dele: Arisholm et al. (2002) que apresentam um ambiente de suporte a experimentos nomeado SESE (*Simula Experiment Support Environments*) e Torii et al. (1999) apresentam o Ginger2, um ambiente como uma instância do framework CAESE (*Computer-Aided Empirical Software Engineering*). O CAESE foi proposto para fornecer suporte à engenharia de software empírica do mesmo modo que um ambiente CASE fornece suporte a ciclo de vida de desenvolvimento de software (Torii et al., 1999).

Com a automatização viabilizando um maior número de replicações, gerando assim uma maior e mais confiável quantidade de dados, surge o interesse por meta-análise entre os experimentos. Uma outra linha de investigação no escopo do projeto *Readers* é a aplicação de técnicas e ferramentas de *Visual Data Mining* para a análise e visualização dos dados obtidos, procurando identificar se outras perspectivas de análises poderiam ser exploradas nesse domínio. Um trabalho está sendo desenvolvido propondo-se a investigar como a visualização de dados pode apoiar tanto a análise dos dados coletados nas replicações de experimentos do projeto *Readers*, quanto o próprio processo de VV&T, pelo uso de técnicas de Visualização de Informação e de Visualização de Software, respectivamente. Um dos objetivos é verificar se o uso de técnicas de Visualização traz benefícios ao processo de experimentação, não só na análise dos dados como também na melhoria do processo de aquisição de conhecimento como um todo (Garcia, 2003).

A principal razão da utilização de estudos empíricos quantitativos, como experimentos, na engenharia de software é a possibilidade de obter resultados objetivos e estatisticamente significativos com respeito ao entendimento, controle, prognóstico e melhoramento do desenvolvimento de software (Wohlin et al., 2000). Para se obter sucesso no desenvolvimento de software busca-se a melhoria contínua dos processos utilizados, por meio de estudos empíricos, conduzindo-se experimentos e repetindo-os. Replicações de experimentos são essenciais para obter resultados comparativos ao experimento original, além de gerar e facilitar pesquisas experimentais colaborativas, transferir experiência na execução de experimentos e replicações, obter volume de dados para meta-análise e melhorar os artefatos do pacote experimental (Maldonado et al., 2004?a; Shull et al., 2002). O Paradigma de Melhoria da Experimentação (EIP - Experimentation Improvement Paradigm), baseado no Paradigma de Melhoria da Qualidade (QIP - Quality Improvement Paradigm), visa a gerenciar o conhecimento e melhorar replicações de experimentos controlados em engenharia de software. Nas Seções a seguir apresentam-se o modelo de conversão do conhecimento no contexto da Engenharia de Software (EKSM - *Experimentation Knowledge Sharing Model*), o QIP e o EIP.

3.8 Compartilhamento do Conhecimento e Melhoria do Processo de Experimentação

3.8.1 Compartilhamento do Conhecimento

O conhecimento pode ser diferenciado em conhecimento tácito e conhecimento explícito (Nonaka e Takeuchi, 1997). O conhecimento tácito é pessoal, específico ao contexto e está profundamente enraizado nas ações e experiências de um indivíduo, bem como em suas emoções, valores ou ideais, portanto é difícil de formalizar, o que dificulta sua transmissão e compartilhamento. Por outro lado, o conhecimento explícito pode ser formalizado e sistematizado, se tornando mais facilmente transmissível. Nonaka e Takeuchi (1997) defende que o conhecimento é criado por meio da interação entre o conhecimento tácito e o conhecimento explícito. Então para a criação do conhecimento organizacional necessita-se de uma interação contínua e dinâmica entre esses conhecimentos por meio de diferentes modos de transformação do conhecimento, ilustrada com a espiral do conhecimento na Figura 3.7. Nessa figura, adaptada por Shull et al. (2004), *I* representa um indivíduo, *G* representa um grupo e *O* uma organização.

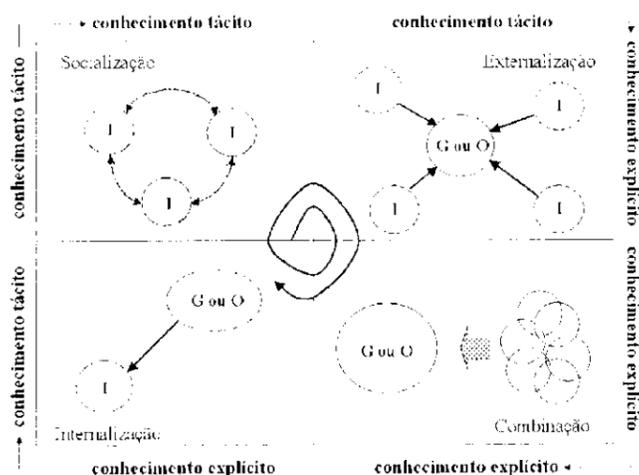


Figura 3.7: Modelo de Compartilhamento do Conhecimento (Shull et al., 2004)

A **socialização** é um processo de compartilhamento de experiência, nele o conhecimento tácito é transferido entre indivíduos (Mendonça et al., 2004?; Nonaka e Takeuchi, 1997; Shull et al., 2004).

A **externalização** é um processo de articulação do conhecimento tácito em conhecimento explícito. Ela é a chave para criação do conhecimento pois cria conceitos novos e explícitos a partir do conhecimento tácito (Nonaka e Takeuchi, 1997). Os indivíduos explicitam seu conhecimento tácito para o grupo ou organização (Mendonça et al., 2004?; Shull et al., 2004).

A **combinação** envolve conjuntos diferentes de conhecimentos podendo levar a novos conhecimentos (Nonaka e Takeuchi, 1997). Conhecimento explícito pode ser reorganizado em conhecimento mais sofisticado ou abstrato (Mendonça et al., 2004?; Shull et al., 2004).

E a **internalização** é o processo de incorporação do conhecimento explícito no conhecimento tácito. Os indivíduos absorvem conhecimento explícito combinando-o com seu próprio conhecimento e experiências para produzir novo conhecimento tácito (Mendonça et al., 2004?; Shull et al., 2004).

3.8.1.1 Compartilhamento do Conhecimento no Contexto de Experimentação em Engenharia de Software

Uma adaptação do modelo Nonaka-Takeuchi para o contexto da experimentação em Engenharia de Software é proposta por Shull et al. (2004), ilustrada na Figura 3.8.

As fases do modelo original foram adaptadas para:

- **socialização** de conhecimento tácito entre replicadores: envolve discussão de projetos experimentais e seus riscos de validade, variáveis procedimentos, métodos de coleta de dados e procedimentos para análise dos dados. Para Shull et al. (2004) a interação entre os replicadores e os pesquisadores originais do experimento é fundamental para garantir a transferência do conhecimento tácito (o qual não é possível ser escrito no pacote de laboratório).
- **externalização** do conhecimento tácito (tácito para explícito) com pacotes de laboratório, registrando todas as informações úteis: projeto experimental, dados brutos, análises e resultados.
- melhoria do conhecimento explícito através da evolução do pacote de laboratório (**combinação**).

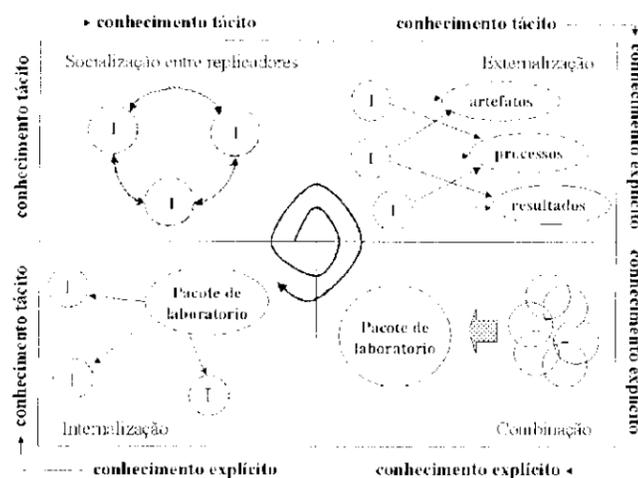


Figura 3.8: Modelo de Compartilhamento do Conhecimento na Engenharia de Software Experimental (EKSM - *Experimentation Knowledge Sharing Model*) (Mendonça et al., 2001?)

- **internalização** do conhecimento experimental pelo uso de pacotes de laboratório. Envolve a leitura, abstração, interpretação e execução do que está no pacote de laboratório.

Shull et al. (2004) ressaltam que os processos de compartilhamento do conhecimento nos níveis intra-grupo e inter-grupos possuem algumas características diferentes. No nível intra-grupo a socialização é facilitada devido ao grupo ter comunicação maior e experiências em comum. Mas em relação ao nível inter-grupos se torna um ponto crítico devido à comunicação limitada, interesses nem sempre convergentes e cultura diferente.

3.8.2 Paradigma de Melhoria da Qualidade (QIP)

Como solução proposta para atender as necessidades de compreensão, acompanhamento, controle e melhoria contínua de processo, empacotamento de experiências para formação de competência visando à reutilização, e também como mecanismo de aprendizado e infusão de novas tecnologias, Basili (1985) apresenta o paradigma de melhoria da qualidade (*Quality Improvement Paradigm* - QIP). O QIP é o resultado da aplicação do método científico para o problema de melhoria da qualidade de software (Basili et al., 1994),

baseando-se no princípio de que a engenharia de software é, por sua natureza, evolutiva e experimental.

O Paradigma de Melhoria da Qualidade implementa iterações de dois ciclos de realimentação (*feedback*) - o ciclo de aprendizado organizacional e o ciclo de aprendizado do projeto (Figura 3.9).

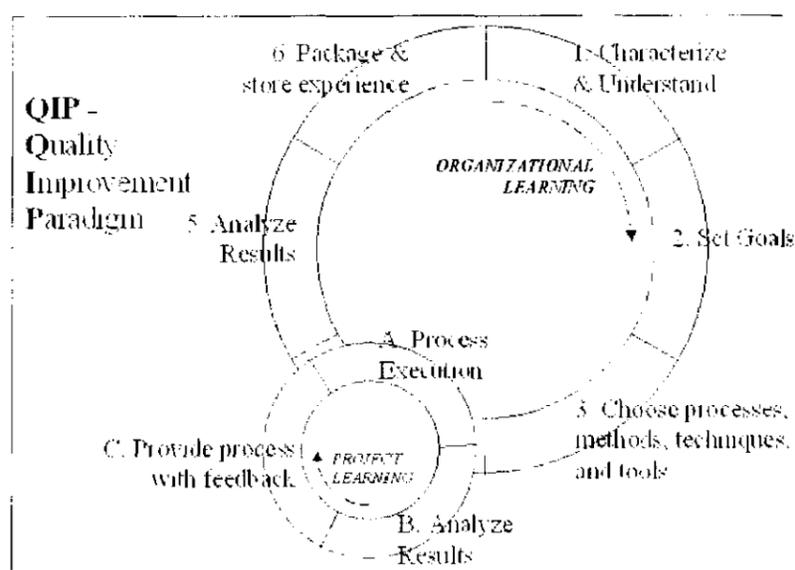


Figura 3.9: Ciclos de Realimentação (*Feedback*) - QIP (Kinnula, 2001)

Ciclo Organizacional

O Ciclo de aprendizado organizacional (ciclo maior na Figura 3.9) corresponde ao ciclo de controle, que representa como a organização aprende e fornece realimentação para o projeto durante a fase de execução. Indicadores quantitativos são úteis para prever e solucionar problemas, monitorar e dar suporte ao projeto, redirecionando o projeto para seus objetivos (Basili et al., 1994). O propósito desse ciclo é analisar a concordância e a discrepância dos dados coletados em comparação à experiências e modelos anteriores, o que aumenta a compreensão e acumula experiência reutilizável de forma que possa ser usada por outros projetos (Rombach (1994) apud Kinnula (2001)).

O ciclo de aprendizado organizacional é composto de seis etapas (Figura 3.10) (Basili, 1985; Basili et al., 1992, 1994):

1. **Caracterização:** É a etapa de início do ciclo. O objetivo é compreender e descrever o ambiente baseando-se nos modelos disponíveis, dados, intuição etc. e identificar os diversos fatores que podem influenciar o processo. Devem-se estabelecer referências quantificáveis baseadas nos processos existentes e caracterizar sua criticalidade.
2. **Definição de Metas:** Definir metas quantificáveis para o projeto ser bem sucedido e para desempenho e melhoria da organização. Essas metas devem ser definidas com base nas referências estabelecidas na etapa de caracterização (etapa 1).
3. **Escolha dos processos:** Com base na caracterização do ambiente e nas metas definidas, devem-se escolher os processos apropriados para melhoria, além dos métodos e ferramentas de suporte, certificando-se de que são consistentes com as metas estabelecidas.
4. **Execução:** Executar o processo construindo os produtos, coletando dados e fornecendo realimentação ao projeto baseado nos dados coletados. Compõe o ciclo de projeto.
5. **Análise:** Analisar os dados e as informações coletadas para avaliar as práticas atuais, determinar problemas, registrar descobertas, e fazer recomendações para futuras melhorias de projeto. É uma etapa chave do processo de melhoria. Os dados devem ser analisados para determinar quanto o projeto satisfaz as metas, onde os métodos foram mais efetivos, onde poderiam ser modificados e refinados, onde deve haver mais treinamento etc.
6. **Empacotamento:** Consolidar a experiência adquirida na forma de novos modelos ou modelos atualizados e refinados e outras formas de conhecimento estruturado obtido deste projeto ou de projetos anteriores. Armazenar em uma base de experiência de modo que esteja disponível para projetos futuros. Para empacotar experiências pode ser necessário diversas iterações do ciclo e diversos projetos para obter informações suficientes.

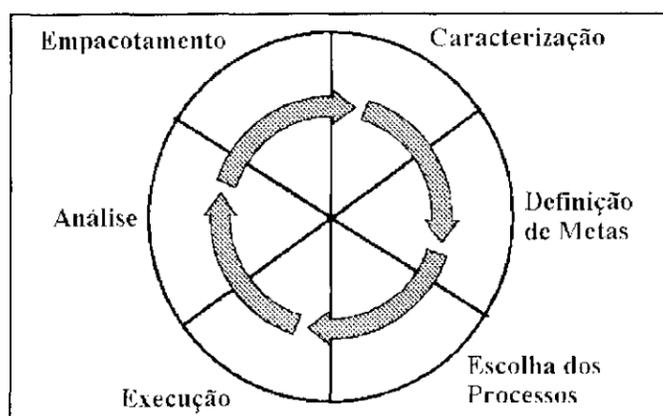


Figura 3.10: Seis Etapas do QIP (Basili et al., 1994)

Ciclo do Projeto

O ciclo de aprendizado do projeto, ou ciclo de capitalização, é o ciclo que implementa iterações da quarta etapa do ciclo organizacional, a etapa de execução (ciclo menor na Figura 3.9). É a realimentação (*feedback*) para a organização e tem o duplo propósito de (Basili et al., 1994):

- fornecer informação analítica sobre a execução do projeto, comparando os dados do projeto com as metas da organização e analisando concordância e discrepâncias;
- acumular experiência reutilizável que seja aplicável para outros projetos

Três atividades compõem o ciclo de projeto:

1. **Execução do Processo**
2. **Análise dos Resultados**
3. **Realimentação (*Feedback*) para o Processo**

O processo é executado gerando seus produtos e coletando, validando e analisando dados para avaliar a realização das metas. Estas informações são retornadas ao projeto para possíveis ações corretivas (Basili (1994) *apud* Kimmla (2001)).

3.8.2.1 Paradigma de Melhoria da Experimentação (EIP)

Com base no QIP, Mendonça et al. (2004?) propõem o Paradigma de Melhoria da Experimentação (*Experimentation Improvement Paradigm - EIP*) para engenharia de software empírica. O EIP é um *framework* para gerenciamento de conhecimento e melhoria da experimentação, com três ciclos (adaptação e extensão dos ciclos do QIP), como apresentado na Figura 3.11.

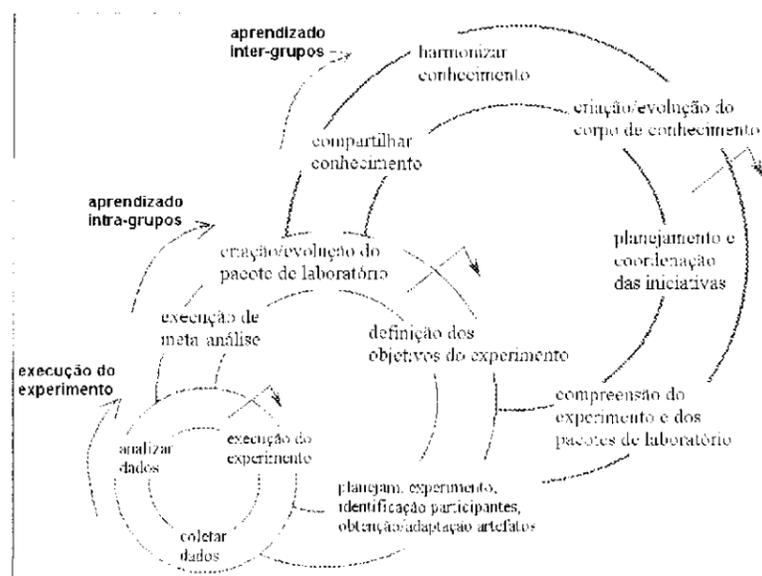


Figura 3.11: Ciclos do EIP (Mendonça et al., 2004?)

O ciclo de Execução do Experimento (ciclo menor), análogo ao ciclo de aprendizado do projeto no QIP, corresponde ao controle e execução do experimento em estudo. O ciclo de Aprendizado Intra-Grupo (ciclo do meio), similar ao ciclo de aprendizado organizacional no QIP, retrata o aprendizado, empacotamento e planejamento de novos experimentos dentro do grupo de pesquisadores, visando a construir um corpo de conhecimento local. As atividades desse ciclo consistem em:

- Etapa 1: Definir metas do experimento;
- Etapa 2: Planejar experimento, identificar participantes e obter/adaptar artefatos; preparação do experimento;

- Executar experimento (ciclo de Execução do Experimento):
- Etapa 3: Executar meta-análise: analisar resultados experimentais em conjunto com outros dados experimentais disponíveis;
- Etapa 4: Criar/Evoluir pacote de laboratório reutilizáveis, armazenando experiência e aprendida.

O ciclo de Aprendizado Inter-Grupos (ciclo maior) trata da padronização do pacote, evolução do pacote de experimentação e compartilhamento de conhecimento entre os grupos de pesquisa, visando a construir um amplo corpo de conhecimento baseado em uma série de replicações de experimento executadas por diferentes grupos de pesquisa. As atividades chave são:

- Etapa 1: Planejamento e coordenação das atividades experimentais entre os diferentes grupos;
- Etapa 2: Compreensão dos pacotes de laboratório e dos resultados experimentais disponibilizados por outros grupos de pesquisa;
- Execução de uma série de replicações e evolução do pacote de laboratório local (ciclo de aprendizado intra-grupo);
- Etapa 3: Compartilhamento e consolidação do aprendizado com outros grupos;
- Etapa 4: Harmonização do conhecimento: uniformizar o conhecimento adquirido pelos grupos;
- Etapa 5: Criação/Evolução do corpo de conhecimento (repositório de conhecimento).

A externalização e internalização do conhecimento na forma de pacotes de laboratório e a conversão do conhecimento tácito - que não está inserido nos pacotes de laboratório - são os pontos críticos do EIP.

3.9 Considerações Finais

Neste capítulo foi apresentado o processo de experimentação utilizado nos experimentos realizados ao longo deste trabalho. A experimentação tem se tornado uma maneira de produzir resultados que formem uma base de conhecimento que dê suporte aos desenvolvedores de software fornecendo subsídios à escolha de novos métodos e técnicas propostos na literatura. Com a replicação de experimentos pode-se consolidar, ou não, resultados anteriores, aumentando-se a significância estatística dos dados associados às técnicas exploradas nos mesmos. Exemplificando o processo de experimentação, foi apresentado o experimento realizado com o objetivo de avaliar a efetividade da técnica de leitura PBR (*Perspective-Based Reading*) na detecção de defeitos em documentos de especificação de requisitos, quando comparada com outras técnicas. Quatro replicações desse experimento foram realizadas no contexto do projeto Readers, sendo que duas delas (R1 e R2) foram comentadas neste capítulo. Com a experiência adquirida nessas replicações notou-se a necessidade de ferramentas que apoiem o processo de experimentação, para torná-lo mais sistemático e padronizado. Assim, algumas iniciativas de apoio à experimentação foram comentadas. Apresentou-se também o modelo de compartilhamento do conhecimento instanciado para experimentação, uma vez que um fator muito importante para que um experimento possa ser replicado é o conhecimento tácito associado ao processo. Esse tipo de conhecimento dificilmente pode ser colocado em papel e pode influenciar e dificultar uma replicação propriamente dita de um experimento anterior. Apresentou-se também o paradigma de melhoria da qualidade (QIP), o qual foi instanciado para o contexto da melhoria da experimentação (EIP), apresentando-se as etapas pelas quais se passa durante a replicação de um experimento realizado originalmente por outros pesquisadores. Muita interação é necessária e muitas etapas devem ser realizadas para que os grupos de replicadores atuais e originais possam se comunicar, tomar decisões e compartilhar conhecimento. Com base no EIP duas replicações do experimento PBR realizadas no projeto Readers serão analisadas no Capítulo 4.

Replicações nos Ambientes Acadêmico e Industrial

4.1 Considerações Iniciais

Na Seção 4.2 apresentam-se duas replicações do experimento com o pacote da técnica PBR, conduzidas no escopo deste trabalho e no âmbito do projeto *Readers*. Na Seção 4.3 são discutidas as etapas do paradigma de melhoria da experimentação (EIP) para essas replicações.

4.2 Terceira e Quarta Replicações - R3 e R4

Outras duas replicações foram conduzidas no âmbito do projeto *Readers*, num total de quatro replicações (Figura 4.1). A Seção 4.2.1 descreve a terceira replicação (R3), conduzida em ambiente acadêmico, e a Seção 4.2.2 relata a quarta replicação (R4), que foi conduzida em ambiente industrial.

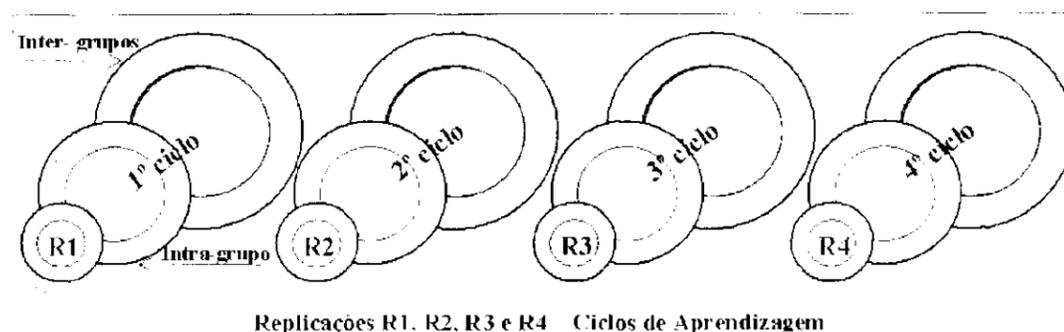


Figura 4.1: Replicações USP/UFSCar (adaptado de (Mendonça et al., 2001?))

4.2.1 Replicação em Ambiente Acadêmico - R3

A terceira replicação foi realizada em setembro de 2001 e, assim como R1 e R2, ocorreu em ambiente acadêmico. Os participantes da R3 tinham um perfil ligeiramente diferente, pois eram alunos de pós-graduação.

4.2.1.1 Objetivos do estudo

Os objetivos dessa replicação foram os mesmos da primeira e segunda replicações, que consistiam nas questões vindas do estudo original e de outras três questões que visam a analisar diretamente a técnica PBR.

Questões do estudo original:

- 1) Grupos utilizando PBR detectam mais defeitos que grupos utilizando *Checklist*?
- 2) Revisores, individualmente, encontraram mais defeitos utilizando PBR ou *Checklist*?
- 3) A experiência do revisor afeta sua efetividade?

Novas questões (incluídas a partir de R1):

- 4) Revisores individualmente acharam defeitos diferentes usando PBR e *Checklist*?
- 5) As perspectivas PBR têm a mesma efetividade?
- 6) As perspectivas PBR acham tipos de defeitos diferentes?

4.2.1.2 Participantes

Essa replicação contou com a participação de 18 estudantes de pós-graduação em computação da Universidade de São Paulo e da Universidade Federal de São Carlos. Um levantamento do perfil dos participantes foi feito através de um formulário (*Analyst Survey Form*), no qual a maioria dos participantes (66,67%) classificou-se com Avançado (*high*) quanto à proficiência da língua utilizada no experimento (inglês), no item Leitura.

4.2.1.3 Materiais

Foram utilizados os mesmos dois documentos de especificação de requisitos das replicações 1 e 2: ATM (*Automated Teller Machine*) e PG (*Parking Garage*). O documento ATM possui 17 páginas, 39 requisitos, sendo 26 funcionais e 13 não-funcionais, e 37 defeitos e o documento PG possui 17 páginas, 37 requisitos (21 funcionais e 16 não-funcionais) e 32 defeitos. As técnicas aplicadas foram *Checklist* e PBR, com três perspectivas: projetista, testador e usuário.

4.2.1.4 Procedimento

Projeto: O projeto do experimento (Tabela 4.1) foi semelhante ao da segunda replicação, alterando apenas a quantidade de dias do experimento. No primeiro dia, os participantes receberam treinamento na técnica *Checklist*, aplicando-a no segundo dia (Grupo 1 revisou o documento de requisitos PG e o Grupo 2, ATM). No terceiro dia cada participante foi treinado em uma das três perspectivas da técnica PBR e aplicou-a no outro documento de requisitos (Grupo 1 revisou ATM e Grupo 2 revisou PG).

Tabela 4.1: Projeto Experimental R3

Técnica	Grupo 1			Grupo 2			
	Projetista 3 partic.	Testador 3 partic.	Usuário 3 partic.	Projetista 3 partic.	Testador 3 partic.	Usuário 3 partic.	
Checklist	Treinamento			Treinamento			1o. Dia
	Aplicação no documento PG			Aplicação no documento ATM			2o. Dia
PBR	Treinamento			Treinamento			3o. Dia
	Aplicação no documento ATM			Aplicação no documento PG			

Treinamento: O treinamento teórico foi modificado após R2, enfatizando mais a importância da inspeção e detalhando e exemplificando a aplicação das técnicas. No treinamento das técnicas foi utilizado um documento de especificação de requisitos diferente do utilizado nas duas primeiras replicações, o *Gas Station Control System-GS*, que é um documento reduzido, com apenas 2 páginas e 10 requisitos (8 funcionais e 2 não-funcionais). O tempo de treinamento prático de cada técnica também foi reduzido para 30 minutos. No primeiro dia os participantes foram treinados na técnica Checklist e no terceiro dia, no período da manhã, na técnica PBR.

Aplicação das técnicas: A aplicação das técnicas nos documentos de requisitos teve duração máxima de 1h15min, cada. A aplicação do *Checklist* nos documentos ocorreu no segundo dia, onde os participantes do Grupo 1 aplicaram no PG e os do Grupo 2 aplicaram no ATM. No terceiro dia houve a aplicação da PBR nos documentos no período da tarde (Grupo 1 utilizou ATM e Grupo 2 aplicou no PG).

Coleta de dados: Foram utilizadas quatro métricas para analisar os dados coletados:

- **Defeitos encontrados:** número de defeitos encontrados utilizando uma técnica específica;
- **Ocorrência de defeitos:** número total de defeitos encontrados considerando-se todos os participantes. O total de ocorrência é calculado por

$$\sum_{i=1}^n (x_i)$$

onde x_i é o número de defeitos encontrados pelo participante i .

- **Efetividade:** percentagem média dos defeitos encontrados por um grupo de participantes. A efetividade é calculada por

$$\left(\sum_{i=1}^n (x_i/y) \right) * 100/n$$

onde x_i é o número de defeitos encontrados pelo participante i , y é o número total de defeitos no documento e n é o número de participantes no grupo.

- **Eficiência:** média dos defeitos encontrados por cada participante por hora. A eficiência é calculada por

$$\left(\sum_{i=1}^n (x_i/k_i) \right) / n$$

onde x_i é o número de defeitos encontrados pelo participante i , k_i é o tempo (em horas) usado pelo participante i e n é o número de participantes no grupo.

4.2.1.5 Resultados

Grupos utilizando PBR detectam mais defeitos que grupos utilizando Checklist?

A hipótese nula (H_0) estabelecida, que diz que não há diferença na razão de detecção de defeitos de grupos que aplicam PBR dos grupos que aplicam *Checklist*, não pôde ser rejeitada com um valor igual a 0.88 por ser maior que o nível de significância adotado (0.05). Foram gerados, através do teste de permutação, 48620 modos de ordenar os revisores em grupos de 9 e a ordenação do grupo sem diluição foi 42730%. Resultado semelhante foi encontrado na primeira replicação (R1) (Maldonado et al., 2004a), contrariando o resultado encontrado no estudo original (Basili et al., 1996).

Revisores individualmente encontraram mais defeitos utilizando PBR ou Checklist?

As três hipóteses abaixo foram formuladas para avaliar os resultados individuais dos participantes e testá-los em relação à efetividade e eficiência (variáveis dependentes).

1. Efeito da interação Técnica de Leitura (RT) X Documento de Requisitos (RD)
 - H_0 : Não há diferença entre o grupo que aplicou PBR e o grupo que aplicou *Checklist* com respeito à efetividade/eficiência dos participantes, individualmente.
 - H_a : O grupo que aplicou PBR e o grupo que aplicou *Checklist* diferem com respeito à efetividade/eficiência dos participantes, individualmente.
2. Efeito principal RT
 - H_0 : Não há diferença entre os participantes aplicando PBR e participantes aplicando *Checklist*, com respeito à efetividade/eficiência individual.
 - H_a : Participantes aplicando PBR e participante aplicando *Checklist* diferem com respeito à efetividade/eficiência individual.

3. Efeito principal RD

H_0 : Não há diferença entre participantes revisando ATM e participantes revisando PG com respeito à efetividade/eficiência individual.

H_a : Participantes revisando ATM e participantes revisando PG diferem com respeito à efetividade/eficiência individual.

Os resultados estatísticos são apresentados nas Tabelas 4.2 e 4.3 (percentagem média MINITAB). Para o efeito RT, a hipótese não pôde ser rejeitada porque o valor de significância P está acima de 5% (0.588), o que significa que as variáveis influenciam no resultado. Por outro lado, H_0 pôde ser rejeitada para o efeito RD porque o valor P está abaixo de 5% (0.012), o que significa que as variáveis não têm influência nos resultados. Para o efeito do grupo (RT X RD), a H_0 também pode ser rejeitada (P=0.024).

Tabela 4.2: ANOVA em Relação à Efetividade

	Variáveis Independentes		
	RT X RD	RT	RD
Efetividade	-	<i>Checklist</i> = 13.402; PBR = 12.009	ATM = 9.610; PG = 15.801
P	0.024	0.588	0.012

Em relação à eficiência, H_0 não pode ser rejeitada para o efeito RT, porque o valor de significância P foi acima de 5% (0.860), como pode ser observado na Tabela 4.3. H_0 também não pode ser rejeitada para RD (P=0.202) nem para o efeito do grupo RT X RD (P=0.058).

Tabela 4.3: ANOVA em Relação à Eficiência

	Variáveis Independentes		
	RT X RD	RT	RD
Eficiência	-	<i>Checklist</i> = 3.014; PBR = 2.897	ATM = 2.535; PG = 3.376
P	0.058	0.860	0.202

O total de defeitos encontrados, o total de ocorrência de defeitos, a efetividade e a eficiência das técnicas para ambos documentos estão na Tabela 4.4. A técnica *Checklist* encontrou maior número de defeitos diferentes (23 defeitos encontrados de 37 contidos no documento) e maior número de ocorrências de defeitos observados para o documento

ATM (13 ocorrências das 333 possíveis). Para o documento PG, a técnica PBR encontrou um maior número de defeitos diferentes (18 de 32) e maior número de ocorrências também (51 de 288). Quanto à efetividade e eficiência, a técnica *Checklist* foi mais eficaz (12.91%) e mais eficiente (3.21%) para o documento ATM. Enquanto que a técnica PBR foi mais eficaz (17.71%) e mais eficiente (3.94%) para o documento PG.

Tabela 4.4: Resultado das Técnicas por Documento de Requisitos

Métricas	Doc. ATM		Doc. PG	
	<i>Checklist</i>	PBR	<i>Checklist</i>	PBR
Defeitos Encontrados	23/37 (62.2%)	11/37 (29.7%)	13/32 (40.6%)	18/32 (56.3%)
Ocorrência de Defeitos	13/333	21/333	40/288	51/288
Efetividade	12.91	6.31	13.89	17.71
Eficiência	3.21	1.86	2.82	3.94

A experiência do revisor afeta sua efetividade?

Através do formulário para levantamento do perfil (*Analyst Survey Form*) calculou-se a média da experiência dos participantes. Nessa replicação, diferentemente das anteriores devido às mudanças efetuadas no questionário, atribuiu-se uma pontuação relacionada com intervalos de experiência. Na Figura 4.2 mostra-se que a relação entre experiência e efetividade é fraca (Coeficiente de Correlação Pearson = -0,036).

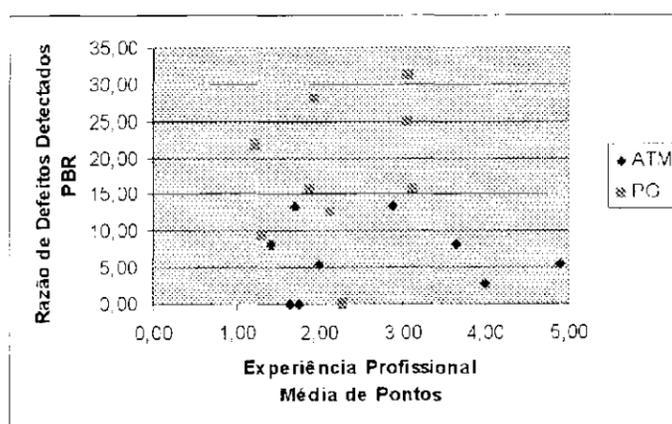


Figura 4.2: PBR - Efetividade versus Experiência

Individualmente, os revisores encontraram defeitos diferentes utilizando PBR e Checklist?

Na Figura 4.3 mostram-se os defeitos encontrados por técnica e por documento. A parte (a) mostra os defeitos encontrados por ambas as técnicas no documento ATM. As duas técnicas encontraram 9 defeitos em comum. 14 defeitos foram encontrados apenas pela técnica *Checklist* e outros 2 apenas pela PBR. No documento PG (Figura 4.3(b)) as duas técnicas encontraram 11 defeitos em comum. Isoladamente, *Checklist* detectou 2 defeitos e PBR, 7.

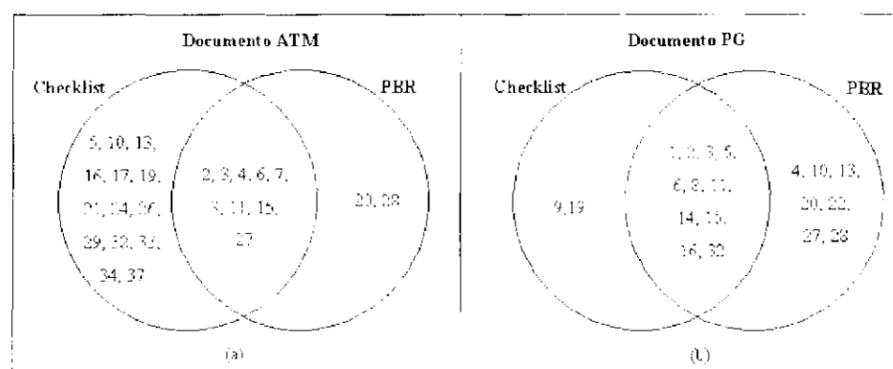


Figura 4.3: ATM/PG - Defeitos Encontrados pelas Técnicas

Nas Tabelas 4.5 a e b apresentam-se os tipos de defeitos mais encontrados pelas técnicas nos documentos ATM e PG, respectivamente. Para defeitos do tipo Ambigüidade as duas técnicas foram igualmente efetivas nos dois documentos. Em outros casos como, por exemplo, para o tipo Omissão a técnica *Checklist* (71.43) foi mais efetiva que a técnica PBR (21.43) considerando o documento ATM, mas para o documento PG a PBR foi melhor (58.33).

As perspectivas PBR têm a mesma efetividade?

Pode-se observar, pela Tabela 4.6, que a perspectiva do projetista obteve melhor média de defeitos encontrados (8.11%) do que as outras duas perspectivas, para o documento ATM. E para este mesmo documento, a perspectiva do testador obteve a maior razão defeitos/hora, 2.49%. Para o documento PG, a perspectiva do testador conseguiu a melhor média de defeitos encontrados (20.93%) e a maior razão defeitos/hora (5.85%).

Tabela 4.5: Percentagem de Defeitos e Ocorrência de Defeitos por Tipo de Defeito

(a) ATM						
Tipo	Número de Defeitos	% de Defeitos Encontrados		Possíveis Ocorrências de Defeitos	% do Número de Ocorrências	
		Checklist	PBR		Checklist	PBR
A	8	50.00	50.00	72	6.94	9.72
E	1	0.00	0.00	9	0.00	0.00
II	4	50.00	25.00	36	19.44	5.56
IF	8	75.00	37.50	72	19.44	12.50
MD	2	50.00	0.00	18	5.56	0.00
O	14	71.43	21.43	126	12.70	2.38

(b) PG						
Tipo	Número de Defeitos	% de Defeitos Encontrados		Possíveis Ocorrências de Defeitos	% do Número de Ocorrências	
		Checklist	PBR		Checklist	PBR
A	4	75.00	75.00	36	33.33	33.33
E	1	0.00	100.00	9	0.00	11.11
II	10	40.00	50.00	90	4.44	12.22
IF	2	50.00	100.00	18	38.89	66.67
MD	3	0.00	0.00	27	0.00	0.00
O	12	41.67	58.33	108	15.74	13.89

Os resultados da Análise de Variância (ANOVA) para verificar a influência do fator Perspectiva na efetividade e na eficiência dos participantes indicaram que não há influência em um nível de significância de 0.05 ($p=0.727$ para efetividade e $p=0.113$ para eficiência).

Tabela 4.6: Percentagem Média dos Defeitos Encontrados e Razão de Defeitos Observados

Perspectiva	% de Defeitos Encontrados (média)			Razão defeitos observados (defeito/hora)		
	ATM	PG	ATM+PG	ATM	PG	ATM+PG
Projetista	8.11	13.54	10.82	1.72	2.49	2.10
Testador	7.21	20.83	14.02	2.49	5.85	4.17
Usuário	3.60	18.75	11.17	1.35	3.46	2.40
Média	6.31	17.71	12.01	1.86	3.94	2.90

As perspectivas PBR descobrem tipos de defeitos diferentes?

Na parte (a) das Figuras 4.4 e 4.5 mostram-se os defeitos encontrados pelas perspectivas individualmente e em comum (nas intersecções dos círculos) e na parte (b) mostra-se que perspectiva encontrou o maior número de ocorrência para cada defeito.

Para o documento ATM as perspectivas encontram poucos defeitos em comum, o máximo de 2 em comum que foram encontrados pelo projetista e pelo usuário, apenas 1 defeito em comum foi encontrado pelas três perspectivas. Considerando o documento PG, 7 defeitos foram encontrados em comum pelas três perspectivas.

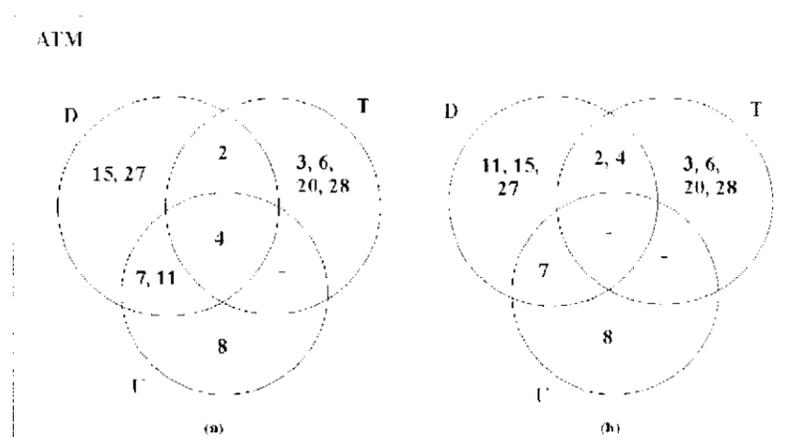


Figura 4.4: ATM: (a) Defeitos Encontrados pelas Perspectivas; (b) Defeitos Agrupados pela Perspectiva com Melhor Desempenho

4.2.1.6 Riscos à validade

Os riscos à validade foram praticamente todos os identificados para as replicações 1 e 2, resumidos abaixo, com exceção do risco quanto à maturidade experimental dos replicadores, pois essa foi a terceira replicação conduzida por eles.

- Linguagem: Todo o material utilizado no experimento está em inglês, então a falta de proficiência no inglês pode ter efeito nos resultados.
- Aspectos Culturais: Algumas outras diferenças culturais, além da linguagem dos participantes podem ter influência, como, por exemplo, o conhecimento informal nos domínios de aplicação dos documentos de especificação de requisitos utilizados.

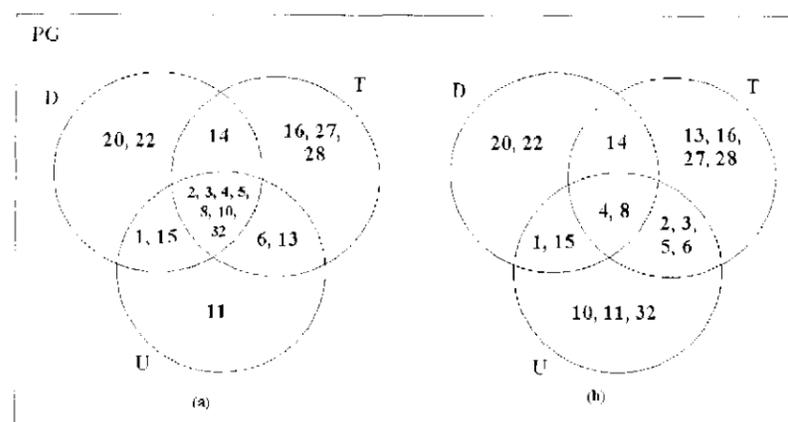


Figura 4.5: PG: (a) Defeitos Encontrados pelas Perspectivas; (b) Defeitos Agrupados pela Perspectiva com Melhor Desempenho

- Conformidade do Processo: O documento utilizado para o treinamento prático foi alterado, sendo utilizado um menor (*Gas Station*) com tempo de aplicação da técnica reduzido para 30 minutos. Quanto ao tempo para maturidade e assimilação dos conceitos aprendidos, a técnica *Checklist* teve um tempo maior entre o treinamento, realizado no primeiro dia, e a aplicação, realizada no dia seguinte. O treinamento e a aplicação na técnica PBR foram efetuados no mesmo dia.
- Nível de Experiência dos Participantes: a maioria dos participantes não tinha muita experiência como profissionais da indústria.

4.2.2 Replicação em Ambiente Industrial - R4

A quarta replicação ocorreu em setembro de 2001, com profissionais e estudantes de pós-graduação. Além dos objetivos principais do experimento em estudo, essa replicação possibilitou apresentar as técnicas para as empresas participantes como uma opção das técnicas de VV&T, gerando interesse em estudar a viabilidade de introduzi-la no processo da empresa.

4.2.2.1 Objetivos do estudo

Os objetivos são os mesmos das replicações anteriores e já descritos na Seção 4.2.1.1.

4.2.2.2 Participantes

Essa replicação foi planejada com 18 participantes, sendo 12 profissionais de software do CPqD e 3 EMBRAPA e 3 alunos de pós-graduação da UNICAMP.

Quanto ao perfil dos participantes, feito através de um formulário (*Analyst Survey Form*), a maioria dos participantes classificou-se como Avançado (73.33%) quanto à proficiência da língua utilizada no experimento (inglês), no item Leitura.

4.2.2.3 Materiais

Foram utilizados os mesmos dois documentos de especificação de requisitos e as mesmas técnicas descritos para a terceira replicação.

4.2.2.4 Procedimento

Projeto: O projeto do experimento (Tabela 4.7) foi semelhante ao da primeira replicação. No primeiro dia, os participantes receberam treinamento na técnica *Checklist*, aplicando-a nos documentos (Grupo 1 revisou o documento de requisitos PG e o Grupo 2, ATM). No segundo dia cada participante foi treinado em uma das três perspectivas da técnica PBR e aplicou-a no outro documento de requisitos (Grupo 1 revisou ATM e Grupo 2 revisou PG).

Tabela 4.7: Projeto Experimental R4

Técnica	Grupo 1			Grupo 2			
	Projetista 3 partic.	Testador 3 partic.	Usuário 3 partic.	Projetista 3 partic.	Testador 3 partic.	Usuário 3 partic.	
Checklist	Treinamento			Treinamento			1o. Dia
	Aplicação no documento PG			Aplicação no documento ATM			
PBR	Treinamento			Treinamento			2o. Dia
	Aplicação no documento ATM			Aplicação no documento PG			

Devido a problemas de troca de material e ausência de dois participantes durante a aplicação das técnicas, o experimento contou com apenas 15 participantes. O grupo 1 que era composta por profissionais do CPqD ficou com apenas 6 participantes. Na aplicação da técnica PBR, esse grupo teve 2 participantes utilizando a perspectiva do projetista, 3 utilizando do testador e 1 do usuário.

Treinamento: O treinamento das técnicas foi semelhante ao do R3, sendo que no primeiro dia os participantes foram treinados na técnica *Checklist* e no segundo na técnica PBR.

Aplicação das técnicas: A aplicação das técnicas nos documentos de requisitos teve a seqüência apresentada no projeto experimental e mesmo tempo de duração da terceira replicação, ou seja, 1h45min.

Coleta de dados: Foram utilizadas as mesmas métricas descritas para a terceira replicação.

4.2.2.5 Resultados

Grupos utilizando PBR detectam mais defeitos que grupos utilizando Checklist?

De forma análoga às três replicações anteriores, a hipótese nula (H_0) estabelecida (não há diferença na razão de detecção de defeitos de grupos que aplicam PBR dos grupos que aplicam *Checklist*) não pôde ser rejeitada para um valor de significância igual a 0,99 (no nível de significância de 0,05). Foram gerados, através do teste de permutação, 5005 modos de ordenar os revisores em grupos de 9 (havia apenas 15 participantes) e a ordenação do grupo sem diluição foi 4956^o.

Revisores individualmente encontraram mais defeitos utilizando PBR ou Checklist?

Os resultados estatísticos são apresentados nas Tabelas 4.8 e 4.9 (percentagem média MINITAB). Para o efeito RT, H_0 pode ser rejeitada porque o valor de significância P está abaixo de 5% (0.014), o que significa que as variáveis não influenciam no resultado. Por outro lado, H_0 não pode ser rejeitada para o efeito RD porque o valor P está acima de 5% (0.690). O valor de significância P não pôde ser calculado para o efeito do grupo (RT X RD) devido ao projeto desta replicação ter ficado desbalanceado (grupos com quantidade diferente de participantes).

Em relação à eficiência, H_0 pode ser rejeitada para o efeito RT, porque o valor de significância P ficou abaixo de 5% (0.005), como pode ser observado na Tabela 4.9, mas para RD, a hipótese H_0 não pode ser rejeitada (P=0.388). Não foi possível calcular o valor de significância P para o efeito do grupo RT X RD.

Tabela 4.8: ANOVA em Relação à Efetividade

		Variáveis Independentes	
		RT X RD	
Efetividade	-	Checklist = 16.453; PBR = 9.353	ATM = 13.513; PG = 12.293
P		0.014	0.690

Tabela 4.9: ANOVA em Relação à Eficiência

		Variáveis Independentes	
		RT X RD	
Eficiência	-	Checklist = 3.657; PBR = 1.896	ATM = 3.063; PG = 2.490
P		0.005	0.388

O total de defeitos encontrados, o total de ocorrência de defeitos, a efetividade e a eficiência das técnicas para ambos documentos são apresentados na Tabela 4.10. A técnica *Checklist* encontrou maior número de defeitos diferentes (22 defeitos encontrados de 37 contidos no documento) e maior número de ocorrências de defeitos observados para o documento ATM (52 ocorrências das 333 possíveis). O grupo que aplicou PBR nesse documento detectou 15 defeitos dos 37, sendo 23 ocorrências de 222 possíveis, pois esse grupo possuía apenas 6 participantes. Para o documento PG, a técnica *Checklist* também encontrou um maior número de defeitos diferentes (16 de 32) e maior número de ocorrências (34 de 192 - grupo com apenas 6 participantes). A PBR encontrou 11 defeitos e o número de ocorrências foi 25 (de 288 possíveis). Quanto à efetividade e eficiência considerando o documento ATM, a técnica *Checklist* foi mais eficaz (15.62%) e mais eficiente (3.70%) que a PBR (10.36% de eficácia e 2.11% de eficiência). O mesmo ocorreu para o documento PG. *Checklist* obteve 17.71% de eficácia (PBR obteve 8.68%) e 3.59% de eficiência (contra 1.76% para PBR).

A experiência do revisor afeta sua efetividade?

A experiência dos participantes foi calculada do mesmo modo que em R3, através de uma pontuação relacionada com intervalos de experiência. Na Figura 4.6 mostra-se que a relação entre experiência e efetividade é fraca (Coeficiente de Correlação Pearson = -0.175). Resultados análogos foram encontrados nas outras três replicações.

Tabela 4.10: Resultado das Técnicas por Documento de Requisitos

Métricas	Doc. ATM		Doc. PG	
	Checklist	PBR	Checklist	PBR
Defeitos Encontrados	22/37 (59.5%)	15/37 (40.5%)	16/32 (50.0%)	11/32 (34.4%)
Ocorrência de Defeitos	52/333	23/222	34/192	25/288
Efetividade	15.62	10.36	17.71	8.68
Eficiência	3.70	2.11	3.59	1.76

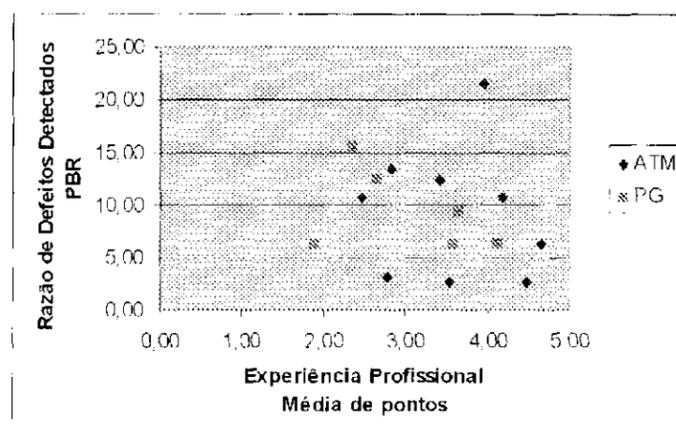


Figura 4.6: PBR - Efetividade versus Experiência

Revisores individualmente encontraram defeitos diferentes utilizando PBR e Checklist?

Na Figura 4.7 mostra-se os defeitos encontrados por técnica e por documento. A parte (a) mostra os defeitos encontrados por ambas as técnicas no documento ATM. 11 defeitos foram encontrados em comum pelas duas técnicas. 11 defeitos foram encontrados apenas pela técnica *Checklist* e outros 4 apenas pela PBR. No documento PG (Figura 4.7 (b)) as duas técnicas encontraram 8 defeitos em comum. Isoladamente, *Checklist* detectou 3 defeitos e PBR, 7.

Nas Tabelas 4.11 a e b mostram-se os tipos de defeitos mais encontrados pelas técnicas nos documentos ATM e PG, respectivamente. Para defeitos do tipo Omissão, por exemplo, considerando o documento ATM as duas técnicas foram igualmente efetivas (50.00), mas para o documento PG a PBR foi um pouco melhor (50.00) do que a técnica *Checklist* (41.67). Ressalta-se que as quantidades de possíveis ocorrências de defeitos são distintas

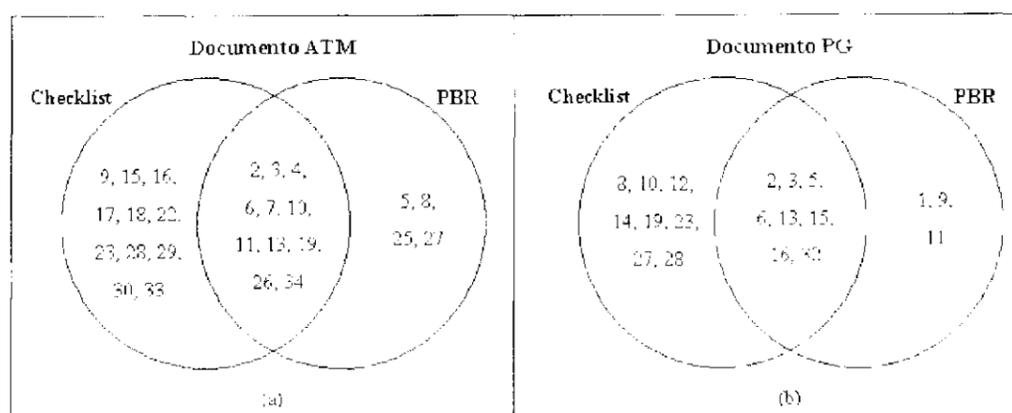


Figura 4.7: ATM/PG - Defeitos Encontrados pelas Técnicas

para cada técnica devido ao número de participantes dos grupos que as aplicaram ter sido diferente.

Tabela 4.11: Percentagem de Defeitos e Ocorrência de Defeitos por Tipo de Defeito

(a) ATM							
Tipo	Número de Defeitos	% de Defeitos Encontrados		Possíveis Ocorrências de Defeitos		% do Número de Ocorrências	
		Checklist	PBR	Checklist	PBR	Checklist	PBR
A	8	62.50	37.50	72	48	11.11	14.58
E	1	0.00	0.00	9	6	0.00	0.00
II	4	75.00	25.00	36	24	19.44	4.17
IF	8	75.00	50.00	72	48	29.17	12.50
MD	2	50.00	0.00	18	12	5.56	0.00
O	14	50.00	50.00	126	84	7.94	10.71

(b) PG							
Tipo	Número de Defeitos	% de Defeitos Encontrados		Possíveis Ocorrências de Defeitos		% do Número de Ocorrências	
		Checklist	PBR	Checklist	PBR	Checklist	PBR
A	4	100.00	50.00	24	36	45.83	19.44
E	1	0.00	0.00	6	9	0.00	0.00
II	10	30.00	30.00	60	90	6.67	3.33
IF	2	100.00	50.00	12	18	41.67	16.67
MD	3	33.33	0.00	18	27	5.56	0.00
O	12	50.00	41.67	72	108	18.06	11.11

As perspectivas PBR têm a mesma efetividade?

Pela Tabela 4.12, pode-se observar que a perspectiva do testador obteve melhor média de defeitos encontrados (14.41%) e maior razão defeitos/hora (2.96%) do que as outras duas perspectivas, considerando-se o documento ATM. Para o documento PG, foi a perspectiva do usuário que conseguiu a melhor média de defeitos encontrados (11.46%) e a maior razão defeitos/hora (2.07%). Os resultados da Análise de Variância (ANOVA) para verificar a influência do fator Perspectiva na efetividade e na eficiência dos participantes indicaram que não há influência em um nível de significância de 0.05 ($p=0.427$ para efetividade e $p=0.727$ para eficiência).

Tabela 4.12: Percentagem Média dos Defeitos Encontrados e Razão de Defeitos Observados

Perspectiva	% de Defeitos Encontrados (média)			Razão defeitos observados (defeito/hora)		
	ATM	PG	ATM+PG	ATM	PG	ATM+PG
Projetista	8.11	6.25	7.18	1.58	1.75	1.66
Testador	14.41	8.33	11.37	2.96	1.45	2.20
Usuário	2.70	11.46	7.08	0.57	2.07	1.32
Média	8.41	8.68	8.54	1.71	1.76	1.73

As perspectivas PBR descobrem tipos de defeitos diferentes?

Na parte (a) das Figuras 4.8 e 4.9 mostram-se os defeitos encontrados pelas perspectivas individualmente e em comum (nas intersecções dos círculos) e na parte (b) mostra-se que perspectiva encontrou o maior número de ocorrência para cada defeito.

Para o documento ATM as perspectivas encontram apenas um defeito em comum e a perspectiva do testador encontrou isoladamente nove defeitos. Considerando o documento PG, o número de defeitos encontrados pelas perspectivas isoladamente e em comum foram praticamente iguais.

4.2.2.6 Riscos à validade

Os riscos à validade interna são os mesmos considerados para a terceira replicação R3 e descritos na Seção 4.2.1.6, com exceção do item Nível de Experiência dos Participantes, pois os participantes da quarta replicação eram, em sua maioria, profissionais de software.

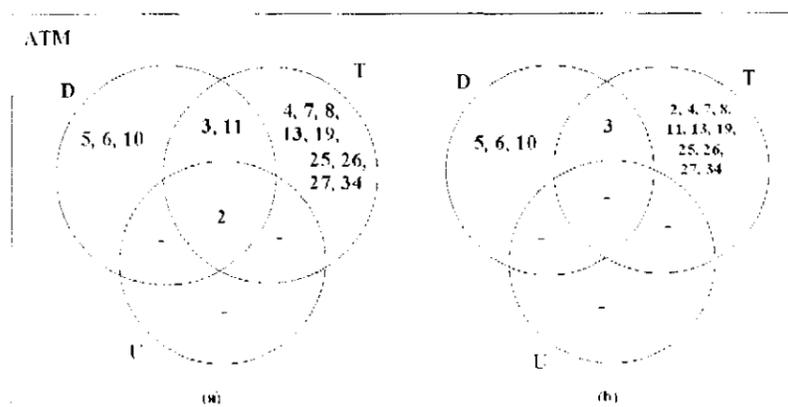


Figura 4.8: ATM: (a) Defeitos Encontrados pelas Perspectivas; (b) Defeitos Agrupados pela Perspectiva com Melhor Desempenho

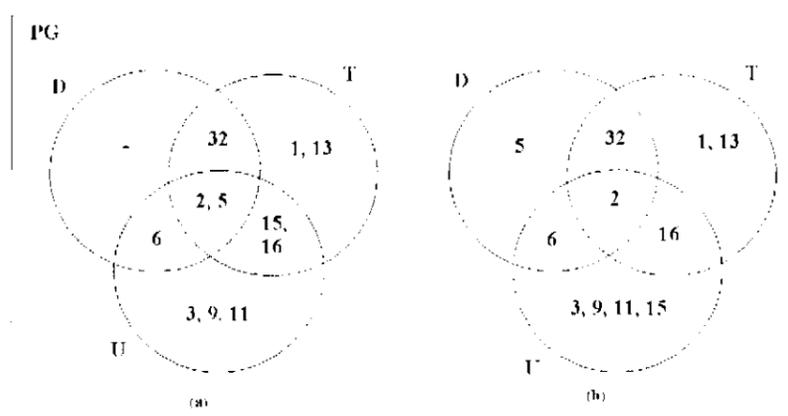


Figura 4.9: PG: (a) Defeitos Encontrados pelas Perspectivas; (b) Defeitos Agrupados pela Perspectiva com Melhor Desempenho

Além dos riscos já considerados podemos acrescentar o risco de maturação, que ocorre em função do tempo, como cansaço ou tédio (Basili et al., 1996). Nesse experimento, esse efeito poderia ser devido à aplicação dos testes durante dois dias seguidos, nos dois turnos, com intervalo apenas para o almoço.

4.2.2.7 Evolução do pacote

Após as execuções de R3 e R4 as listas de defeitos dos documentos ATM e PG foram atualizadas acrescentando-se dois novos defeitos em cada uma. A lista do documento ATM

ficou com 39 defeitos e a do documento PG com 34 defeitos. O material do treinamento também foi atualizado, criando-se versões do treinamento direcionadas aos diversos perfis de participantes. Participantes menos experientes devem receber treinamento mais detalhado nos modelos subjacentes, na taxonomia e nas técnicas, enquanto que para participantes mais experientes o treinamento deve focar mais na própria técnica.

4.3 Melhoria da Experimentação Durante as Replicações R3 e R4

Nesta seção os ciclos inter-grupos e intra-grupo para melhoria da experimentação são analisados no contexto das replicações R3 e R4. Na Seção 4.3.1 são discutidas as etapas do EIP para a replicação R3, que é o terceiro ciclo de aprendizado dentro do projeto *Readers* e a Seção 4.3.2 discute a replicação R4, o quarto ciclo de aprendizado. Os dois primeiros ciclos, replicações R1 e R2, são discutidas por Mendonça et al. (2004?).

4.3.1 Terceiro Ciclo de Aprendizado - R3

Ciclo de Aprendizado Inter-grupos do R3

Etapa 1 - Planejamento e coordenação das iniciativas: A execução de diversas replicações do experimento original já era objetivo do projeto *Readers* visando a obter uma maior quantidade de dados experimentais, preferencialmente com participantes de perfis e ambientes diferentes, para assim confrontar os resultados obtidos e melhorar ajustando, a cada replicação, o processo experimental. A terceira replicação foi planejada com dois intuitos, além dos objetivos iniciais do projeto: validar o novo treinamento, que tinha sofrido alterações, e obter resultados da aplicação das técnicas por participantes com perfil diferente das replicações anteriores, pois nesse caso eram estudantes do curso de pós-graduação.

Etapa 2 - Compreensão do experimento e dos pacotes de laboratório: Compreensão já adquirida em R1 e R2.

Ciclo de Aprendizado Intra-grupo do R3

Etapa 1 - Definição dos objetivos do experimento: Os objetivos eram os mesmos das replicações R1 e R2 (apresentados na Seção 4.2.1.1).

Etapa 2 - Preparação do experimento: O projeto experimental da terceira replicação seguiu a mesma seqüência de técnicas e documentos do R2, mas sendo executado em 3 dias consecutivos. No primeiro e no segundo dias foram no período noturno, devido à disponibilidade de horário dos participantes. Além das alterações efetuadas no treinamento, o formulário de levantamento do perfil (*Analyst Survey Form*) foi modificado. A questão sobre tempo de experiência nas funções Manager/Developer/Tester/Analyst foi retirada, acrescentando-se questões sobre experiência com requisitos, em projeto e em testes de software. Essa alteração foi proposta, após a execução de R2, de forma que o formulário possa ser usado por todos os tipos de experimentos no âmbito do projeto *Readers* (PBR, OORTs e Basili-Selby). O formulário de registro de defeitos (*Defect Collection Form*) também foi alterado recebendo uma coluna para registro do tipo de defeito, pois só havia para classe de defeito e o questionário de opiniões (*Feedback Questionnaire*) foi reestruturado agrupando as questões por assunto avaliado. As planilhas de coleta de dados, utilizadas para registro dos resultados dos experimentos, foram padronizadas com os outros grupos do projeto *Readers*, como foi decidido após a execução de R2 (Mendonça et al., 2004?).

Ciclo de Execução do Experimento do R3

Essa replicação seguiu os mesmos passos das anteriores, como descrito nas Seções 3.6.1 e 3.6.2 do Capítulo 3.

Ciclo de Aprendizado Intra-grupo do R3 - cont.

Etapa 3 - Execução de meta-análise: Comparando os resultados de R3 com as duas replicações anteriores (Gráficos 4.10 e 4.11), observa-se que para o documento ATM e a técnica *Checklist*, R3 obteve melhores resultados que R1 e R2, acontecendo o inverso para a técnica PBR. Considerando o documento PG e *Checklist*, R3 foi ligeiramente pior do que R2 e ambas obtiveram desempenho menor do que R1. Para

a técnica PBR, os resultados de R3 foram mais próximos dos resultados de R2 e ambas foram melhor do que R1.

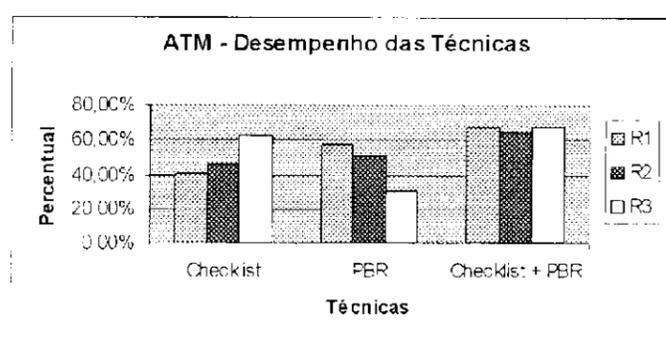


Figura 4.10: Desempenho de R1, R2 e R3 para o Documento ATM

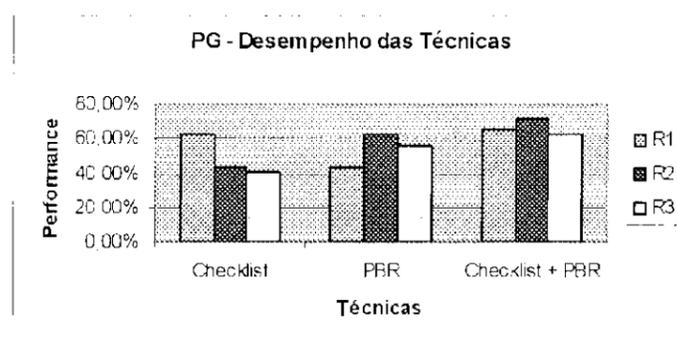


Figura 4.11: Desempenho de R1, R2 e R3 para o Documento PG

Etapa 4 - Criação/Evolução do pacote de laboratório: Não houve alterações no pacote após a realização de R3.

Ciclo de Aprendizado Inter-grupo do R3 - cont.

Etapa 3 - Compartilhamento do aprendizado: Considerando os artefatos que foram alterados para a realização da R3 (descritos na Epata 2 do Ciclo Intra-grupo), quanto ao formulário de levantamento do perfil (*Analyst Survey Form*), embora tenha sido modificado com a intenção de padronização, observou-se que o novo formato não ficou tão objetivo quanto o anterior no que diz respeito à caracterização do tempo de experiência profissional do participante relacionadas às perspectivas da

técnica. Quanto ao formulário de registro dos defeitos, a alteração efetuada facilitou, para os revisores, o registro dos tipos de defeitos. Outro artefato modificado foi a planilha de coleta de dados, que foram padronizadas entre os grupos participantes do projeto *Readers*, facilitando a coleta e a análise dos dados, possibilitando a geração imediata de alguns gráficos.

Etapa 4 - Harmonização do conhecimento: Devido à execução da R1 ter sido muito próxima à R3 essa etapa foi discutida após a realização de R1.

Etapa 5 - Criação/Evolução de corpo de conhecimento: Assim como na etapa anterior, essa etapa foi discutida após a realização de R1.

4.3.2 Quarto Ciclo de Aprendizado - R4

Ciclo de Aprendizado Inter-grupos do R4

Etapa 1 - Planejamento e coordenação das iniciativas: A realização de diversas replicações era objetivo inicial do projeto *Readers* e com a oportunidade de realização em ambiente industrial foi planejada a quarta replicação - R4.

Etapa 2 - Compreensão do experimento e dos pacotes de laboratório: Compreensão já adquirida em R1, R2 e R3.

Ciclo de Aprendizado Intra-grupo do R4

Etapa 1 - Definição dos objetivos do experimento: Os objetivos eram os mesmos das replicações R1, R2 e R3.

Etapa 2 - Preparação do experimento: O projeto experimental da quarta replicação seguiu a mesma sequência de técnicas e documentos da R1, executado em 2 dias consecutivos (turnos da manhã e tarde).

Ciclo de Execução do Experimento do R4

Essa replicação contou com apenas 15 participantes. O Grupo 1, composto por profissionais do CPqD, contou com apenas 6 participantes devido a problemas de troca de material e ausência de dois participantes durante a execução das técnicas. A distribuição

dos participantes dentro desse grupo para a aplicação da PBR foi: dois participantes utilizando a perspectiva do projetista, três utilizando o testador e um sendo usuário. O grupo 2 era composto por três profissionais do CPqD, três da EMBRAPA e três estudantes da pós-graduação da UNICAMP.

Ciclo de Aprendizado Intra-grupo do R4 - cont.

Etapa 3 - Execução de meta-análise: Os Gráficos 4.12 e 4.13 mostram o desempenho das quatro replicações considerando o documento ATM e o PG, respectivamente. Observa-se que para o documento ATM e a técnica *Checklist*, R4 obteve desempenho próximo ao de R3 e quanto à PBR, R4 foi melhor que R3 mas não alcançou a efetividade de R1 e R2. Considerando o documento PG e *Checklist*, R4 teve desempenho melhor que R2 e R3. Para a técnica PBR, R4 obteve o menor desempenho.

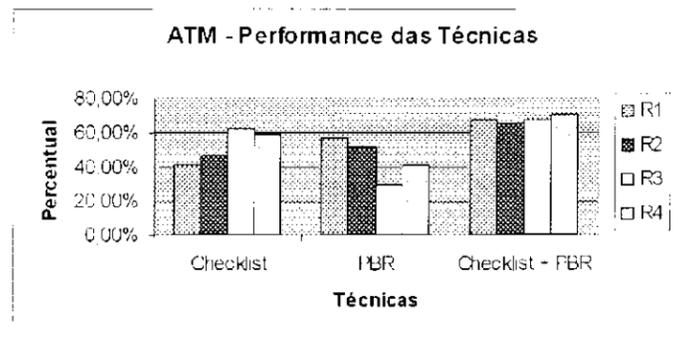


Figura 4.12: Desempenho das Quatro Replicações para o Documento ATM

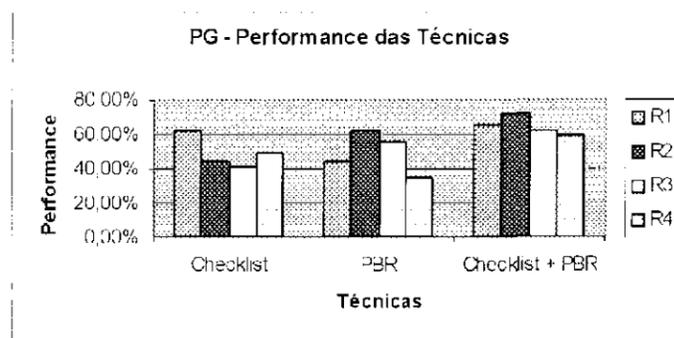


Figura 4.13: Desempenho das Quatro Replicações para o Documento PG

Um estudo estatístico de um maior volume de dados pode se tornar difícil e em alguns casos até inviável, como no teste de permutação dos grupos de participantes, abordagem utilizada nos estudos anteriores (Basili et al., 1996; Maldonado et al., 2004a).

devido crescimento do volume de dados (Maldonado et al., 2004c). A utilização de técnicas de visualização está sendo estudada para viabilizar a análise dos dados dos experimentos em conjunto.

Etapa 4 - Criação/Evolução do pacote de laboratório: A lista de defeitos dos documentos ATM e PG foram atualizadas com os novos defeitos descobertos em R3 e R4.

Ciclo de Aprendizado Inter-grupo do R4 - cont.

Etapa 3 - Compartilhamento do aprendizado: Nessa replicação foi constatado que a atividade dentro do ambiente da própria empresa pode propiciar algumas ocorrências não desejáveis durante um experimento. Houve várias solicitações aos participantes, fazendo com que estes tivessem que se ausentar da sala durante alguns períodos de tempo e mesmo solicitações que obrigaram o participante a abandonar o experimento totalmente, interferindo na análise dos dados e na validade interna, uma vez que os grupos ficaram desbalanceados.

Outro ponto importante é que, por se tratar de participantes mais experientes, a expectativa era de que a efetividade e eficiência fossem melhores do que nas três replicações anteriores. No entanto, observou-se justamente o contrário, ou seja, eles foram menos efetivos na aplicação de uma técnica mais procedimental, a PBR. No caso do *Checklist* o rendimento manteve-se na média das replicações R2 e R3.

Etapa 4 - Harmonização do conhecimento: Em vista das observações relatadas na etapa anterior, durante o *Workshop* do projeto *Readers* realizado em janeiro de 2002 destacou-se que o treinamento nas técnicas poderia ter uma grande influência no desempenho dos participantes, então um treinamento mais detalhado deveria ser considerado. Essa questão foi novamente discutida no *Workshop* realizado em julho de 2002, nos Estados Unidos, e foi decidido que o treinamento deveria abordar cada item separadamente, ou seja, deveria haver treinamento no domínio

das aplicações, nos modelos subjacentes, na taxonomia e nas próprias técnicas. O grau de detalhamento do treinamento seria determinado pelo nível de experiência dos participantes.

Etapa 5 - Criação/Evolução de corpo de conhecimento: Com o volume de dados que foi criado em decorrência das quatro replicações fortaleceu-se a idéia da utilização de recursos automatizados para análise dos dados, como a análise visual. Esta seria útil tanto para identificar/levantar hipóteses com base nos dados existentes quanto para realizar novas análises e meta-análises desses dados.

4.4 Considerações Finais

A realização de replicações de um experimento, principalmente em ambientes e culturas diversos, é essencial para se obterem mais dados que possam confirmar ou negar os resultados obtidos no estudo original e para compor uma base de conhecimento do objeto em estudo. As replicações R3 e R4, apresentadas neste capítulo, alimentaram a base de dados do experimento que estuda a efetividade e eficiência da técnica PBR em comparação à técnica *Checklist* e colaboraram para a melhoria do processo de experimentação, através do ajuste e validação dos materiais utilizados ou procedimentos seguidos no experimento e dos conhecimentos adquiridos e compartilhados com os grupos que participam do projeto de pesquisa, como foi relatado ao longo dos ciclos de aprendizagem inter-grupos e intra-grupo do EIP (*Experimentation Improvement Paradigm*). A realização da quarta replicação (R4) em ambiente industrial motivou tanto os replicadores (academia - USP/UFSCar) quanto a indústria (CPqD) para instanciar o pacote de laboratório das técnicas de leitura PBR, visando a transferência da técnica para a empresa. Essa experiência de transferência de tecnologia baseada em experimentação é descrita no Capítulo 5.

Transferência da Técnica PBR para Indústria por Meio do Processo de Experimentação

5.1 Considerações Iniciais

A integração indústria e academia é essencial ao desenvolvimento, validação e evolução de novas tecnologias, gerando uma base de conhecimento que fornece suporte à reutilização dessas tecnologias em diversos contextos e projetos. Nesse tipo de cooperação a academia atua como pesquisadora e agente de transferência de tecnologias e a indústria como agente de validação, aprimoramento e evolução das mesmas. Nesse contexto existe a preocupação em estabelecer as etapas necessárias para uma transferência tecnológica e em identificar como ocorre o compartilhamento de conhecimento durante essas fases. Essa preocupação é refletida em modelos de melhoria de processos, como o QIP (*Quality Improvement Paradigm*) (Basili et al., 1994) e o IDEAL (*Initiating, Diagnosing, Establishing, Acting & Learning*) (Gremba e Myers, 1997).

Neste capítulo descreve-se uma experiência de transferência da técnica de leitura PBR para um ambiente industrial por meio de experimentação. Na Seção 5.2 discutem-se

iniciativas de utilização do processo de experimentação como mecanismo de treinamento e adaptação de tecnologia e do ciclo de compartilhamento do conhecimento na transferência de tecnologias. Na Seção 5.3 descrevem-se os projetos pilotos realizados durante a experiência de transferência da técnica PBR na empresa e, na Seção 5.4, analisam-se esses estudos conforme o paradigma EIP e o modelo de compartilhamento de conhecimento. Na Seção 5.5 apresentam-se as contribuições obtidas com essa experiência em relação ao processo de transferência tecnológica.

5.2 Transferência de Tecnologia Utilizando Experimentação

A procura por novas tecnologias não é a principal preocupação das indústrias, mas sim a orientação de como usar a tecnologia existente. Esse é um dos maiores problemas associados à transferência tecnológica (Basili, 1985). Linkman e Rombach (1997) destacam três problemas em transferência de tecnologias de software para aplicações na indústria:

- Novas tecnologias são consideradas não adaptáveis às necessidades comerciais;
- A gerência e assistentes não são suficientemente convencidos dos benefícios da nova tecnologia para considerar o seu uso em projetos;
- Experiências de projetos passados não são reutilizadas em novos projetos, geralmente porque não houve esforço suficiente para registrar e demonstrar seus benefícios.

Para esses autores, uma maneira de amenizar esses problema é utilizar a experimentação para apoiar a transferência tecnológica. Sua aplicação pode ser um mecanismo eficaz e necessário para conduzir a apresentação de novas tecnologias em ambiente industrial, mostrando-se ser um componente tanto de pesquisa quanto educacional na engenharia de software. A experimentação torna o processo de aprendizado em engenharia de software fundamentado no método científico, o qual pode ser resumido nas seguintes etapas:

- Construção de modelos;
- Análise da corretude dos modelos por meio de teste e experimentação;

- Análise dos resultados de experimentos visando ao aprendizado, ao empacotamento do conhecimento e ao refinamento do modelo existente;
- Evolução do conhecimento baseada na experiência ao longo do tempo.

O trabalho experimental entre indústria e academia resulta em relevantes ganhos para ambos os lados. A indústria encontra nessa cooperação um universo de replicações, dados de pacotes de laboratório e artefatos de domínios genéricos que possibilitam comparar sua efetividade com a efetividade média obtida em outras experiências, motivando-a, conseqüentemente, para a utilização da tecnologia e levando-a também a compartilhar os resultados obtidos internamente.

Para a academia essa integração possibilita a validação externa dos resultados experimentais obtidos por uma nova tecnologia, o que permite a generalização destes, viabiliza a evolução dos pacotes de laboratório, inclusive introduzindo artefatos de domínios de aplicações reais e motiva a pesquisa de novas tecnologias.

As principais etapas identificadas para transferir uma tecnologia para uma indústria utilizando experimentação são:

1. Instanciação do pacote de laboratório no domínio alvo da organização;
2. Formação de replicadores;
3. Disseminação na organização e
4. Disseminação e troca de experiência com a comunidade.

Com essa visão foi planejado um trabalho de transferência tecnológica no CPqD (Centro de Pesquisa e Desenvolvimento em Telecomunicações) com o propósito de adaptar as técnicas de leitura PBR e de promover sua implantação de acordo com objetivos e tarefas específicos ao desenvolvimento de software e tipos específicos de documentos do contexto do domínio de aplicação da empresa.

No convênio efetuado com a empresa foi acordada a execução completa da primeira etapa do processo de transferência, relatada neste trabalho, dando também subsídios para as demais etapas.

A execução desse trabalho contemplou também alguns pontos identificados pelos pesquisadores do projeto *Readers* como necessários para melhorar o treinamento na técnica e

evoluir o pacote de laboratório, discutidos durante os *workshops* do grupo de acompanhamento do projeto. A relação do conhecimento do revisor quanto ao domínio de aplicação do documento a ser revisado com a efetividade por ele obtida, o grau de detalhamento do treinamento dos revisores, a distribuição dos defeitos ao longo do documento de requisitos e o vínculo dos defeitos com a questão que podem detectá-lo foram pontos destacados para análise. Para estudar esses pontos foi decidido analisar os defeitos do documento de requisitos considerando sua quantidade, tipo, distribuição no documento, dependência do domínio, grau de dificuldade em detecção e relação com as questões das técnicas que os detectam. Quanto ao treinamento foi decidido fazê-lo de forma mais detalhada, abordando treinamento na taxonomia, no domínio de aplicação, nos modelos subjacentes que apóiam a aplicação da técnica e na própria técnica. Para tanto, era necessário ter materiais em domínios diferentes aos já encontrados no pacote.

O planejamento para execução da etapa de instanciação do pacote de laboratório no domínio da empresa abordou esses pontos e consistiu em:

- a. Seleção e preparação do artefato para ser utilizado no pacote;
 - b. Elaboração da lista de defeitos;
 - c. Análise da distribuição dos defeitos pelo documento e caracterização dos defeitos;
 - d. Execução do projeto piloto 1 visando ao treinamento da equipe da empresa envolvida na transferência tecnológica;
 - e. Execução do projeto piloto 2 para verificar a aceitação da técnica por profissionais e para validar o pacote instanciado;
- a. Seleção e preparação do artefato:** Dentre os artefatos disponíveis para esse estudo, foi escolhido o documento de especificação de requisitos OPER do módulo de Operação (Apêndice A), de domínio específico da empresa, levando-se em consideração o tamanho do documento (8 páginas) e tempo de aplicação da técnica necessário durante o experimento, mantendo a conformidade com os documentos do pacote de laboratório existente. A preparação do documento consistiu em colocá-lo no padrão para documento de requisitos IEEE (IEEESTD830, 1998).
- b. Elaboração da Lista de Defeitos:** Uma lista de defeitos inicial foi elaborada para esse documento com base na comparação realizada entre a primeira versão (V_0) e

a última (V_1) existentes do documento de requisitos (Figura 5.1). Essas versões do documento de requisitos são provenientes de inspeções realizadas com técnicas próprias da empresa e alterações ocorridas pela evolução dos requisitos. Essa primeira lista foi composta por 20 defeitos e foi denominada de Lista de Defeitos Históricos (Apêndice C).

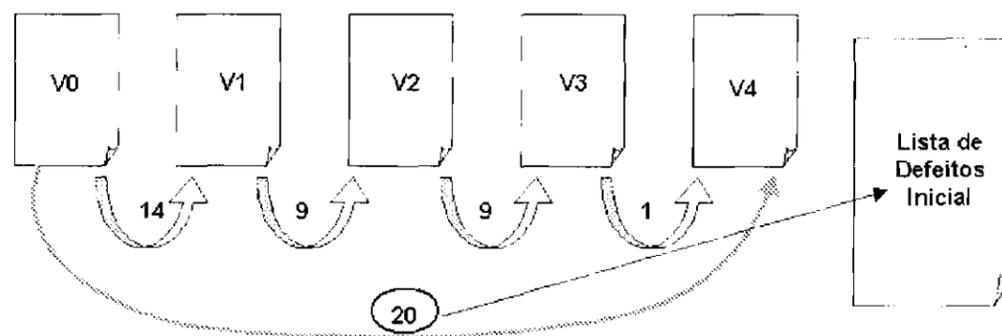


Figura 5.1: Elaboração da Lista de Defeitos Históricos

A taxonomia pela qual os defeitos foram classificados nos Projetos Pilotos 1 e 2, apresentada na Tabela 5.1, foi a que compõe o pacote de experimentação PBR, utilizada em todas as replicações anteriores do projeto *Readers*. Essa taxonomia foi conhecida pela empresa durante a execução de R1 em 2001 e adotada com algumas adaptações para uso nas inspeções *ad hoc*.

Tabela 5.1: Taxonomia do Pacote de Experimentação

Omission (O)	Omissão (O)
Ambiguous Information (AI)	Ambigüidade (A)
Inconsistent Information (II)	Informação Inconsistente (II)
Incorrect Fact (IF)	Fato Incorreto (FI)
Extraneous Information (EI)	Informação Estranha (IE)
Miscellaneous Defect (MD)	Defeitos Diversos (DD)

c. Análise da distribuição dos defeitos pelo documento e caracterização dos defeitos:

Os seguintes pontos foram analisados:

- distribuição dos defeitos no documento: os defeitos foram marcados no documento, diferenciados por tipo, possibilitando uma visão da sua distribuição, identificando as possíveis áreas de maior concentração de defeitos e os tipos de defeitos de maior ocorrência. (Apêndice B)
- caracterização dos defeitos: foi feita uma classificação dos defeitos quanto à dependência do domínio e ao grau de dificuldade em detecção e um vínculo de cada defeito com as questões das técnicas que possivelmente levam à sua identificação durante a revisão (Apêndice E) Esse estudo auxilia na caracterização do perfil necessário dos revisores para uma atividade de inspeção em um determinado domínio e na evolução da técnica, identificando possíveis adaptações das questões a esse domínio.

Na próxima seção descrevem-se os dois projetos pilotos, P1 e P2, referentes às etapas (d) e (e) respectivamente, do planejamento para instanciação do pacote de laboratório. Detalham-se o material utilizado, os perfis dos participantes, o projeto experimental e os resultados. Em seguida, analisam-se esses estudos com base nos ciclos do EIP, apresentados no Capítulo 3, Seção 3.8.2.1, no qual comentam-se as etapas realizadas durante os ciclos inter-grupos e intra-grupo.

5.3 Descrição dos Projetos Pilotos

5.3.1 Projeto Piloto 1 - P1

O Projeto Piloto 1 foi realizado com 2 dias de atividades na empresa, em agosto de 2002, e 5 dias de atividades na USP, em outubro de 2002.

Objetivo do estudo

1) Familiarização com o domínio de aplicação da empresa, por parte da equipe acadêmica:

2) Conhecimento de um pacote de laboratório e do processo de experimentação, por parte da equipe da empresa;

3) Utilização dos artefatos instanciados para o domínio da empresa visando à validação dos mesmos e à identificação de novos ajustes, se necessários.

Participantes

Os participantes desse projeto piloto estavam ligados ao processo de transferência das técnicas PBR para a empresa. Do lado da empresa, participaram 3 profissionais do grupo de pesquisa aplicada em qualidade de software e do lado acadêmico participaram 2 alunos de mestrado, do laboratório de engenharia de software da USP (LABES-ICMC/USP). Pelo formulário de levantamento do perfil dos participantes (*Analyst Survey Form*), a maioria classificou-se quanto à proficiência da língua (inglês), no item Leitura, em nível alto (*high*).

Material

Foram utilizados três documentos de especificações de requisitos do pacote de laboratório PBR (ABC *Video*, ATM e PG) e um documento de domínio específico da empresa (OPER). Devido ao caráter de treinamento do Projeto Piloto 1, todos os participantes aplicaram as duas perspectivas (Usuário e Testador) nos documentos ABC *Video* e ATM. No último documento a ser aplicado, os 2 participantes da empresa (participantes S1 e S4) que trabalharam na preparação do documento OPER aplicaram no documento de Parking Garage (PG) e os demais no documento de domínio específico OPER, como apresentado na Tabela 5.2.

Tabela 5.2: Documentos Utilizados no P1

	S1 (CPqD)	S2 (CPqD)	S3 (USP)	S4 (CPqD)	S5 (USP)
Treinamento	Gas Station	Gas Station	Gas Station	Gas Station	
	ABC Video	ABC Video	ABC Video	ABC Video	
Aplicação	ATM	ATM	ATM	ATM	-
	PG	PG	OPER	OPER	OPER

Procedimento

Projeto: O Projeto Piloto 1 seguiu o planejamento mostrado na Tabela 5.3.

Tabela 5.3: Projeto Piloto 1

Treinamento da Equipe Acadêmica no Domínio da Empresa	
1o. Dia	Conceitos Básicos de Rede Externa - Rede Metálica
	Visão Geral do SAGRE
	GAT: conceitos e demonstração
2o. Dia	SAGRE/Cad: projeto de uma rede metálica (demonstração)
	SAGRE/OPER: movimentação de facilidades (demonstração)
	SAGRE/OPER-FCT (Folha de Corte): alterações no projeto de uma rede ocupada (demonstração)
Treinamento da Equipe da Empresa no Pacote de Laboratório	
1o. Dia	Modelo subjacente <i>Error Guessing</i>
	Taxonomia de Erros
	Domínio do doc. req. <i>Gas Station</i>
	PBR - Parte Geral
	PBR - Testador
	Aplicação no doc. req. <i>Gas Station</i> - Testador
2o. Dia	Modelo subjacente Caso de Uso (UML)
	PBR - Usuário
	Aplicação no doc. req. <i>Gas Station</i> - Usuário
	<i>Feedback</i> dos defeitos
3o. Dia	Domínio do doc. req. ABC Vídeo
	Aplicação no doc. req. ABC Vídeo - Usuário
	Aplicação no doc. req. ABC Vídeo - Testador
	<i>Feedback</i> dos defeitos
Aplicação	
4o. Dia	Domínio no doc. req. ATM
	Aplicação no doc. req. ATM - Usuário
	Aplicação no doc. req. ATM - Testador
5o. Dia	Domínio do doc. req. PG
	Domínio do doc. req. OPER
	Aplicação no doc. req. OPER/PG - Usuário
	Aplicação no doc. req. OPER/PG - Testador

Treinamento: Iniciando o treinamento e integração de ambas as equipes ocorreu, primeiramente, um treinamento no domínio de aplicação específico da empresa para

a equipe da USP, realizado na empresa em agosto de 2002, com duração de 2 dias (Tabela 5.3). Foi programado o treinamento em inspeção e nas técnicas de leitura PBR para a equipe da empresa, utilizando o pacote de experimentação, descrito no projeto experimental (Tabela 5.3).

Resultados

Para posterior efeito comparativo entre o Projeto Piloto 1 e o Projeto Piloto 2, apresentam-se os resultados do documento ATM e OPER, comuns às duas aplicações e, para o Projeto Piloto 1 e documento de requisitos ATM, apresentam-se os grupos compostos apenas com os participantes da empresa.

Os resultados do Projeto Piloto 1 para os documentos de requisitos ATM e OPER estão sintetizados nas Tabelas 5.4 e 5.5, respectivamente. É importante ressaltar que os resultados obtidos pela perspectiva do Testador são de participantes que já haviam aplicado a perspectiva do Usuário no mesmo documento e que nem sempre repetiram os defeitos encontrados anteriormente, o que compromete a análise.

Tabela 5.4: Resultados ATM - Projeto Piloto 1

Perspectiva	Participante	Média de Defeitos Diferentes	Defeitos Encontrados	Falsos Positivos	Outras Discrepâncias
Usuário	S1	2.33	0	2	15
	S2		6	1	7
	S4		1	1	13
Testador	S1	3	2	2	11
	S2		3	1	7
	S1		6	2	27

Na Tabela 5.4 apresentam-se os resultados encontrados pelos indivíduos agrupados por perspectiva. A terceira coluna (Média de Defeitos Diferentes) representa a média dos defeitos diferentes encontrada pelo grupo de Usuários e pelo grupo dos Testadores. As demais apresentam o número de defeitos, o número de falsos positivos e o número de outras discrepâncias (as discrepâncias que não são defeitos da lista nem falsos positivos) encontrados por cada participante.

Na Tabela 5.5 apresentam-se os resultados encontrados pelos indivíduos também agrupados por perspectiva. Estes resultados estão atualizados com a última versão da lista

Tabela 5.5: Resultados OPER - Projeto Piloto 1

Perspectiva	Participante	Defeitos Históricos	Novos Defeitos	Total de Defeitos	Falsos Positivos	Outras Discrepâncias
Usuário	S2	1	6	7	3	2
	S3	2	3	5	1	0
	S5	1	3	4	1	3
Testador	S2	0	2	2	2	3
	S3	0	0	0	3	0
	S5	0	2	2	1	0

de defeitos, que contém 47 defeitos (20 defeitos históricos e 27 defeitos novos) e 12 falsos positivos (Apêndice C). O total dos defeitos encontrados pelos participantes está relacionado na quinta coluna (Total de defeitos), que consiste na soma da terceira com a quarta colunas (Defeitos Históricos e Novos Defeitos, respectivamente). Os Defeitos Históricos são aqueles que compõem a lista inicial de defeitos e os Defeitos Novos são os que foram detectados durante a execução dos projetos pilotos desse estudo. Dentre as discrepâncias relatadas nesse Projeto Piloto, 11 viraram novos defeitos e foram acrescentados na lista de defeitos do documento OPER, sendo 10 do tipo Omissão e 1 do tipo Informação Inconsistente.

5.3.2 Projeto Piloto 2 - P2

O Projeto Piloto 2 foi realizado em 4 dias de atividades na empresa, em fevereiro de 2003.

Objetivo do estudo

- 1) Validação das alterações efetuadas nos artefatos de domínio específico da empresa (documento de especificação e respectiva lista de defeitos);
- 2) Apresentação das técnicas para profissionais de desenvolvimento de software da empresa.

Participantes

Para aplicação do Projeto Piloto 2 contou-se com a participação de profissionais desenvolvedores de software e da área de requisitos de software da empresa. Os participantes

foram distribuídos nos grupos de Usuário e Testador, com 3 participantes cada um, de acordo com a análise do perfil feita por intermédio do Formulário de Levantamento de Perfil (*Analyst Survey Form*) preenchido pelos mesmos. Desse mesmo formulário extraiu-se o nível em proficiência da língua utilizada em muitos dos materiais do experimento (inglês) e a maioria classificou-se como Moderado.

Material

Foram utilizados os mesmos materiais do Projeto Piloto 1, com exceção do documento PG, pois nessa etapa apenas 3 documentos foram inspecionados (ABC Video, ATM e OPER).

Procedimento

Projeto: O projeto experimental do Projeto Piloto 2 seguiu as atividades descritas na Tabela 5.6. Devido à dificuldade em conseguir tempo disponível dos profissionais, as atividades ocorreram com certo intervalo entre os dias de treinamento e aplicações das técnicas.

Tabela 5.6: Projeto Piloto 2

	Grupo 1 - Usuário (3 revisores)	Grupo 2 - Testador (3 revisores)
Treinamento		
1o Dia	Taxonomia	
	Modelo subjacente Caso de Uso	Modelo Subjacente Error Guessing
	Domínio Doc. Req. Domínio Genérico Gas Station	
	PBR - Parte geral	
	PBR - Usuário	PBR - Testador
2o Dia	Domínio Doc. Req. Domínio Genérico ABC Video	
	Aplicação Doc. Req. ABC Video	
	Feedback dos defeitos	
Aplicação		
3o Dia	Treinamento Domínio Doc. Req. Domínio Genérico ATM	
	Aplicação Doc. Req. ATM	
4o Dia	Aplicação Doc. Req. específico OPER	
	Preenchimento do Questionário de Opiniões	

Tabela 5.9: Comparativo dos Resultados ATM - Projeto Piloto 1 e 2

	Defeitos Diferentes		Ocorrências		Falsos Positivos		Outras Discrepâncias	
	P1	P2	P1	P2	P1	P2	P1	P2
Usuário	2.33	2.33	2.33	2.33	1.33	1	11.67	5
Testador	3	1.67	3.67	1.67	1.67	0	15	4.67
Usuário + Testador	2	1.83	3	2	1.5	0.5	13.33	4.83

Os grupos utilizando a perspectiva do Usuário de ambos Projetos Pilotos obtiveram médias semelhantes, com exceção do item Outras Discrepâncias, no qual os Usuários do Projeto Piloto 1 relataram um número bem maior de discrepâncias nessa categoria.

Na Tabela 5.10 mostra-se a média dos resultados das Replicações R1, R2, R3 e R4, executadas no âmbito do Projeto *Readers*.

Tabela 5.10: Comparativo dos Resultados ATM - R1, R2, R3 e R4

		Usuário	Testador	Usuário + Testador
Defeitos Diferentes	R1	3	3	2.5
	R2	3.33	1.67	2.66
	R3	1.33	2	1.5
Ocorrências	R1	4	4	4
	R2	4	5.33	4.66
	R3	1.33	2.67	2
	R4	1	5.33	4.25
Falsos Positivos	R1	0.67	0.67	0.67
	R2	0.33	0.33	0.33
	R3	0.00	0.33	0.17
	R4	0.00	0.33	0.25
Outras Discrepâncias	R1	4	3.33	3.67
	R2	4	9	6.5
	R3	4.67	4.33	4.5
	R4	1	6	4.75

Na Tabela 5.11 mostra-se um comparativo entre os resultados do Projeto Piloto 1 e do Projeto Piloto 2, para o documento de requisitos OPER, através da média obtida pelo grupo de Usuários, pelo grupo de Testadores e pela junção das perspectivas Usuário e Testador. O Projeto Piloto 2 obteve melhores resultados (Defeitos Diferentes, Ocorrências

e Outras Discrepâncias) que o Projeto Piloto 1, tanto para a perspectiva do Usuário quanto para a do Testador.

Tabela 5.11: Comparativo dos Resultados OPER - Projeto Piloto 1 e 2

	Defeitos Diferentes		Ocorrências		Falsos Positivos		Outras Discrepâncias	
	P1	P2	P1	P2	P1	P2	P1	P2
	Usuário	3.33	5	5.33	7.67	1.67	0	1.67
Testador	1.33	4.33	1.33	6.67	2	1	1	4
Usuário + Testador	2.33	3.5	3.33	7.17	1.83	0.5	1.33	6.17

Foram acrescentados na Lista de Defeitos 16 novos defeitos procedentes do Projeto Piloto 2, sendo 11 do tipo Omissão, 2 do tipo Informação Incorreta, 1 do tipo Informação Inconsistente e 2 do tipo Defeitos Diversos. A lista de defeitos final ficou com 47 defeitos, sendo 20 históricos e 27 novos. Em relação à taxonomia a distribuição dos defeitos ficou: 23 defeitos de Omissão (*Omission* - O), 11 de Fato Incorreto (*Incorrect Fact* - IF), 8 de Informação Inconsistente (*Inconsistent Information* - II) e 5 do tipo Defeitos Diversos (*Miscellaneous Defect* - MD) (Tabela 5.12).

Tabela 5.12: Composição da Lista de Defeitos após Projetos Pilotos

	Tipos de Defeito				Total
	O	IF	II	MD	
Defeitos Históricos	2	9	6	3	20
Defeitos Novos	21	2	2	2	27
Total	23	11	8	5	47

Na Figura 5.2 mostram-se os defeitos encontrados por cada perspectiva (Usuário e Testador) no Projeto Piloto 1 e Projeto Piloto 2. No Projeto Piloto 1 não houve defeito encontrado em comum pelas duas perspectivas e os Usuários detectaram um maior número de defeitos diferentes. Como explicado anteriormente, a perspectiva do Testador foi aplicada após a do Usuário, pelos mesmos participantes, que nem sempre relatavam novamente as discrepâncias encontradas na aplicação anterior, influenciando no resultado obtido. No Projeto Piloto 2, 7 defeitos foram encontrados em comum pelas duas perspectivas. Os Usuários encontraram maior número de defeitos diferentes (8) do que os Testadores (6). Em relação à taxonomia, dos 15 defeitos detectados pelos Usuários, 11 foram de Omissão, 2 de Informação Inconsistente, 1 de Fato Incorreto e 1 do tipo

Defeitos Diversos. Os Testadores encontraram 6 defeitos de Omissão, 3 de Informação Inconsistente, 1 de Fato Incorreto e 3 do tipo Defeitos Diversos, no total de 13 defeitos.

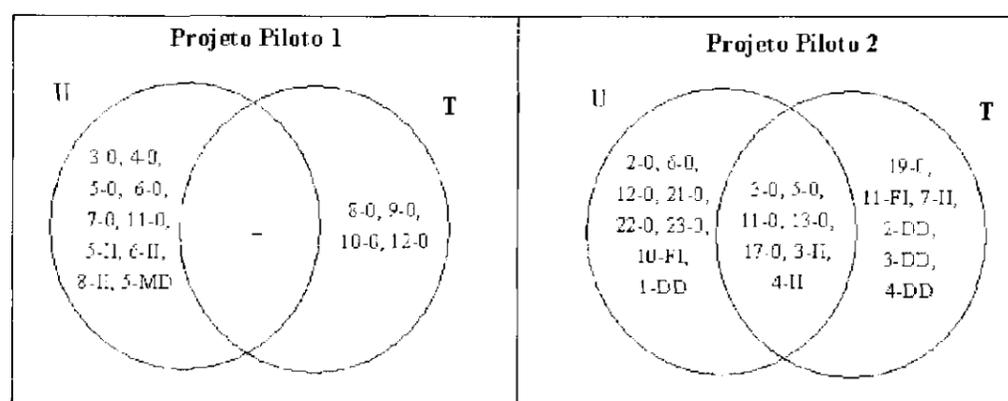


Figura 5.2: Defeitos Encontrados por Perspectiva - Projeto Piloto 1 e 2

Em relação aos defeitos históricos, na Tabela 5.13 mostram-se as ocorrências encontrados, nos Projetos Pilotos 1 e 2. Em ambos, os revisores relataram maior quantidade de defeitos novos e do tipo Omissão. Como apresentado na Tabela 5.14, no Projeto Piloto 2 detectou-se uma maior quantidade de defeitos distintos, 21 defeitos dos 47 existentes na lista, enquanto que no Projeto Piloto 1 foram encontrados 14. Dos 23 defeitos do tipo Omissão, no Piloto 2 foram encontrados 12 e no Piloto 1 foram encontrados 10. O tipo Informação Incorreta só foi encontrado no Projeto Piloto 2, sendo detectados os 2 defeitos existentes. Também apenas nesse projeto piloto foram encontrados os 3 defeitos históricos do tipo Defeitos Diversos.

Dos 20 defeitos históricos que constam na lista de defeitos, 8 foram detectados novamente durante a aplicação dos estudos pilotos (2 pelo Projeto Piloto 1 e 6 pelo Projeto Piloto 2). Um fator de provável influência nesse resultado é o fato da revisão ter sido feita bem próxima ao treinamento, sem um período de maturação das técnicas. Outro ponto a ser destacado é o grau de dependência do domínio e de dificuldade dos defeitos históricos. 10 dos 20 defeitos históricos são totalmente dependentes do domínio. Quanto ao grau de dificuldade, 10 defeitos são considerados muito difíceis de serem detectados por revisores que não têm conhecimento do domínio (Apêndice E). Um ponto positivo observado é a descoberta de 27 defeitos novos, em duas revisões, o que corresponde quase à mesma proporção dos defeitos históricos.

Tabela 5.13: Ocorrência de Defeitos nos Projetos Pilotos

	Projeto Piloto 1				
	O	IF	II	MD	
Defeitos Históricos	0	0	4	0	4
Defeitos Novos	14	0	1	1	16
Total	14	0	5	1	20
	Projeto Piloto 2				
	O	IF	II	MD	
Defeitos Históricos	1	0	6	3	10
Defeitos Novos	29	2	1	1	33
Total	30	2	7	4	43

Tabela 5.14: Total de Defeitos Diferentes Detectados nos Projetos Pilotos

	Projeto Piloto 1				
	O	IF	II	MD	
Defeitos Históricos	0	0	2	0	2
Defeitos Novos	10	0	1	1	12
Total	10	0	3	1	14
	Projeto Piloto 2				
	O	IF	II	MD	
Defeitos Históricos	1	0	2	3	6
Defeitos Novos	11	2	1	1	15
Total	12	2	3	4	21

Os resultados obtidos nos dois projetos pilotos motivaram a empresa na utilização de técnicas que auxiliam a inspeção de documentos de requisitos e aumentaram a preocupação com a fase de elaboração desses documentos, além de manter o interesse no prosseguimento de trabalhos integrados com a academia e na troca de conhecimento com a comunidade. A continuidade do trabalho internamente à empresa e a publicação de resultados ratificam esses pontos (Pagliuso et al., 2003).

5.4 Etapas da Transferência das Técnicas Conforme o Paradigma EIP

5.4.1 Compartilhamento do Conhecimento no Processo de Transferência Tecnológica

A Figura 5.3 representa a instanciiação do modelo de compartilhamento do conhecimento em engenharia de software experimental, abordado na Seção 3.8.1.1 do Capítulo 3, para o processo de transferência tecnológica abordado neste trabalho.

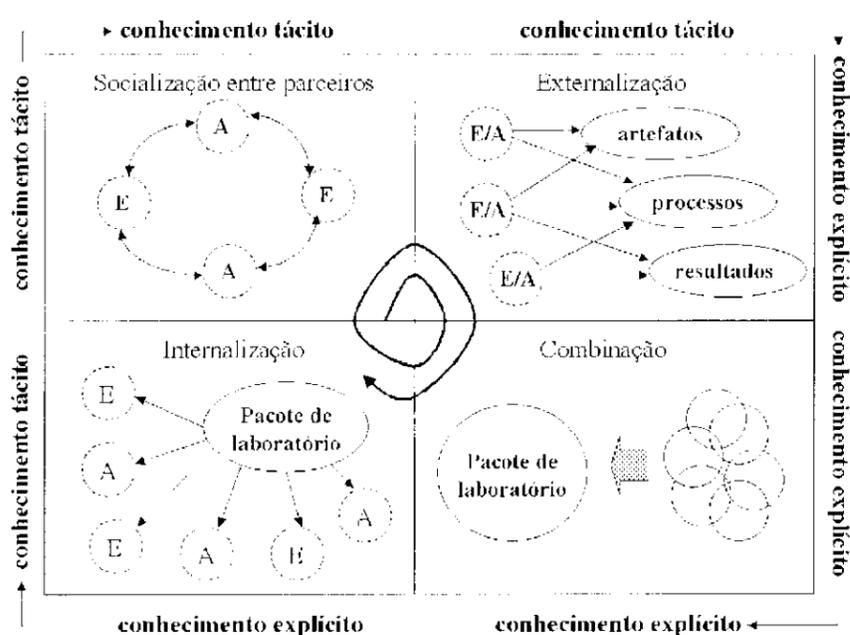


Figura 5.3: Compartilhamento do Conhecimento no Processo de Transferência Tecnológica (adaptado de (Mendonça et al., 2004?))

A socialização durante o processo de transferência de tecnologia ocorre entre as equipes parceiras compartilhando o conhecimento tácito: de um lado o conhecimento da tecnologia (academia) e de outro o conhecimento do domínio e das atividades do local a receber a nova tecnologia (empresa) (Figura 5.3, onde 'A' é Academia e 'E', Empresa). Na externalização do conhecimento a tecnologia é transmitida por meio de treinamentos e de pacotes de

laboratório, o que facilita demonstrar a efetividade da tecnologia a ser adotada e motivar a sua implantação. Na fase de combinação, o pacote de laboratório é instanciado para o domínio da empresa (processos, técnicas e artefatos são adaptados). E a internalização ocorre pela aplicação de experimentos com esse pacote de laboratório.

Analisando as atividades realizadas para transferência das técnicas PBR utilizando experimentação sob a ótica do Paradigma de Melhoria da Experimentação (EIP), observa-se um ciclo de aprendizado sob duas perspectivas: o aprendizado do processo de experimentação e das técnicas do pacote de laboratório por parte da equipe da empresa e o aprendizado do domínio de aplicação utilizado na empresa e do processo de instanciação do pacote e transferência de tecnologia, por parte da equipe da universidade (Figura 5.4). Devido à essa característica, o compartilhamento do aprendizado entre as duas equipes foi necessário ao longo de todos os ciclos e etapas, e não caracterizado apenas na etapa 5 do ciclo intra-grupos (compartilhamento e consolidação do aprendizado com outros grupos). Essa cooperação certamente contribui para a consolidação do conhecimento no nível da comunidade e redes de cooperação como o ISERN (ISERN, 2003).

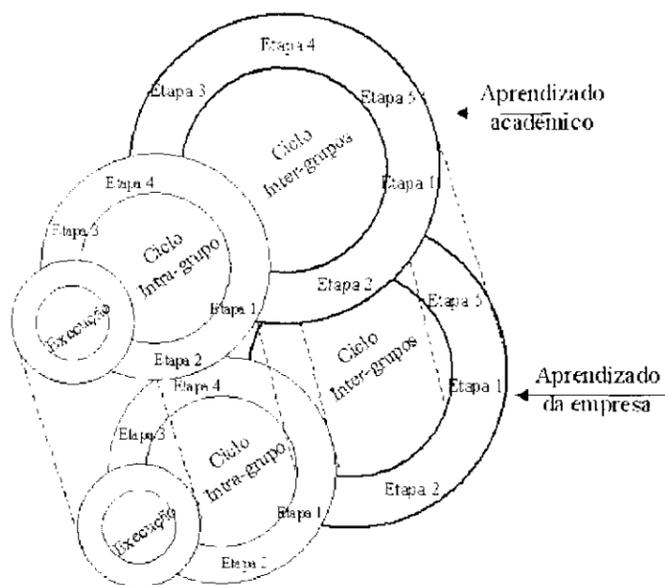


Figura 5.4: Ciclo de Aprendizagem Durante Processo de Transferência (adaptado de (Mendonça et al., 2004?))

5.4.2 Primeiro Ciclo de Aprendizado

Ciclo de Aprendizado Inter-Grupos

Etapa 1 - Planejamento e coordenação das iniciativas: Foi estabelecido o convênio entre o LABES/ICMC-USP e o CPqD, tendo como objetivo instanciar o pacote de laboratório PBR para o domínio de aplicação da empresa visando a transferência da técnica para a empresa. Planejar-se-então as atividades que constavam de seleção de um artefato, preparação do artefato e instanciação do pacote e execução dos projetos pilotos P1 e P2, cujos principais objetivos eram dar treinamento da equipe da empresa dando subsídios à instanciação do pacote de laboratório e validação desse pacote com desenvolvedores da empresa, respectivamente.

Etapa 2 - Compreensão do experimento e dos pacotes de laboratório: Para realizar a instanciação das técnicas na empresa decidiu-se compor um pacote de laboratório contendo um documento de requisitos de domínio específico da empresa, como fator motivador para a análise da efetividade da técnica nesse domínio. Para a equipe acadêmica poder avaliar os artefatos que iriam fazer parte do pacote precisaria conhecer o domínio de aplicação da empresa. Isso foi feito de acordo com o planejamento apresentado na Tabela 5.3. Do ponto de vista da equipe da empresa, era necessário que eles conhecessem as técnicas, o pacote de laboratório existente e o processo de experimentação, que ocorreu justamente com a realização do Projeto Piloto 1, conduzido pela equipe acadêmica.

Essas duas etapas do ciclo Inter-grupos estão relacionadas à etapa de socialização do modelo de compartilhamento do conhecimento apresentado na Figura 5.3, Seção 5.4.1. As atividades consistem em troca de conhecimento tácito entre a equipe acadêmica e a equipe da empresa.

Ciclo de Aprendizado Intra-Grupo

Etapa 1 - Definição dos objetivos do experimento: O Projeto Piloto 1 tinha por objetivo a compreensão do domínio de aplicação da empresa, por parte da equipe acadêmica e o conhecimento da técnica PBR e do pacote de laboratório por parte da equipe da empresa. Dentre as etapas de transferência tecnológicas, esse projeto piloto colaborou na etapa de formação de replicadores, uma vez que a equipe da empresa treinada poderia tornar-se futuros replicadores.

Etapa 2 - Preparação do experimento: O Projeto Piloto 1 seguiu o planejamento mostrado na Tabela 5.3, da Seção 5.3.1. Alguns participantes do Projeto Piloto 1 eram da empresa e outros do meio acadêmico, com o objetivo de que os primeiros utilizassem a técnica em documentos de especificação de requisitos fora de seu domínio de aplicação e os segundos dessem maior ênfase nos artefatos do domínio específico da empresa. Quanto aos artefatos utilizados, adicionou-se ao pacote um documento de especificação de requisitos do domínio da empresa, sendo necessárias a adequação do mesmo, colocando-o no padrão IEEE, e a elaboração de uma lista de defeitos inicial, conforme descrito no item Material da Seção 5.3.1. Esse novo artefato colaborou para a evolução do pacote de laboratório utilizado nos estudos do projeto *Readers*, atendendo a demanda de acrescentar documentos de uso real.

Ciclo de Execução do Experimento A execução do Projeto Piloto 1 ocorreu em outubro de 2002, durante 5 dias. Assim como nas outras replicações, durante a revisão de cada documento os participantes registraram todos os problemas encontrados por eles. Esses problemas relatados, chamados de discrepâncias, foram analisados com base na lista de defeitos existente para cada um dos documentos de requisitos. As discrepâncias que correspondiam a defeitos da lista de defeitos foram sintetizadas em planilhas de defeitos encontrados e as outras foram analisadas e classificadas em falsos positivos, não defeitos ou possíveis novos defeitos, o que contribuiu para evoluir a lista de defeitos do documento de requisitos OPER.

Os treinamentos ocorridos e a utilização do pacote de laboratório durante essa etapa representam a transformação do conhecimento tácito em conhecimento explícito relativa à etapa de externalização do modelo de compartilhamento do conhecimento (Figura 5.3 da Seção 5.4.1).

Ciclo de Aprendizado Intra-Grupo - continuação

Etapa 4 - Execução de meta-análise: Nesse trabalho de instanciação das técnicas o principal foco era no documento de requisitos do domínio específico da empresa. O Projeto Piloto 1 foi a primeira aplicação utilizando esse documento. Novas aplicações são necessárias para melhor caracterização dos defeitos e estudo de novas hipóteses para analisar relações da efetividade das técnicas com a dependência do domínio atribuída aos defeitos, com o nível de conhecimento dos revisores em relação ao domínio e com os tipos de defeitos mais descobertos.

Etapa 5 - Criação/Evolução do pacote de laboratório: Após a execução do Projeto Piloto 1, a lista de defeitos foi atualizada com os novos defeitos detectados pelos revisores. Os defeitos da lista foram diferenciados em Defeitos Históricos e Defeitos Novos. Outras mudanças efetuadas na lista foram a reorganização da mesma, agrupando os defeitos pela taxonomia, e a inserção de outros defeitos históricos em decorrência da inserção desses defeitos no documento de especificação de requisitos. Esses outros defeitos históricos inseridos procederam da comparação entre as versões intermediárias do documento de requisitos OPER (da V_0 à V_1), nas quais alguns defeitos foram inseridos involuntariamente, ou foram descobertos como consequência das atividades de inspeção *ad hoc* pelas quais ele passou ou foram alterações ocorridas pela própria evolução dos requisitos. A nova lista resultante e o documento de requisitos acrescidos desses defeitos foram preparados para serem utilizados no Projeto Piloto 2. Ressalta-se aqui a dificuldade de ambas as equipes em estabelecerem os defeitos da lista devido à pouca familiaridade tanto da equipe acadêmica com o domínio de aplicação, quanto da equipe da empresa, com o discernimento do que é um defeito e um falso positivo dentro da taxonomia adotada. Essa etapa corresponde à etapa de Combinação do modelo de compartilhamento do conhecimento (Figura 5.3 da Seção 5.4.1), na qual o pacote é instanciado agregando conhecimentos específicos da empresa.

Ciclo de Aprendizado Inter-Grupos - continuação

Etapa 3 - Compartilhamento do aprendizado: Foi decidido pelas duas equipes que deveria ser feita uma análise mais criteriosa dos defeitos que compunham a lista de defeitos do documento OPER e uma análise das questões das técnicas, associando-as aos defeitos que elas levavam o revisor a detectar. Além disso, os defeitos deveriam ser analisados e classificados quanto ao grau de dependência do domínio e quanto ao grau de dificuldade em detecção. Ambas as classificações são importantes para determinar eventual estratégia de alocação de recursos humanos para a atividade de revisão.

Etapa 4 - Harmonização do conhecimento: Foi feita uma classificação inicial dos defeitos quanto ao grau de dependência do domínio e ao grau de dificuldade em detecção e uma análise do vínculo do defeito com as questões que podem detectá-lo

(Apêndice F), com a intenção de ser revista e, se necessário ajustada, após a execução de outros experimentos com o documento de requisitos.

Classificação quanto ao grau de dependência do domínio

Analisando-se os defeitos do documento de requisitos OPER observou-se que alguns defeitos são muito dependentes do domínio, ou seja, requerem conhecimento do domínio por parte do revisor para que possam ser detectados. Para analisar essa dependência, foi definida uma classificação para os defeitos que consiste em:

- Completamente dependente do domínio;
- Fortemente dependente do domínio;
- Fracamente dependente do domínio;
- Independente do domínio.

Classificação quanto ao grau de dificuldade em detecção

Dependendo do nível de conhecimento do revisor no domínio, cada defeito tem seu grau de dificuldade para ser detectado. Definiu-se que cada defeito deve ser classificado quanto ao grau de dificuldade em relação a revisores com conhecimento no domínio e a revisores sem conhecimento no domínio. A classificação consiste em:

- Muito difícil;
- Difícil;
- Fácil;
- Muito fácil.

Vínculo do defeito com as questões que podem detectá-lo

Foi avaliada a relação entre as questões constantes nas técnicas de leitura do pacote utilizado (perspectivas do Usuário e do Testador) e os defeitos da lista visando a determinar que questão, ou questões, que levam a detectar cada defeito. Para cada par Defeito/Questão foram analisadas a ordem de adequação da questão e a

probabilidade dela levar a detectar aquele defeito. A classificação da probabilidade de detecção foi definida em:

- Muito Provável;
- Provável;
- Pouco Provável;
- Improvável.

Etapa 5 - Criação/Evolução do corpo de conhecimento: Como o Projeto Piloto 2 já estava programado, foi decidido congelar uma versão do pacote de laboratório para aplicá-lo nesse projeto, enquanto que, paralelamente, era dada a continuidade nos trabalhos de evolução da lista de defeitos e caracterização dos defeitos.

5.4.3 Segundo Ciclo de Aprendizado

Ciclo de Aprendizado Inter-Grupos

Etapa 1 - Planejamento e coordenação das iniciativas: No cronograma de atividades do convênio LABES/ICMC-USP e CPqD já foi planejada a execução do Projeto Piloto 2 com profissionais da área de desenvolvimento da empresa.

Etapa 2 - Compreensão do experimento e dos pacotes de laboratório: A condução do Projeto Piloto 2 também foi feita pela equipe do LABES/ICMC-USP, com conhecimento prévio no processo de experimentação e no pacote de laboratório resultante das etapas do Projeto Piloto 1.

Ciclo de Aprendizado Intra-Grupo

Etapa 1 - Definição dos objetivos do experimento: A execução do Projeto Piloto 2 teve por objetivo avaliar o pacote instanciado e a assimilação da técnica PBR por um grupo de profissionais de desenvolvimento da empresa, familiarizados com o domínio de aplicação. A condução desse projeto piloto foi acompanhada pela equipe da empresa responsável pela transferência tecnológica, atividade que faz parte da etapa de formação de replicadores.

Etapa 2 - Preparação do experimento: Uma das grandes dificuldades dessa etapa foi a falta de disponibilidade de tempo dos participantes do Projeto Piloto 2. Por não ser possível ter dedicação exclusiva dos mesmos por vários turnos seguidos, as atividades foram realizadas com intervalos entre os dias de treinamento e aplicações, como está apresentado na Tabela 5.6, da Seção 5.3.2. Basili et al. (1996) ressaltam que para se obter maior validade interna nos resultados de qualquer replicação deve-se usar uma caracterização mais uniforme.

Ciclo de Execução do Experimento Ocorreu a desistência de alguns participantes durante a execução de Projeto Piloto 2, devido a solicitações dos mesmos para outras atividades na empresa. Esses participantes foram substituídos por outros profissionais e os grupos foram novamente arranjados. Em estudos em ambiente industrial é necessário o comprometimento da empresa para minimizar a ocorrência de problemas que possam alterar o planejamento. A execução de Projeto Piloto 2 corresponde à etapa de Internalização do modelo de compartilhamento do conhecimento (Figura 5.3 da Seção 5.4.1), na qual a utilização do pacote de laboratório por parte de profissionais da empresa representa a transformação do conhecimento explícito constante no pacote em conhecimento tácito por parte desses profissionais.

Ciclo de Aprendizado Intra-Grupo - continuação

Etapa 4 - Execução de meta-análise: Foi realizada uma análise comparando os resultados dos dois Projetos Pilotos com os resultados das replicações anteriores do experimento PBR, com relação aos documentos de domínio genérico, na qual observou-se que ambos os Projetos Pilotos obtiveram resultados na média das outras replicações. Analisando-se os resultados dos Projetos Pilotos com relação ao documento de domínio específico reforçou o interesse em estudar a relação entre o conhecimento do domínio por parte do revisor e os defeitos por ele detectados quanto ao tipo de defeito e quanto à dependência do domínio.

Etapa 5 - Criação/Evolução do pacote de laboratório: Após a aplicação do Projeto Piloto 2 foi feita a avaliação das discrepâncias relatadas e novos defeitos identificados. Foram geradas uma nova lista de defeitos, uma lista de falsos positivos e uma lista de não defeitos. Muitas dessas alterações foram efetuadas na lista de defeitos, além do acréscimo de novos defeitos após os projetos pilotos, devido

ao “amadurecimento” das equipes tanto no domínio de aplicação do documento de requisitos utilizado, quanto na classificação das discrepâncias em defeitos, não defeitos e falsos positivos. Maiores detalhes dessas alterações estão descritas no documento Histórico da Lista de Defeitos (Apêndice D). Assim como no Projeto Piloto 1, uma das grandes dificuldades desta etapa está relacionada à lista de defeitos, ou seja, estabelecer o que é defeito, falso positivo e não defeito.

Ciclo de Aprendizado Inter-Grupos- continuação

Etapa 3 - Compartilhamento do aprendizado: Assim como foi discutido após o Projeto Piloto 1, no Projeto Piloto 2 decidiu-se fazer a avaliação dos novos defeitos detectados de acordo com a classificação relacionada à dependência do domínio, dificuldade de detecção e probabilidade de serem detectados pelas questões das técnicas. Um outro ponto discutido foi a taxonomia. Algumas discrepâncias relatadas nos estudos pilotos indicaram a necessidade de um tipo de defeito para classificar informação em local errado no documento de requisitos, devendo ser acrescentado na taxonomia adaptada pela empresa.

Etapa 4 - Harmonização do conhecimento: Foi executada a análise dos defeitos quanto ao grau de dependência do domínio e ao grau de dificuldade em detecção para os novos defeitos identificados no Projeto Piloto 2. Dando continuidade à personalização das técnicas PBR, os novos defeitos também foram classificados quanto à probabilidade de detecção pelas questões das técnicas.

Etapa 5 - Criação/Evolução do corpo de conhecimento: Mantém-se o interesse na interação entre a indústria e a academia tendo em vista os benefícios mútuos obtidos nessa experiência e na realização de outros experimentos com o documento de requisitos OPER, tanto na área acadêmica quanto na indústria, para validar a análise e classificação dos defeitos e para obter maior quantidade de dados.

5.5 Síntese da Experiência Adquirida

Com a experiência obtida durante a transferência da técnica PBR para o contexto industrial, identificaram-se as etapas gerais para transferir uma tecnologia utilizando o processo de experimentação, apresentadas na Seção 5.2, e definiu-se o esquema de atividades abaixo,

representando todas as fases de um processo para transferência tecnológica utilizando pacotes de laboratório.

1. Instanciamento das técnicas

- a) Treinamento do domínio de aplicação;
- b) Treinamento das técnicas a serem utilizadas;
- c) Seleção/elaboração dos artefatos do domínio da aplicação a serem utilizados no processo de experimentação;
- d) Realização de um estudo piloto com base do pacote de experimentação;
- e) Avaliação dos resultados do estudo piloto;
- f) Avaliação e ajuste, se necessário, do pacote de laboratório ao contexto da indústria;
- g) Realização de outro estudo piloto para avaliação do pacote instanciado.

2. Formação de replicadores

- a) Estudo do pacote de laboratório e da teoria utilizada no treinamento das técnicas;
- b) Estudo do processo de experimentação;
- c) Observação da condução de uma replicação após aquisição dos conhecimentos dos passos a e b;
- d) Condução de um projeto piloto objetivando dominar os conceitos fundamentais e o conhecimento tácito e também como mecanismo de garantia da conformidade do processo da experimentação (recomendado por Shull et al. (2004), que ressaltam que projetos pilotos também auxiliam no processo de internalização e socialização.);
- e) Avaliação do projeto piloto, podendo necessitar reforçar pontos dos passos a e b;
- f) Condução de uma replicação do experimento sob a observação de replicadores experientes.

3. Disseminação na empresa

- a) Replicação do experimento com objetivo de treinamento de outras equipes;

- b) Aplicação das técnicas em projetos reais;
- c) Coleta de dados e análise dos dados.

4. Disseminação e troca de experiência com a comunidade

As análises dos defeitos efetuadas para classificá-los quanto ao grau de dependência do domínio e quanto ao grau de dificuldade em detecção, os resultados obtidos no Projeto Piloto 1, que contou com alguns participantes sem conhecimento no domínio e outros com conhecimentos básicos no domínio, e os resultados do Projeto Piloto 2, no qual todos participantes tinham conhecimento no domínio, suscitou novos questionamentos. Novas hipóteses podem ser formuladas para investigar algumas relações, como as identificadas a seguir:

- relação entre a efetividade do revisor em detectar defeitos dependentes do domínio e o seu conhecimento nesse domínio:
 - a) Revisores que conhecem bem o domínio descobrem defeitos dependentes do domínio?
 - b) Revisores sem conhecimento do domínio descobrem defeitos dependentes do domínio?
- relação entre a efetividade do revisor em detectar defeitos de determinado tipo e o seu conhecimento no domínio:
 - a) Revisores que não conhecem o domínio descobrem mais defeitos de omissão?

5.6 Considerações Finais

A transferência de uma nova tecnologia para a indústria requer grande interação entre a equipe receptora - equipe da indústria familiarizada com o domínio de aplicação, e a equipe transmissora, que possui o conhecimento da tecnologia. Durante o trabalho de transferência da técnica de leitura PBR efetuado na empresa e apresentado neste capítulo essa interação foi analisada por meio do modelo de compartilhamento do conhecimento (ESKM) e pelo paradigma de melhoria da experimentação (EIP), pois utilizou-se o processo de experimentação como veículo de treinamento e difusão. Dentre as etapas

identificadas para transferência tecnológica por meio de pacotes de laboratório, a primeira etapa - Instanciação do pacote de laboratório no domínio alvo da organização - foi planejada e executada dentro de um convênio de cooperação firmado entre o CPqD e o ICMC/USP. As atividades para instanciação do pacote consistiram na seleção de um artefato, na preparação deste e instanciação do pacote de laboratório e na execução de dois projetos pilotos, sendo que o primeiro visava ao treinamento da equipe da empresa envolvida na transferência tecnológica e o segundo visava a verificação e aceitação da técnica por profissionais conhecedores do domínio de aplicação. Os resultados obtidos nessa experiência estimulam a cooperação entre a indústria e a academia com benefícios para ambas.

Conclusões e Trabalhos Futuros

Neste trabalho foram apresentadas duas replicações do experimento com a técnica de leitura PBR conduzidas pelo ICMC/USP e UFSCar no âmbito do projeto *Readers*, sendo que uma foi em ambiente acadêmico e a outra em ambiente industrial. A variação do contexto de aplicação permite o estudo de variáveis dependentes, como cultura e experiência dos participantes. Observou-se que os resultados da replicação conduzida em ambiente industrial foram aquém dos resultados das replicações em ambiente acadêmico, assim como dentre as que foram conduzidas nas universidades, a que foi realizada com estudantes de pós-graduação (R3) também obteve resultados menos efetivos que as replicações com estudantes de graduação (R1 e R2), o que leva a considerar que quanto maior a experiência do participante (revisor) menor a tendência por seguir uma nova técnica sistemática, ou seja, os participantes tendem a aplicar seu próprio conhecimento (*ad hoc*). Isso ressalta a necessidade de se avaliar o quanto cada participante compreendeu e seguiu a técnica para poder realmente avaliar o desempenho da mesma e adaptar o seu treinamento para diversos perfis de revisores.

A cooperação entre a indústria e a academia possibilitou uma experiência de transferência da técnica de leitura PBR para a empresa, utilizando o processo de experimentação como veículo de treinamento e difusão da técnica. Do ponto de vista da empresa, essa

experiência motivou a utilização da técnica para auxiliar a fase de inspeção de documentos de especificação de requisitos, além de aumentar a preocupação com a qualidade durante a fase de elaboração desses documentos. Para a academia, esse trabalho colaborou na evolução do pacote de laboratório da técnica, incluindo um artefato de domínio específico e de uso real nos estudos, caracterizando os defeitos quanto à dependência de domínio, ao grau de dificuldade em detecção e ao vínculo que possuem com as questões das técnicas que os detectam, analisando a distribuição dos defeitos pelo documento e na validação do treinamento mais detalhado.

Essa experiência de transferência tecnológica foi discutida sob o aspecto do Paradigma de Melhoria da Experimentação (EIP - *Experimentation Improvement Paradigm*) e do modelo de compartilhamento do conhecimento no contexto de experimentação (EKSM - *Experimentation Knowledge Sharing Model*). Algumas dificuldades foram encontradas durante a fase de socialização no nível intra-grupos devido à visões, prioridades e prazos diferentes entre a área industrial e a acadêmica. Na fase de internalização do conhecimento, ressalta-se a importância da utilização de pacotes de laboratório por profissionais durante a aquisição de novas tecnologias, uma vez que essa prática representa a transformação do conhecimento explícito constante no pacote em conhecimento tácito por parte desses profissionais.

Proveniente dessa experiência pôde-se abstrair as fases necessárias e suas atividades para efetuar um trabalho de transferência tecnológica utilizando experimentação e suscitar novos pontos para investigação. A realização desse trabalho propiciou a experiência na instanciamento de pacotes de laboratório, pois o pacote utilizado nos experimentos teve que ser adequado ao contexto da empresa para a qual foi feita a transferência, além de ter ressaltado a contribuição da experimentação na perspectiva da cooperação entre a indústria e a academia, possibilitando a validação de novas tecnologias, a evolução de pacotes de laboratório e a obtenção de resultados de projetos de uso real.

6.1 Trabalhos Futuros

As seguintes atividades são vistas como continuidade a este trabalho:

- execução das etapas previstas para o processo de transferência tecnológica que não foram tratadas completamente no convênio de cooperação: formação de replica-

dores, disseminação na organização e disseminação e troca de experiência com a comunidade;

- localização dos artefatos do pacote de laboratório utilizado nos experimentos para a língua portuguesa e cultural local;
- planejamento e execução de replicações, tanto no contexto industrial quanto acadêmico, utilizando o pacote instanciado para a empresa para avaliar sua qualidade e validar as classificações dos defeitos da lista de defeitos existente. Fazer o mesmo utilizando o pacote de laboratório localizado para a língua e cultura local;
- automatização do processo de experimentação trazendo maior rigor na coleta de dados durante a execução do experimento, facilitando a análise dos dados e viabilizando a execução de mais replicações;
- planejamento e execução de estudos que investiguem as novas hipóteses apresentadas.

Referências Bibliográficas

AMARAL, E. A. G. G. *Empacotamento de Experimentos em Engenharia de Software*. Dissertação de Mestrado, COPPE/UFRJ, 2003.

ANDRIOLE, S. J. *Software Validation, Verification, Testing and Documentation*. Petrocelli Books, 1986.

ARISHOLM, E.; SJOBERG, D. I. K.; CARELIUS, G. J.; LINDSJORN, Y. [†]SESE - an Experiment Support Environment for Evaluating Software Engineering Technologies. p. 81-98. 2002.

Disponível em <http://www.itu.dk/people/kasper/NWPER2002/papers/arisholm.pdf> (Acessado em 03/10/2003)

BARROS NETO, B.; SCARMINIO, I. S.; BRUNS, R. E. *Como Fazer Experimentos: Pesquisa e Desenvolvimento na Ciência e na Indústria*. Editora da UNICAMP, 2001.

BASILI, V. *Quantitative Evaluation of Software Methodology*. Technical report TR-1519, University of Maryland, 1985.

Disponível em <http://www.cs.umd.edu/projects/SoftEng/ESEG/papers/83.29.pdf>

BASILI, V. The Maturing of the Quality Improvement Paradigm in the SEL. 1994.

BASILI, V.; GREEN, S.; LAITENBERGER, O.; LANUBILE, F.; SHULL, F.; SORUMGARD, S.; ZELKOWITZ, M. Lab Package for the Empirical Investigation of Perspective-Based Reading. [On-line], University of Maryland.

Disponível em http://www.cs.umd.edu/projects/SoftEng/ESEG/manual/pbr_package/manual.h%tml (Acessado em 08/04/2002)

BASILI, V. R.; CALDIERA, G.; MCGARRY, F.; PAJERSKY, R.; PAGE, G.; WALIGORA, S. The Software Engineering Laboratory - An Operational Software Experience Factory. 1992.

Disponível em <http://www.cs.umd.edu/projects/SoftEng/ESEG/papers/83.55.pdf>

BASILI, V. R.; CALDIERA, G.; ROMBACH, H. D. The Experience Factory. *Encyclopedia of Software Engineering*, v. 1, p. 469-476, 1994.

Disponível em <http://www.wagse.informatik.uni-kl.de/publications/books/encyclo.ef.pdf>

BASILI, V. R.; GREEN, S.; LAITENBERGER, O.; SHULL, F.; SORUMGARD, S.; ZELKOWITZ, M. The Empirical Investigation of Perspective-Based Reading. *Empirical Software Engineering: An International Journal*, v. 1, n. 2, p. 133-161, 1996.

BASILI, V. R.; SHULL, F.; LANUBILLE, F. Building Knowledge Through Families of Experiments. *IEEE Transactions on Software Engineering*, v. 25, n. 4, p. 456-473, 1999.

BOEHM, B. W. *Software Engineering Economics*. Prentice-Hall, 1981.

BOX, G. E. P.; HUNTER, W. G.; HUNTER, J. S. *Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building*. John Wiley and Sons, 1978.

BRACKETT, J. W. *Software Requirements*. SEI Curriculum Module SEI-CM-19-1.2. Software Engineering Institute, 1990.

CEBASE - NSF Center for Empirically Based Software Engineering. 2003.

Disponível em <http://www.cebase.org/www/home/index.htm> (Acessado em 15/07/2003)

CIOLKOWSKI, M.; DIFFERDING, C.; LAITENBERGER, O.; MÜNCH, J. *Empirical Investigation of Perspective-based Reading: A Replicated Experiment*. Technical Report ISERN-97-13, International Software Engineering Research Network, 1997.

COPPE/UFRJ - Engenharia de Software Experimental. 2003.

Disponível em <http://www.cos.ufrj.br/~ese/> (Acessado em 01/10/2003)

- CYSNE, F. P. Transferência de Tecnologia e Desenvolvimento. *Revista Ciência da Informação*, v. 25, n. 1, 1995.
Disponível em <http://www.ibict.br/cionline/250196/25019604.htm> (Acessado em 01/10/2003)
- DESMET – Determining an Evaluation Methodology for Software Methods and Tools. Software Engineering Group at the National Computing Centre, 1994.
- DEUTSCH, M. S. *Software Verification and Validation*. Prentice-Hall, 1982.
- DEVINNEY, T. M. Knowledge, Tacit Understanding and Strategy. 1997.
Disponível em <http://www.agsm.unsw.edu.au/~timdev/research/RABO.PDF> (Acessado em 01/10/2003)
- DÓRIA, E. S. *Replicação de Estudos Empíricos em Engenharia de Software*. Dissertação de Mestrado, Universidade de São Paulo, 2001.
- DURÁN, A.; RUIZ, A.; TORO, M. An Automated Approach for Verification of Software Requirements. 2001.
Disponível em citeseer.nj.nec.com/463132.html
- EBERLEIN, A.; PRADO LEITE, J. C. S. Agile Requirements Definition: A View from Requirements Engineering. 2002.
Disponível em citeseer.nj.nec.com/eberlein02agile.html
- FAGAN, M. Design and Code Inspections to Reduce Errors in Program Development. *IBM Systems Journal*, v. 15, n. 3, 1976.
- FELIZARDO, K. *COTEST - Uma Ferramenta de Apoio à Replicação de um Experimento Baseado em Código Fonte*. Dissertação de Mestrado, PPG-CC UFSCar, 2003.
- GARCIA, R. E. Visualização para Apoio a Estudos Empíricos em Verificação, Validação e Teste de Software. Trabalho para qualificação de doutorado no ICMC/USP, 2003.
- GREMBA, J.; MYERS, C. The IDEAL Model: A Practical Guide for Improvement. [On-line], software Engineering Institute (SEI). [On-line].
Disponível em <http://www.sei.cmu.edu/ideal/ideal.bridge.html> (Acessado em 30/11/2003)

- IEEESTD1233 IEEE Guide for Developing System Requirements Specifications. Software Engineering Standards Committee of the IEEE Computer Society, 1998.
- IEEESTD830 IEEE Recommended Practice for Software Requirements Specifications. Software Engineering Standards Committee of the IEEE Computer Society, 1998.
- INTHURN, C. *Qualidade e Teste de Software*. Visual Books, 2001.
- ISERN International Software Engineering Research Network. 2003.
Disponível em <http://www.iese.fhg.de/ISERN/> (Acessado em 15/07/2003)
- KINNULA, A. *Software Process Engineering Systems: Models and Industry Cases*. 2001.
Disponível em <http://herkules.oulu.fi/isbn9514265084/isbn9514265084.pdf>
- LAMSWEERDE, A. *Requirements Engineering in the Year 00: a Research Perspective*. 2000.
Disponível em citeseer.nj.nec.com/vanlamsweerde00requirements.html
- LINKMAN, S.; ROMBACH, H. D. Experimentation as a Vehicle for Software Technology Transfer - A Family of Software Reading Techniques. *Information and Software Technology*, v. 39, p. 777-780, 1997.
- MALDONADO, J. C.; DÓRIA, E.; MARTIMIANO, L.; FABBRI, S.; MENDONÇA, M.; BASILI, V.; SHULL, F.; CARVER, J. Perspective-Based Reading: a Replicated Experiment Focused on Individual Reviewer Effectiveness. (Artigo submetido). 2004?a.
- MALDONADO, J. C.; MARTIMIANO, L.; HÖHN, E.; OLIVEIRA, M. C.; FABBRI, S. C.; MENDONÇA, M.; SHULL, F.; CARVER, J.; BASILI, V. PBR Replication Studies: Insights from Conflicting Results. (Artigo em preparação). 2004?b.
- MALDONADO, J. C.; OLIVEIRA, M. C.; HÖHN, E.; FABBRI, S. C.; MENDONÇA, M.; SHULL, F.; CARVER, J.; BASILI, V. A Meta Analysis of a PBR Experiments Set: Results from Four Replications. (Artigo em preparação). 2004?c.
- MALTEZ JR., U. P. *Ambiente de Apoio à Replicação de Experimentos de Estudos Empíricos em Engenharia de Software*. Processo FAPESP n.º 03/04617-0, 2003.

- MARUCCI, R. A. *Definição de uma Estratégia de Inspeção para um Processo de Desenvolvimento de Software Orientado a Objetos*. Dissertação de Mestrado, PPG-CC UFSCar, 2002.
- MAYRHAUSER, A. *Software Engineering - Methods and Management*. Academic Press, 1990.
- MENDONÇA, M.; MALDONADO, J. C.; OLIVEIRA, M. C. F.; FABBRI, S. C. P. F.; SHULL, F.; CARVER, J.; BASILI, V. Knowledge Sharing and Improvement Cycles in Software Engineering Experimentation, (Artigo em preparação), 2004?
- NONAKA, I.; TAKEUCHI, H. *Criação de Conhecimento na Empresa*. Campus, 1997.
- PAGLIUSO, P. B. B.; ANDRADE TAMBASCIA, C.; VILLAS-BOAS, A.; FREITAS, M. E. GVR - Guia de Validação de Requisitos Baseado nas Técnicas PBR e *ad-hoc*. Resultante de um Estudo de Caso no CPqD. Anais do WER 2003 - Workshop de Engenharia de Requisitos, 2003.
- PORTER, A.; VOTTA, L.; BASILI, V. Comparing Detection Methods for Software Requirements Inspections: A Replicated Experiment. *IEEE Transaction on Software Engineering*, v. 21, n. 6, p. 563-575, 1995.
- PRESSMAN, R. S. *Engenharia de Software*. 5 ed. McGraw-Hill, 2002.
- READERS - NSF-CNPq Readers Project: A Collaborative Research to Develop, Validate and Package Reading Techniques for Software Defect Detection. 2003.
Disponível em <http://www.labes.icmc.usp.br/readers/> (Acessado em 16/07/2003)
- REIS, L. N. M. Apoio Automatizado à Configuração e Aplicação de OORTs. Anais do WTES - Workshop de Teses em Engenharia de Software, 2003, p. 35-40.
- ROBERTSON, J.; ROBERTSON, S. Volere - Requirements Specification Template. [On-line], atlantic Systems Guild.
Disponível em <http://www.guild.demon.co.uk/SpecTemplate8.pdf> (Acessado em 08/04/2002)
- ROCHA, A. R. C.; MALDONADO, J. C.; WEBER, K. C. *Qualidade de Software - Teoria e Prática*. Prentice Hall, 2001.

- ROMBACH, D. Experimentation: Engine for Applied Research and Technology Transfer in Software Engineering. 1999.
Disponível em http://sel.gsfc.nasa.gov/website/sew/1999/topics/rombach_SEW99paper.pdf (Acessado em 04/10/2003)
- ROMBACH, H. Systematic Quality Improvement. Lecture on June 3 in Dipoli, Helsinki, Finlândia, 1994.
- SALLIS, P. J.; TATE, G.; MACDONEEL, S. G. *Software Engineering: Practice, Management, Improvement*. Addison-Wesley Publishing Company, 1995.
- SHAN-JARVIS, A.; CRANDALL, V. *Introductions to Software Quality: "How To" Guide and Toolkit*. Prentice-Hall, 1997.
- SHULL, F. *Developing Techniques for Using Software Documents: A Series of Empirical Studies*. Tese de Doutorado, University of Maryland, 1998.
Disponível em http://www.cs.umd.edu/projects/SoftEng/ESEG/papers/postscript/shull_dis%20s.ps.gz
- SHULL, F.; BASILI, V.; CARVER, J.; MALDONADO, J. C.; TRAVASSOS, G. H.; MENDONÇA, M.; FABBRI, S. Replicating Software Engineering Experiments: Addressing the Tacit Knowledge Problem. *IEEE Computer Society*, p. 7-16, 2002.
- SHULL, F.; MENDONÇA, M.; BASILI, V.; CARVER, J.; MALDONADO, J. C.; FABBRI, S.; TRAVASSOS, G. H.; OLIVEIRA, M. C. F. Knowledge-Sharing Issues in Experimental Software Engineering. *Empirical Software Engineering: An International Journal*, v. 9, n. 1, p. 1-15. (Artigo aceito. A publicar.). 2004.
- SHULL, F.; RUS, I.; BASILI, V. R. How Perspective-Based Reading Can Improve Requirements Inspections. *Computer*, v. 33, n. 7, p. 73-79, 2000.
- SILVA, L. F. S. Apoio Ferramental para Aplicação de Técnicas de Leitura Baseada em Perspectiva (PBR). Anais do WTES - Workshop de Teses em Engenharia de Software, 2003, p. 83-88.
- SMITH, D. J.; WOOD, K. B. *Engineering Quality Software*. Elsevier Applied Science, 1989.

- SOMMERVILLE, I. *Software Engineering*. 5 ed. Addison-Wesley, 1995.
- SOMMERVILLE, I. *Engenharia de Software*. 6 ed. Addison-Wesley, 2003.
- SORUMGARD, S. An Empirical Study of Process Conformance. 1996.
Disponível em citeseer.nj.nec.com/123830.html
- SPÍNOLA, R. O. Uma Abordagem para Integração de Ferramentas. Anais do WTES - Workshop de Teses em Engenharia de Software, 2003, p. 59-64.
- TORII, K.; MATSUMOTO, K.; NAKAKOJI, K.; TAKADA, Y.; TAKADA, S.; SHIMA, K. Ginger2: An Environment for Computer-Aided Empirical Software Engineering. *IEEE Transactions on Software Engineering*, v. 25, n. 4, p. 456-473, 1999.
- TRAVASSOS, G. H.; SHULL, F.; CARVER, J.; BASILI, V. Reading Techniques for OO Design Inspections. 1999.
Disponível em http://sel.gsfc.nasa.gov/website/sew/1999/topics/travassos_SEW99paper.p%df (Acessado em 01/10/2003)
- WOHLIN, C.; RUNESON, P.; HÖST, M.; OHLSSON, M. C.; REGNELL, B.; WESSLÉN, A. *Experimentation in Software Engineering: an Introduction*. Kluwer Academic Publishers, 2000.
- ZELKOWITZ, M. V. Software Engineering Technology Infusion Within NASA. *IEEE Transactions on Engineering Management*, v. 43, n. 3, p. 250-261, 1996.
Disponível em citeseer.nj.nec.com/zelkowitz96software.html (Acessado em 01/10/2003)

Documento de Especificação de Requisitos
OPER



SAGRE

Documento de Especificação de Requisitos Importação de LICPINO na FCT

Código: OR-043-2001-0000	Versão: 00.00	Arquivo: Apendice_DocumentoRequisitosSAGRE_Oper_Fev2003
Data de geração: 7/2/2003 11:24	Nível de sigilo: RESERVADO	
Data impressão: 20/10/2003 18:50		
Elaborado por: Julio Burjato	UA: 2230	Responsável: Geovane Cayres Magalhães
	Assinatura:	
Verificado por: <u>Assinatura</u> Carlos Alberto Previdelli, CPqD	Aprovado por: <u>Assinatura</u>	
Data: ___/___/___	Data: ___/___/___	

Substitui os documentos:

SUMÁRIO

1	Histórico das Alterações	4
1.1	Lista de distribuição deste documento	4
2	Introdução	5
2.1	Objetivos deste documento	5
2.2	Escopo deste documento	5
2.3	Glossário	5
2.3.1	Abreviaturas	5
2.3.2	Conceitos	5
2.3.3	Convenções Gráficas do Documento	5
2.4	Referências	6
2.4.1	Documentos Fontes	6
2.4.2	Documentos Relacionados	6
3	Objetivos e Escopo do Produto	6
3.1	Objetivos do Produto	6
3.2	Escopo do Produto	6
4	Descrição do Problema	6
4.1	Características principais (<i>features</i>)	6
5	Pressupostos e Dependências	6
6	Requisitos do Produto	7
6.1	Fluxo de Entrada	7
6.1.1	SLNS – Solicitação de LICPINO de Número de Serviço	7
6.1.2	RSLNS – Resposta da solicitação de LICPINO de Número de Serviço	8
6.1.3	RSLNS (ERRO) – Resposta da solicitação de EQN de Número de Serviço com erro	8
6.2	Processo de importação de LICPINO	8
7	Atributos dos Requisitos	9
8	Restrições ao Projeto	9
9	Lista de TBD	9



NOTA DE COPYRIGHT

© 1998 Fundação Centro de Pesquisa e Desenvolvimento em Telecomunicações. Todos os direitos reservados.

Nenhuma parte desta publicação pode ser reproduzida, transmitida, transcrita, armazenada em sistema de recuperação de informações ou traduzida para qualquer língua em qualquer forma e por qualquer meio sem a autorização prévia escrita da Fundação Centro de Pesquisa e Desenvolvimento em Telecomunicações.

CPqD é uma marca registrada da Fundação Centro de Pesquisa e Desenvolvimento em Telecomunicações
SAGRE é uma marca registrada da Fundação Centro de Pesquisa e Desenvolvimento em Telecomunicações
Outras

referências



ÍNDICE DE REQUISITOS

[R] 01	SLNS – Solicitação de LICPINO de Número de Serviço.....	7
[R] 02	RSLNS - Resposta da solicitação de LICPINO de Número de Serviço.....	8
[R] 03	RSLNS (ERRO) – Resposta da solicitação de EQN de Número de Serviço com erro ...	8

1 HISTÓRICO DAS ALTERAÇÕES

Data	Versão	Alteração
DD.MM.YYYY	VV.RR	<Descrição da alteração: o que foi alterado, por quem foi alterado>

1.1 Lista de distribuição deste documento

CPqD:

- Júlio Burjato – Gerência de Requisitos
- Daniel Garcia Teijeiro - Gerência de Requisitos
- Carlos Alberto Previdelli - Gerência de Desenvolvimento
- Renato Tutumi – Gerência de Desenvolvimento
- Sônia - Gerência de Desenvolvimento
- Marcos Borges – Gerência de Desenvolvimento
- Amanda - Gerência de Desenvolvimento
- Eliane Zambon Victorelli Dias - Gerência de Desenvolvimento
- Fátima Izumi Obata - Gerência de Desenvolvimento
- Maurício Tonsig – Gerência de Implantação
- Deise Mara Gouveia Elias - Gerência de Implantação

GVT

- Pascale
- Kinita
- Ogawa

2 INTRODUÇÃO

2.1 Objetivos deste documento

O objetivo deste documento é descrever as funcionalidades e requisitos necessários para suportar a importação dos dados de LICPINO de um sistema de controle de rede interna para o SAGRE/Oper, na emissão de uma folha de corte.

2.2 Escopo deste documento

Neste documento estão descritas as necessidades de negócio e as funcionalidades de usuário e de interface que atendam a estas necessidades. Estão também descritos os requisitos do SAGRE para atender as funcionalidades descritas acima.

Não é escopo deste documento fazer o projeto de software, detalhar (tipo, quantidade, etc.) campos envolvidos na interface com outros sistemas, definir prazos e custos, o que deve ser feito em documentos apropriados.

2.3 Glossário

2.3.1 Abreviaturas

2.3.2 Conceitos

2.3.3 Convenções Gráficas do Documento

Obs.: Os itens desta Seção estão organizados em ordem alfabética.

- **[R] [x] {y}** – símbolo [R] seguido da numeração [x], seguido da numeração {y}, onde:
 - [R] – símbolo que indica que o objeto descrito se trata de um requisito
 - [x] – numeração entre ‘colchetes’ representa o número do requisito no documento em questão
 - {y} – numeração entre ‘chaves’ que representa o número de cadastro do requisito no Sistema SASREQ.

2.4 Referências

2.4.1 Documentos Fontes

2.4.2 Documentos Relacionados

3 OBJETIVOS E ESCOPO DO PRODUTO

3.1 Objetivos do Produto

O objetivo do produto é contemplar a importação dos dados de LICPINO de um sistema de controle de rede interna para o SAGRE/Oper, na emissão de uma folha de corte.

3.2 Escopo do Produto

O escopo do produto é desenvolver funcionalidades que contemplem a importação dos dados de LICPINO de um sistema de controle de rede interna para o SAGRE/Oper, na emissão de uma folha de corte.

4 DESCRIÇÃO DO PROBLEMA

Hoje na emissão de uma folha de corte no SAGRE/Oper não está contemplando os dados referentes aos LICPINO's dos telefones envolvidos, mesmo porque esta informação não está cadastrada na base de dados do SAGRE/Oper e sim em outro sistema que controla a planta interna do cliente. Esta informação é de fundamental importância para a Empresa Operadora, pois quando se vai realizar o corte em campo o DG necessita da informação do LICPINO para executar seu serviço.

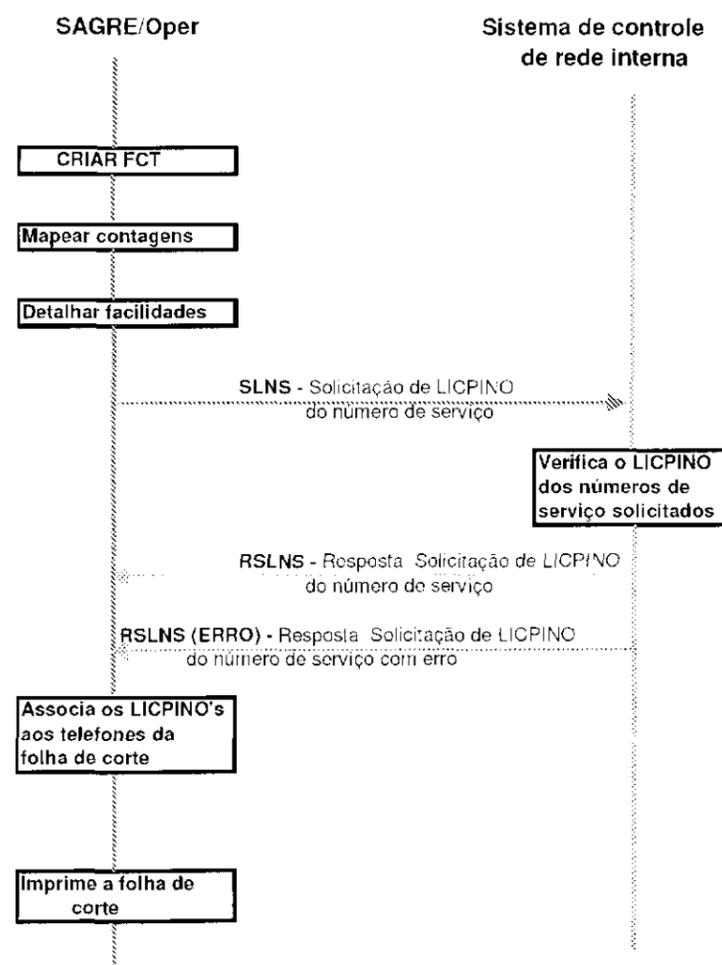
4.1 Características principais (*features*)

5 PRESSUPOSTOS E DEPENDÊNCIAS

Para que as funcionalidades do SAGRE/Oper possam contemplar a importação dos dados de LICPINO de um sistema controle de rede interna para o SAGRE/Oper, na emissão de uma folha de corte, é necessário que haja a contrapartida do sistema de controle de rede interna, dos requisitos aqui definidos.

6 REQUISITOS DO PRODUTO

O diagrama abaixo está demonstrando o fluxo necessário para importação do LICPINO de qualquer sistema que controle a rede interna na Empresa Operadora.



6.1 Fluxo de Entrada

6.1.1 SLNS – Solicitação de LICPINO de Número de Serviço

[R] |01| SLNS – Solicitação de LICPINO de Número de Serviço

O SAGRE/Oper deverá enviar as seguintes informações ao Sistema de controle de rede interna:

Localidade – tamanho variável, máximo 5 caracteres alfanuméricos

Estação Telefônica - tamanho variável, máximo 5 caracteres alfanuméricos

* *Prefixo* - tamanho variável, máximo 5 (de 1 a 9999)

MCDU - tamanho fixo 4 (de 0000 a 9999)

Complemento - tamanho fixo 1 (0,1,3,5, e 9 para Telefônica)

6.1.2 RSLNS – Resposta da solicitação de LICPINO de Número de Serviço

[R] [02] RSLNS - Resposta da solicitação de LICPINO de Número de Serviço

O Sistema de controle de rede interna deverá enviar ao SAGRE/Oper as seguintes informações:

Localidade – tamanho variável, máximo 5 caracteres alfanuméricos

Estação Telefônica - tamanho variável, máximo 5 caracteres alfanuméricos

Prefixo - tamanho variável, máximo 5 (de 1 a 99999)

MCDU - tamanho fixo 4 (de 0000 a 9999)

Complemento - tamanho fixo 1 (0,1,3,5, e 9 para Telefônica)

LICPINO - tamanho variável, Máximo 23 caracteres alfanuméricos

Central - tamanho variável, Máximo 5 caracteres alfanuméricos

Diagnóstico

6.1.3 RSLNS (ERRO) – Resposta da solicitação de EQN de Número de Serviço com erro

[R] [03] RSLNS (ERRO) – Resposta da solicitação de EQN de Número de Serviço com erro

O Sistema de controle de rede interna deverá enviar ao SAGRE/Oper as seguintes informações:

Localidade – tamanho variável, Máximo 5 caracteres alfanuméricos

Estação Telefônica - tamanho variável, Máximo 5 caracteres alfanuméricos

* *Prefixo* - tamanho variável, Máximo 5 (de 1 a 9999)

MCDU - tamanho fixo 4 (de 0000 a 9999)

Complemento - tamanho fixo 1 (0,1,3,5, e 9 para Telefônica)

Diagnóstico de falha

6.2 Processo de importação de LICPINO

As regras técnicas e de negócio para a importação de LICPINO de um sistema de controle da rede interna são as seguintes:

Após o SAGRE/Oper FCT realizar o processo de detalhar facilidades, antes da emissão da FCT deverá solicitar ao Sistema de controle de rede interna os LICPINO's correspondentes aos números de serviço envolvidos na Folha de Corte. Fluxo **SLNS**

Caso as informações enviadas pelo SAGRE/Oper não estiverem corretas deverá ser enviado o fluxo de resposta com erro nos dados enviados - **RSLNS (ERRO)**

O SAGRE/Oper deve ser capaz de tratar a resposta **RSLNS (ERRO)** - Após a emissão do relatório da Folha de corte não será permitido nenhuma inclusão de um número de serviço. Nenhuma inclusão será permitida em uma área que esteja em folha de corte exceto em pares PARA (destino).

Se as informações enviadas pelo SAGRE/Oper estiverem corretas o Sistema de controle da rede interna deverá verificar os LICPINO's correspondentes aos números de serviços solicitados enviar o fluxo de resposta – **RSLNS**

Caso o sistema de controle da rede interna não localizar o LICPINO de um determinado número de serviço este deverá vir em branco.

Após receber os LICPINO's solicitados o SAGRE/Oper deverá associá-los à folha de corte e preparar para a impressão da mesma.

O número de serviço que não constar seu respectivo LICPINO deverá ficar em branco.

Caso o SAGRE/Oper receba a informação de que houve uma falha na comunicação, o sistema deverá permitir que operador possa solicitar novamente a importação do LICPINO ou solicitar que a emissão da folha de corte sem a informação do LICPINO.

Após a emissão da folha de corte qualquer inclusão ou exclusão de um número de serviço deverá ser feita manualmente não sendo possível neste caso pegar automaticamente o LICPINO correspondente.

7 ATRIBUTOS DOS REQUISITOS

8 RESTRIÇÕES AO PROJETO

9 LISTA DE TBD

Documento de Especificação de Requisitos
OPER com Defeitos Marcados



SAGRE

Documento de Especificação de Requisitos Importação de LICPINO na FCT

Código: OR-043-2001-0000	Versão: 00.00	Arquivo: Apendice_DocumentoRequisitosSAGRE_OperErrosMarcados
Data de geração: 12/5/2003 08:53	Nível de sigilo: RESERVADO	
Data impressão: 20/10/2003 18:45		
Elaborado por: Julio Burjato	UA: 2230	Responsável: Geovane Cayres Magalhães
	Assinatura:	
Verificado por: <u>Assinatura</u> Carlos Alberto Previdelli, CPqD	Aprovado por: <u>Assinatura</u>	
Data: ___/___/___	Data: ___/___/___	

Substitui os documentos:

NOTA DE COPYRIGHT

© 1998 Fundação Centro de Pesquisa e Desenvolvimento em Telecomunicações. Todos os direitos reservados.

Nenhuma parte desta publicação pode ser reproduzida, transmitida, transcrita, armazenada em sistema de recuperação de informações ou traduzida para qualquer língua em qualquer forma e por qualquer meio sem a autorização prévia escrita da Fundação Centro de Pesquisa e Desenvolvimento em Telecomunicações.

CPQD é uma marca registrada da Fundação Centro de Pesquisa e Desenvolvimento em Telecomunicações
SAGRE é uma marca registrada da Fundação Centro de Pesquisa e Desenvolvimento em Telecomunicações
Outras

referências

SUMÁRIO

1	Histórico das Alterações	4
1.1	Lista de distribuição deste documento	4
2	Introdução	5
2.1	Objetivos deste documento	5
2.2	Escopo deste documento	5
2.3	Glossário	5
2.3.1	Abreviaturas	5
2.3.2	Conceitos	5
2.3.3	Convenções Gráficas do Documento	5
2.4	Referências	6
2.4.1	Documentos Fontes	6
2.4.2	Documentos Relacionados	6
3	Objetivos e Escopo do Produto	6
3.1	Objetivos do Produto	6
3.2	Escopo do Produto	6
4	Descrição do Problema	6
4.1	Características principais (<i>features</i>)	6
5	Pressupostos e Dependências	6
6	Requisitos do Produto	7
6.1	Fluxo de Entrada	7
6.1.1	SLNS – Solicitação de LICPINO de Número de Serviço	7
6.1.2	RSLNS – Resposta da solicitação de LICPINO de Número de Serviço	8
6.1.3	RSLNS (ERRO) – Resposta da solicitação de EQN de Número de Serviço com erro	8
6.2	Processo de importação de LICPINO	8
7	Atributos dos Requisitos	10
8	Restrições ao Projeto	10
9	Lista de TBD	10



ÍNDICE DE REQUISITOS

[R] 01]	SLNS – Solicitação de LICPINO de Número de Serviço.....	7
[R] 02]	RSLNS - Resposta da solicitação de LICPINO de Número de Serviço.....	8
[R] 03]	RSLNS (ERRO) – Resposta da solicitação de EQN de Número de Serviço com erro ...	8

1 HISTÓRICO DAS ALTERAÇÕES

Data	Versão	Alteração
DD.MM.YYYY	VV.RR	<Descrição da alteração: o que foi alterado, por quem foi alterado>

1.1 Lista de distribuição deste documento

CPqD:

- Júlio Burjato – Gerência de Requisitos
- Daniel Garcia Teijeiro - Gerência de Requisitos
- Carlos Alberto Previdelli - Gerência de Desenvolvimento
- Renato Tutumi - Gerência de Desenvolvimento
- Sônia - Gerência de Desenvolvimento
- Marcos Borges – Gerência de Desenvolvimento
- Amanda - Gerência de Desenvolvimento
- Eliane Zambon Victorelli Dias - Gerência de Desenvolvimento
- Fátima Izumi Obata - Gerência de Desenvolvimento
- Maurício Tonsig – Gerência de Implantação
- Deise Mara Gouveia Elias - Gerência de Implantação

GVT

- Pascale
- Kinita
- Ogawa

2 INTRODUÇÃO

2.1 Objetivos deste documento

O objetivo deste documento é descrever as funcionalidades e requisitos necessários para suportar a importação dos dados de LICPINO (5-II) de um sistema de controle de rede interna para o SAGRE/Oper (6-II), na emissão de uma folha de corte (8-II).

2.2 Escopo deste documento

Neste documento estão descritas as necessidades de negócio e as funcionalidades de usuário e de interface que atendam a estas necessidades. Estão também descritos os requisitos do SAGRE (6-II) para atender as funcionalidades descritas acima.

Não é escopo deste documento fazer o projeto de software, detalhar (tipo, quantidade, etc.) campos envolvidos na interface com outros sistemas, definir prazos e custos, o que deve ser feito em documentos apropriados.

2.3 Glossário

2.3.1 Abreviaturas

(3-O) (6-O) (13-O) (14-O) (15-O) (16-O) (17-O) (18-O) (19-O) (20-O)

2.3.2 Conceitos

2.3.3 Convenções Gráficas do Documento

Obs.: Os itens desta Seção estão organizados em ordem alfabética.

- [R] [x] {y} – símbolo [R] seguido da numeração [x], seguido da numeração {y}, onde:
[R] – símbolo que indica que o objeto descrito se trata de um requisito
[x] – numeração entre ‘colchetes’ representa o número do requisito no documento em questão
{y} – numeração entre ‘chaves’ que representa o número de cadastro do requisito no Sistema SASREQ.

2.4 Referências

2.4.1 Documentos Fontes

2.4.2 Documentos Relacionados

3 OBJETIVOS E ESCOPO DO PRODUTO

3.1 Objetivos do Produto

O objetivo do produto é contemplar a importação dos dados de LICPINO (5-II) de um sistema de controle de rede interna para o SAGRE/Oper (6-II), na emissão de uma folha de corte (8-II).

3.2 Escopo do Produto

O escopo do produto é desenvolver funcionalidades que contemplem a importação dos dados de LICPINO (5-II) de um sistema de controle de rede interna para o SAGRE/Oper (6-II), na emissão de uma folha de corte (8-II).

4 DESCRIÇÃO DO PROBLEMA

Hoje na emissão de uma folha de corte (8-II) no SAGRE/Oper (6-II) não está contemplando os dados referentes aos LICPINO's (5-II) dos telefones envolvidos, mesmo porque esta informação não está cadastrada na base de dados do SAGRE/Oper (6-II) e sim em outro sistema que controla a planta interna do cliente. Esta informação é de fundamental importância para a Empresa Operadora, pois quando se vai realizar o corte em campo o DG necessita da informação do LICPINO para executar seu serviço.

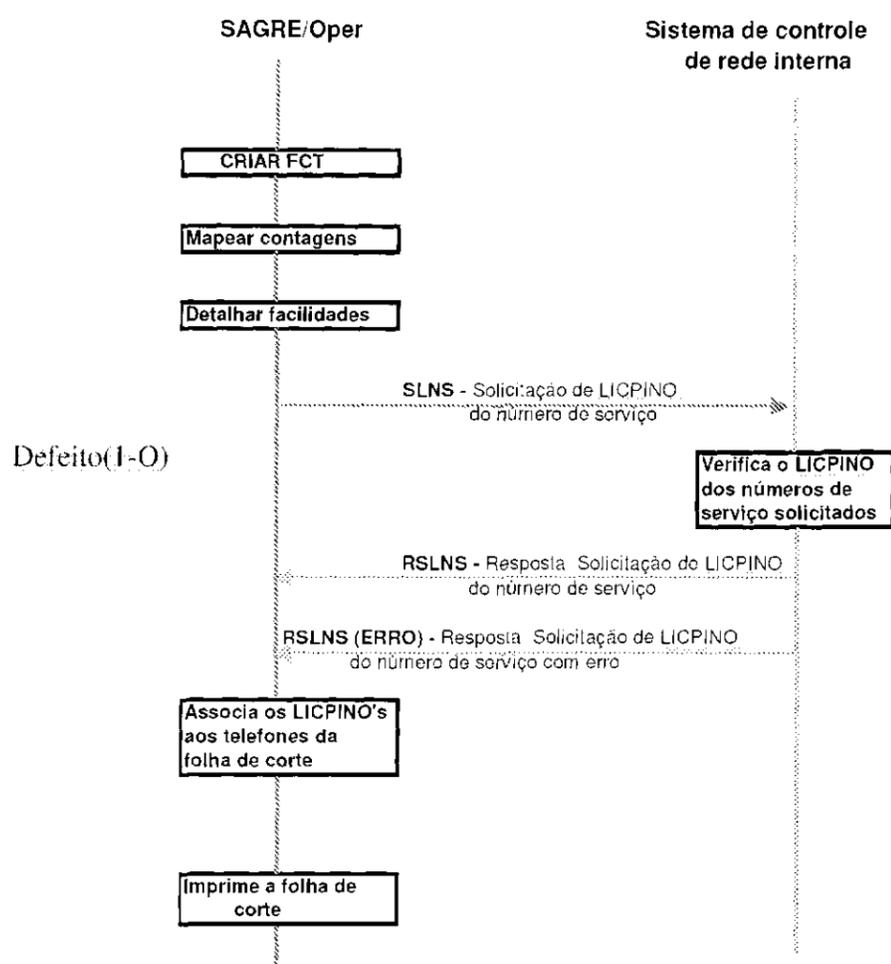
4.1 Características principais (*features*)

5 PRESSUPOSTOS E DEPENDÊNCIAS

Para que as funcionalidades do SAGRE/Oper (6-II) possam contemplar a importação dos dados de LICPINO (5-II) de um sistema controle de rede interna para o SAGRE/Oper (6-II), na emissão de uma folha de corte (8-II), é necessário que haja a contrapartida do sistema de controle de rede interna, dos requisitos aqui definidos.

6 REQUISITOS DO PRODUTO

O diagrama abaixo está demonstrando o fluxo necessário para importação do LICPINO (5-II) de qualquer sistema que controle a rede interna na Empresa Operadora.



6.1 Fluxo de Entrada

6.1.1 SLNS – Solicitação de LICPINO (5-II) de Número de Serviço

[R] | 01 | SLNS – Solicitação de LICPINO (5-II) de Número de Serviço

O SAGRE/Oper (6-II) deverá enviar as seguintes informações ao Sistema de controle de rede interna:

Localidade – tamanho variável, máximo 5 caracteres alfanuméricos (1-II)

Estação Telefônica - tamanho variável, máximo 5 caracteres alfanuméricos (2-II)

Prefixo - tamanho variável, máximo 5 (de 1 a 9999) (3-II) (7-II)

MCDU - tamanho fixo 4 (de 0000 a 9999)

Complemento - tamanho fixo 1 (0,1,3,5, e 9 para Telefônica)

6.1.2 RSLNS – Resposta (1-II) da solicitação de LICPINO (5-II) de Número de Serviço

[R] [02] RSLNS - Resposta da solicitação de LICPINO (5-II) de Número de Serviço

O Sistema de controle de rede interna deverá enviar ao SAGRE/Oper (6-II) as seguintes informações:

Localidade – tamanho variável, máximo 5 caracteres alfanuméricos (4-III)

Estação Telefônica - tamanho variável, máximo 5 caracteres alfanuméricos (5-III)

Prefixo - tamanho variável, máximo 5 (de 1 a 99999) (6-III) (7-II)

MCDU - tamanho fixo 4 (de 0000 a 9999)

Complemento - tamanho fixo 1 (0,1,3,5, e 9 para Telefônica)

LICPINO (5-III) - tamanho variável, Máximo 23 caracteres alfanuméricos.

Central - tamanho variável, Máximo 5 caracteres alfanuméricos

Diagnóstico(5-O)

6.1.3 RSLNS (ERRO) (2-II) – Resposta da solicitação de EQN (5-II) de Número de Serviço com erro

[R] [03] RSLNS (ERRO) Resposta da solicitação de EQN (5-II) de Número de Serviço com erro

O Sistema de controle de rede interna deverá enviar ao SAGRE/Oper (6-II) as seguintes informações:

Localidade tamanho variável, Máximo 5 caracteres alfanuméricos (7-III)

Estação Telefônica - tamanho variável, Máximo 5 caracteres alfanuméricos (8-III)

Prefixo - tamanho variável, Máximo 5 (de 1 a 9999) (9-III) (4-II)

MCDU - tamanho fixo 4 (de 0000 a 9999)

Complemento - tamanho fixo 1 (0,1,3,5, e 9 para Telefônica)

Diagnóstico de falha(11-O) (2-O)

6.2 Processo de importação de LICPINO (5-II) (1-DD)

As regras técnicas e de negócio para a importação de LICPINO (5-II) de um sistema de controle da rede interna são as seguintes:

Após o SAGRE/Oper -- FCT (6-II) realizar o processo de detalhar facilidades, antes da emissão da FCT (8-II) deverá solicitar ao Sistema de controle de rede interna os LICPINO's (5-II) correspondentes aos números de serviço envolvidos na Folha de Corte (8-II). Fluxo **SLNS**

Caso as informações enviadas pelo SAGRE/Oper (6-II) não estiverem corretas (10-O) deverá ser enviado o fluxo de resposta com erro nos dados enviados - **RSLNS (ERRO)**

(9-O): *Se uma RSNLS é enviada com algum campo em branco, a resposta é uma RSLNS(erro) ou um número de serviço, em branco?*

O SAGRE/Oper (6-II) deve ser capaz de tratar a resposta **RSLNS (ERRO)** (21-O)- Após a emissão do relatório da Folha de corte (8-II) não será permitido nenhuma inclusão de um número de serviço.

Se as informações enviadas pelo SAGRE/Oper (6-II) estiverem corretas o Sistema de controle da rede interna deverá verificar os LICPINO's (5-II) correspondentes aos números de serviços solicitados enviar o fluxo de resposta – **RSLNS**

Caso o sistema de controle da rede interna não localizar (2-DD) o LICPINO (5-II) de um determinado número de serviço este deverá vir em branco. (7-O) (5-DD)

Após receber os LICPINO's (5-II) solicitados o SAGRE/Oper (6-II) deverá associá-los à folha de corte (8-II) e preparar para a impressão da mesma.

O número de serviço que não constar seu respectivo LICPINO (5-II) deverá ficar em branco. (4-DD) (5-DD)

Caso o SAGRE/Oper (6-II) receba a informação de que houve uma falha na comunicação (3-DD) (12-O), o sistema deverá permitir que operador possa solicitar novamente a importação do LICPINO (5-II) ou solicitar que a emissão da folha de corte (8-II) sem a informação do LICPINO. (5-II)

Após a emissão da folha de corte (8-II) qualquer inclusão ou exclusão de um número de serviço deverá ser feita manualmente não sendo possível neste caso pegar automaticamente o LICPINO (5-II) correspondente.

(4-O): *não escreve como cancelar a solicitação.*

(8-O) *não especifica o que deve retornar se o sistema localizar mais de um nro de serviço para o mesmo LICPINO. Não é especificado se existe correspondência 1 para 1 entre nro de serviço e LICPINO.*

7 ATRIBUTOS DOS REQUISITOS

8 RESTRIÇÕES AO PROJETO

9 LISTA DE TBD

Ocorrências:

- Omissão (O)
- Informação Inconsistente (II)
- Fato Incoerente (FI)
- Demais Defeitos (DD = MD)
- Informação Ausente
- Informação Estranha (IE)

Lista de Defeitos e Falsos Positivos do
Documento OPER

LISTA DE DEFEITOS
Documento de Especificação de Requisitos de Importação de LICPINO na FCT

Defeito #	Proj. Piloto	Pag.	Req. #	Tipo	Descrição
Defeitos Históricos – Omissão					
1-O	-	07	6	O	Não foi explicitada no diagrama de Requisitos do Produto do documento OR-0043-2001-0000 a delimitação da Emissão de FCT
2-O	-	08	6.1.3	O	Não são informados quais os tipos de erros que podem ser identificados no diagnóstico de falha.
Defeitos Novos – Omissão					
3-O	P1	6	4	O	Não é especificado no documento o termo 'DG'.
4-O	P1	8	6.2	O	Não é informado se é possível cancelar a solicitação.
5-O	P1	8	6.1.2	O	Não é especificado o formato do parâmetro de retorno "Diagnóstico".
6-O	P1	8/9	6.1.1 6.1.2 6.1.3	O	Não existe descrição da sigla MCDU.
7-O	P1	9	6.2	O	Não é especificado através de qual dos dois fluxos de retorno ocorre o "vir em branco".
8-O	P1	9	6.2	O	Não é especificado o que deve retornar se o sistema localizar mais de um LICPINO para o mesmo número de serviço.
9-O	P1	8	6.2	O	Se uma SLNS é enviada com algum campo em branco, não fica claro se a resposta é uma RSLNS (ERRO) ou um número de serviço em branco.
10-O	P1	8	6.2	O	Não fica claro qual a diferença entre uma informação incorreta e uma não localizada.
11-O	P1	8	6.1.3	O	Não é especificado o formato do parâmetro "Diagnóstico de Falha".
12-O	P1	9	6.2	O	Não é descrito como identificar uma falha de comunicação.
13-O	P2	Todo Doc.	Todo Doc.	O	Não existe descrição da sigla LICPINO
14-O	-	Todo Doc.	Todo Doc.	O	Não existe descrição da sigla SAGRE/Oper
15-O	-	5	2.2	O	Não existe descrição da sigla SAGRE
16-O	-	5	2.3.3	O	Não existe descrição da sigla SASREQ
17-O	P2	8	6.1.3	O	Não existe descrição da sigla EQN.
18-O	-	8	6.2	O	Não existe descrição da sigla SAGRE/Oper-FCT.
19-O	P2	8	6.2	O	Não existe descrição da sigla FCT.
20-O	-	9	6.2	O	Não existe descrição da sigla TBD.
21-O	P2	9	6.2 § 4	O	Não especifica a funcionalidade para "tratar a resposta RSLSN (ERRO)".
22-O	P2	5	2.2	O	Não foram descritas as funcionalidades do sistema.
23-O	P2	6	3.1	O	Não é mencionado quais são os dados do LICPINO.
Defeitos Históricos – Fato Incorreto					
1-FI	-	7	6.1.1	IF	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Localidade deveria conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
2-FI	-	7	6.1.1	IF	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Estação telefônica deveria conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos
3-FI	-	7	6.1.1	IF	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Prefixo deveria conter tamanho variável 4 (1 – 9999) ao invés de tamanho variável, máximo 5 (1 – 99999).

Defeito #	Proj Piloto	Pág.	Req. #	Tipo	Descrição
Defeitos Históricos – Fato Incorreto					
4-FI	-	8	6.1.2	IF	No item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço, o Localidade deveria conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
5-FI	-	8	6.1.2	IF	No item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço, o campo Estação telefônica deveria conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
6-FI	-	8	6.1.2	IF	No item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço, o campo Prefixo deveria conter 4 (1 – 9999) ao invés tamanho variável, máximo 5 (1 – 99999).
7-FI	-	8	6.1.3	IF	No item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Localidade deveria conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
8-FI	-	8	6.1.3	IF	No item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Estação Telefônica deveria conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
9-FI	-	8	6.1.3	IF	No item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Prefixo deveria conter 4 (1 – 9999) ao invés tamanho variável, máximo 5 (1 – 99999).
Defeitos Novos – Fato Incorreto					
10-FI	P2	7	6	IF	Nem todos os sistemas de rede interna existentes em Empresas Operadoras têm LICPINO.
11-FI	P2	4	Geral	IF	O documento cita que o cliente é telefônica, mas a lista de distribuição é GVT.
Defeitos Históricos – Informação Inconsistente					
1-II	-	2/8	Índice/ 6.1.2	II	O item RSLNS – Resposta da solicitação de LICPINO de número de serviço do documento OR-0043-2001-0000 está colocado como um item de entrada, mas o próprio nome diz que o item é de resposta.
2-II	-	2/8	Índice/ 6.1.3	II	O item RSLNS (ERRO) – Resposta da solicitação de LICPINO de número de serviço com erro do documento OR-0043-2001-0000 está colocado como um item de entrada, mas o próprio nome diz que o item é de resposta.
3-II	-	7	6.1.1	II	O campo prefixo do item 6.1.1 está descrito como tendo tamanho variável, máximo 5 caracteres e apresenta como máscara 9999.
4-II	-	8	6.1.3	II	O campo prefixo do item 6.1.3 está descrito como tendo tamanho variável, máximo 5 caracteres e apresenta como máscara 9999.
5-II	-	Todo Doc.	Todo Doc.	II	As denominações LICPINO e EQN foram utilizadas em todo o documento para descrever a mesma coisa.
6-II	-	Todo Doc.	Todo Doc.	II	As denominações SAGRE, SAGRE/Oper e SAGRE/Oper-FCT foram usadas indistintamente em todo o documento.

Defeito #	Proj Piloto	Pág.	Req. #	Tipo	Descrição
Defeitos Novos – Informação Inconsistente					
7-II	P2	8	6.1.1/6.1.2	II	Há uma inconsistência no tamanho do campo prefixo do requisito 6.1.2 com o requisito 6.1.1.
8-II	P1	8	6.2	II	São utilizados os termos "FCT" e "folha de corte" para designar a mesma coisa.
Defeitos Históricos – Defeitos Diversos					
1-DD	-	8/9	6.2	MD	As informações apresentadas no item Processo de importação de LICPINO do documento OR-0043-2001-0000 deveriam estar diluídas nos requisitos.
2-DD	-	9	6.2 § 6	MD	O procedimento sobre o que acontece caso o sistema de controle da rede interna não localize o LICPINO de um determinado número de serviço deveria estar no item "RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro".
3-DD	-	9	6.2 § 9	MD	O procedimento para quando ocorrer falha de comunicação na solicitação de LICPINO deveria estar especificado no item "RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro".
Defeitos Novos – Defeitos Diversos					
4-DD	P2	9	6.2 § 8	MD	O procedimento para quando o número de serviço não possuir seu respectivo LICPINO deveria estar especificado em requisito específico.
5-DD	P2	9	6.2 § 6 / § 8	MD	O parágrafo 8 "O número de serviço que não constar seu respectivo LICPINO deverá ficar em branco." está redundante com o parágrafo 6.

LISTA DE FALSOS-POSITIVOS
Documento de Especificação de Requisitos de Importação de LICPINO na FCT

FP #	Proj	Pág.	Req. #	Tipo	Descrição	Justificativa
1	P1	9	6.2	O	Não é especificado se existe número máximo de números de serviço por FCT.	É relatado como defeito devido à falta de delimitação no diagrama.
2	P1	9	6.2 § 10	O	A inclusão manual não é especificada como inclusão no relatório ou no sistema para atualização da folha de corte.	É relatado como defeito devido à falta de delimitação no diagrama.
3	P1	9	6.2 § 10	O	Não descreve o procedimento para quando um número de serviço que não existe, ou que já esteja presente em outra FCT, seja inserido numa folha de corte, através da inclusão manual.	É relatado como defeito devido à falta de delimitação no diagrama.
4	P1	9	6.2 § 10	O	A associação manual pode possibilitar que um LICPINO inválido seja associado a um nº de serviço.	É relatado como defeito devido à falta de delimitação no diagrama.
5	P1	9	6.2	O	Não é descrito o processo "Mapear Contagem" que consta do fluxo SLNS.	É relatado como defeito devido à falta de delimitação no diagrama.
6	P1	9	4 e 6.2 § 9	IF	A folha de corte impressa sem a informação dos LICPINOS não atende ao objetivo do sistema de associar/importar esse dado da rede interna (no caso de falha de comunicação não faz sentido imprimir sem o LICPINO).	É relatado como defeito devido à falta de delimitação no diagrama.
7	P1	9	6.2 § 7	O	A informação de como é feita a associação dos LICPINOS à folha de corte está omissa.	É relatado como defeito devido à falta de delimitação no diagrama.
8	P1	9	6.2 § 7	O	Não especifica em que consiste a preparação para a impressão da folha de corte.	É relatado como defeito devido à falta de delimitação no diagrama.
9	P2	9	6.2 § 10	MD	O último parágrafo da seção 6.2 deveria estar descrito em requisito específico.	É relatado como defeito devido à falta de delimitação no diagrama.
10	P2	5	Geral	O	Não há definição sobre folha de corte, mapear contagens, detalhar facilidades. Apenas aparecem no diagrama.	É relatado como defeito devido à falta de delimitação no diagrama.
11	P2	5	Geral	O	Não há definição sobre folha de corte, mapear contagem, detalhar facilidades. Apenas aparecem no diagrama.	É relatado como defeito devido à falta de delimitação no diagrama.
12	P2	5	2.3.2	O	Falta definir no item de Conceitos o assunto folha de corte.	É relatado como defeito devido à falta de delimitação no diagrama.
13	P2	9	6.2	O	A descrição do requisito não está de acordo com o diagrama. Não há um fluxo representando "falha na comunicação".	

Retirados da Lista de Defeitos

#	Pág.	Req. #	Tipo	Descrição	Justificativa
2)	08	6.1.2	O	Não foi apresentado no documento o fato de que o SAGRE/Oper-FCT não deve verificar a informação da central no Banco de Dados.	Foi retirado porque em nenhum lugar do documento é dito que o BD deve ser verificado.
16-O	7	6.1.1	O	Não é especificado qual deve ser a ordem do envio dos parâmetros SLNS.	No item 2.2 está dito que não será detalhada a interface com outros sistemas.
17-O	9	6.2	O	Não especifica se a recepção do LICPINO se dá na mesma seqüência das solicitações o que gera dúvidas na ação de associação.	No item 2.2 está dito que não será detalhada a interface com outros sistemas.
2-IA	9	6.2 § 8	A	Na expressão "O número de serviço que não constar seu respectivo LICPINO deverá ficar em branco" deixa dúvida sobre o que deve ficar em branco no relatório (o número de serviço ou o LICPINO?)	A redação é clara quanto ao sujeito da oração.

Histórico da Lista de Defeitos

Histórico da Lista de Defeitos

Documento de Requisitos: Importação de LIQPINO na FCT

1ª Lista de Defeitos

A primeira lista foi elaborada com base na comparação realizada entre a versão inicial e a última versão existente do documento de requisitos e foi denominada de Lista de Defeitos Históricos. Composta por 13 defeitos, distribuídos pela taxonomia, segundo a tabela abaixo:

Omissão	Informação Inconsistente	Fato Incorreto	Informação Estranha	Ambigüidade	Defeitos Diversos
3	2	3	0	0	5

1ª Lista

Defeito #	Pag.	Req. #	Tipo	Descrição
1	06	Requisitos do Produto	O	Não foi explicitado no diagrama de Requisitos do Produto do documento OR-0043-2001-0000 a delimitação da Emissão de FCT
2	07	Fluxo de Entrada/ Resposta	O	Foi adicionado ao item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço os seguintes itens: - Caso o sistema de controle da rede interna não localizar o EQN de um determinado número de serviço este deverá vir em branco; - O SAGRE não deverá verificar a informação de central no Banco de Dados.
3	07	Fluxo de Entrada	O	Foi adicionado ao item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro os seguintes itens: - Caso as informações enviadas pelo SAGRE/Oper-FCT não estiverem corretas deverá ser enviado o fluxo de resposta com erro nos dados enviados; - Os Códigos de erro podem ser: . Telefone não existe; . Falha na comunicação; . Time-out (sistema de controle de rede interna não responde em x tempo) - No caso de uma falha de comunicação deverá haver as seguintes opções: . Solicitar novamente a importação do EQN
4	2	Índice	II	O item RSLNS – Resposta da solicitação de LICPINO de número de serviço do documento OR-0043-2001-0000 está colocado como um item de entrada e o próprio nome diz que o item é de resposta.
5	2	Índice	II	O item RSLNS (ERRO) – Resposta da solicitação de LICPINO de número de serviço com erro do documento OR-0043-2001-0000 está colocado como um item de entrada e o próprio nome diz que o item é de resposta.

Defeito #	Pág.	Req. #	Tipo	Descrição
6	7	Fluxo de Entrada	IF	O item SLNS – solicitação de LICPINO de Número de Serviço sofreu algumas alterações com relação ao documento OR-0043-2001-0000 para o documento OR-0043-2001-0100 . Foram: Localidade - passou de tamanho variável, máximo 5 caracteres alfanuméricos para 6 caracteres numéricos; Estação telefônica - passou de tamanho variável, máximo 5 caracteres alfanuméricos para 11 caracteres alfanuméricos; Prefixo - passou de tamanho variável, máximo 5 (1 – 99999) para 4 (1 – 9999);
7	7	Fluxo de Entrada	IF	O item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço sofreu algumas alterações com relação ao documento OR-0043-2001-0000 para o documento OR-0043-2001-0100 . Foram: Localidade - passou de tamanho variável, máximo 5 caracteres alfanuméricos para 6 caracteres numéricos; Estação telefônica - passou de tamanho variável, máximo 5 caracteres alfanuméricos para 11 caracteres alfanuméricos; Prefixo - passou de tamanho variável, máximo 5 (1 – 99999) para 4 (1 – 9999);
8	7	Fluxo de Entrada	IF	O item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro sofreu algumas alterações com relação ao documento OR-0043-2001-0000 para o documento OR-0043-2001-0100 . Foram: Localidade - passou de tamanho variável, máximo 5 caracteres alfanuméricos para 6 caracteres numéricos; Estação telefônica - passou de tamanho variável, máximo 5 caracteres alfanuméricos para 11 caracteres alfanuméricos; Prefixo - passou de tamanho variável, máximo 5 (1 – 99999) para 4 (1 – 9999);
9	2	Índice	MD	O item Processo de importação de LICPINO do documento OR-0043-2001-0000 foi diluído nos Requisitos do Produto (item 4) do documento OR-0043-2001-0100
10	3	Índice de Requisitos	MD	Ocorreu um erro no índice de requisitos do documento OR-0043-2001-0000 e nada foi encontrado.
11	5	Propósito	MD	O item Propósito do documento OR-0043-2001-0100 sofreu algumas alterações textuais do documento OR-0043-2001-0000 .
12	Hole Doc.	Hole Doc.	MD	Em todo documento OR-0043-2001-0100 a denominação SAGRE e SAGRE/Oper foi substituída no documento OR-0043-2001-0000 por SAGRE/Oper – FCT
13	Hole Doc.	Hole Doc.	MD	Em todo documento OR-0043-2001-0100 a denominação LICPINO foi substituída no documento OR-0043-2001-0000 por EQN

2ª Lista de Defeitos

Após consenso das equipes, alguns defeitos da 1ª lista foram desmembrados por tratarem de problemas diferentes.

O defeito 2 da primeira lista foi desmembrado em dois: defeito 2 e defeitos 3 da segunda lista.

O defeito 3 da primeira lista foi dividido em três na segunda lista: defeitos 4, 5 e 6.

O defeito 6 da primeira lista foi desmembrado em três, um defeito para cada item que ele trata, tornando-se os defeitos 9, 10 e 11 da segunda lista. Da mesma forma os defeitos 7 e 8 da primeira lista foram divididos em 3, cada um, correspondendo na segunda lista aos defeitos 12, 13 e 14 e 15, 16 e 17, respectivamente.

O defeito 10 da primeira lista foi retirado por tratar de problemas de edição e foi decidido que esses tipos de defeitos não seriam considerados.

O defeito 11 da primeira lista foi retirado por ser apenas um melhoramento na redação e não um erro.

De acordo com a taxonomia, os defeitos da segunda lista ficaram distribuídos da seguinte forma:

Omissão	Informação Inconsistente	Fato Incorreto	Informação Estranha	Ambigüidade	Defeitos Diversos
6	2	9	0	0	3

Tabela de Correlação entre as Listas 1 e 2

Lista 1	Lista 2
1	1
2	2 e 3
3	4, 5 e 6
4	7
5	8
6	9, 10 e 11
7	12, 13 e 14
8	15, 16 e 17
9	18
10	Retirado
11	Retirado
12	19
13	20

2ª Lista

Defeitos Históricos				
Defeito #	Pág.	Req. #	Tipo	Descrição
1	06	Requisitos do Produto	O	Não foi explicitado no diagrama de Requisitos do Produto do documento OR-0043-2001-0000 a delimitação da Emissão de FCT
2	07	Fluxo de Entrada/Resposta	O	Foi adicionado ao item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço o seguinte item: - Caso o sistema de controle da rede interna não localizar o EQN de um determinado número de serviço este deverá vir em branco:

Defeito #	Pág.	Req. #	Tipo	Descrição
3	07	Fluxo de Entrada/ Resposta	O	Foi adicionado ao item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço o seguinte item: - O SAGRE não deverá verificar a informação de central no Banco de Dados.
4	07	Fluxo de Entrada	O	Foi adicionado ao item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro o seguinte item: - Caso as informações enviadas pelo SAGRE/Oper-FCT não estiverem corretas deverá ser enviado o fluxo de resposta com erro nos dados enviados;
5	07	Fluxo de Entrada	O	Foi adicionado ao item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro o seguinte item: - Os Códigos de erro podem ser: . Telefone não existe; . Falha na comunicação; . Time-out (sistema de controle de rede interna não responde em x tempo)
6	07	Fluxo de Entrada	O	Foi adicionado ao item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro o seguinte item: - No caso de uma falha de comunicação deverá haver as seguintes opções: . Solicitar novamente a importação do EQN
7	2	Índice	II	O item RSLNS – Resposta da solicitação de LICPINO de número de serviço do documento OR-0043-2001-0000 está colocado como um item de entrada e o próprio nome diz que o item é de resposta.
8	2	Índice	II	O item RSLNS (ERRO) – Resposta da solicitação de LICPINO de número de serviço com erro do documento OR-0043-2001-0000 está colocado como um item de entrada e o próprio nome diz que o item é de resposta.
9	7	Fluxo de Entrada	IF	O item SLNS – solicitação de LICPINO de Número de Serviço sofreu alteração com relação ao documento OR-0043-2001-0000 para o documento OR-0043-2001-0100 . Foi: Localidade - passou de tamanho variável, máximo 5 caracteres alfanuméricos para 6 caracteres numéricos;
10	7	Fluxo de Entrada	IF	O item SLNS – solicitação de LICPINO de Número de Serviço sofreu alteração com relação ao documento OR-0043-2001-0000 para o documento OR-0043-2001-0100 . Foi: Estação telefônica - passou de tamanho variável, máximo 5 caracteres alfanuméricos para 11 caracteres alfanuméricos;
11	7	Fluxo de Entrada	IF	O item SLNS – solicitação de LICPINO de Número de Serviço sofreu alteração com relação ao documento OR-0043-2001-0000 para o documento OR-0043-2001-0100 . Foi: Prefixo - passou de tamanho variável, máximo 5 (1 – 99999) para 4 (1 – 9999);

Defeito #	Pág.	Req. #	Tipo	Descrição
12	7	Fluxo de Entrada	IF	O item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço sofreu alteração com relação ao documento OR-0043-2001-0000 para o documento OR-0043-2001-0100 . Foi: Localidade - passou de tamanho variável, máximo 5 caracteres alfanuméricos para 6 caracteres numéricos;
13	7	Fluxo de Entrada	IF	O item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço sofreu alteração com relação ao documento OR-0043-2001-0000 para o documento OR-0043-2001-0100 . Foi: Estação telefônica - passou de tamanho variável, máximo 5 caracteres alfanuméricos para 11 caracteres alfanuméricos;
14	7	Fluxo de Entrada	IF	O item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço sofreu alteração com relação ao documento OR-0043-2001-0000 para o documento OR-0043-2001-0100 . Foi: Prefixo - passou de tamanho variável, máximo 5 (1 – 99999) para 4 (1 – 9999);
15	7	Fluxo de Entrada	IF	O item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro sofreu alteração com relação ao documento OR-0043-2001-0000 para o documento OR-0043-2001-0100 . Foi: Localidade - passou de tamanho variável, máximo 5 caracteres alfanuméricos para 6 caracteres numéricos;
16	7	Fluxo de Entrada	IF	O item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro sofreu alteração com relação ao documento OR-0043-2001-0000 para o documento OR-0043-2001-0100 . Foi: Estação telefônica - passou de tamanho variável, máximo 5 caracteres alfanuméricos para 11 caracteres alfanuméricos;
17	7	Fluxo de Entrada	IF	O item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro sofreu alteração com relação ao documento OR-0043-2001-0000 para o documento OR-0043-2001-0100 . Foi: Prefixo - passou de tamanho variável, máximo 5 (1 – 99999) para 4 (1 – 9999);
18	2	Índice	MD	O item Processo de importação de LICPINO do documento OR-0043-2001-0000 foi diluído nos Requisitos do Produto (item 4) do documento OR-0043-2001-0100
19	Todo Doc.	Todo Doc.	MD	Em todo documento OR-0043-2001-0100 a denominação SAGRE e SAGRE/Oper foi substituída no documento OR-0043-2001-0000 por SAGRE/Oper – FCT
20	Todo Doc.	Todo Doc.	MD	Em todo documento OR-0043-2001-0100 a denominação LICPINO foi substituída no documento OR-0043-2001-0000 por EQN

3ª Lista de Defeitos

A terceira lista de defeitos foi formada com a inclusão de 20 defeitos novos detectados no Projeto Piloto 1 (defeito 21 ao 40), ficando um total de 40 defeitos.

De acordo com a taxonomia, os defeitos da terceira lista ficaram distribuídos da seguinte forma:

Omissão	Informação Inconsistente	Fato Incorreto	Informação Estranha	Ambigüidade	Defeitos Diversos
21	2	10	1	3	3

3ª Lista

Defeitos Históricos				
Defeito #	Pág.	Req. #	Tipo	Descrição
1	06	Requisitos do Produto	O	Não foi explicitado no diagrama de Requisito: do Produto do documento OR-0043-2001-0000 a delimitação da Emissão de FCT
2	07	Fluxo de Entrada/ Resposta	O	Foi adicionado ao item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço o seguinte item: - Caso o sistema de controle da rede interna não localizar o EQN de um determinado número de serviço este deverá vir em branco.
3	07	Fluxo de Entrada/ Resposta	O	Foi adicionado ao item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço o seguinte item: - O SAGRE não deverá verificar a informação de central no Banco de Dados.
4	07	Fluxo de Entrada	O	Foi adicionado ao item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro o seguinte item: - Caso as informações enviadas pelo SAGRE/Oper-FCT não estiverem corretas deverá ser enviado o fluxo de resposta com erro nos dados enviados.
5	07	Fluxo de Entrada	O	Foi adicionado ao item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro o seguinte item: - Os Códigos de erro podem ser: . Telefone não existe; . Falha na comunicação; . Time-out (sistema de controle de rede interna não responde em x tempo)
6	07	Fluxo de Entrada	O	Foi adicionado ao item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro o seguinte item: - No caso de uma falha de comunicação deverá haver as seguintes opções: . Solicitar novamente a importação do EQN

Defeitos Históricos				
Defeito #	Pág.	Req. #	Tipo	Descrição
7	2	Índice	II	O item RSLNS – Resposta da solicitação de LICPINO de número de serviço do documento OR-0043-2001-0000 está colocado como um item de entrada e o próprio nome diz que o item é de resposta.
8	2	Índice	II	O item RSLNS (ERRO) – Resposta da solicitação de LICPINO de número de serviço com erro do documento OR-0043-2001-0000 está colocado como um item de entrada e o próprio nome diz que o item é de resposta.
9	7	Fluxo de Entrada	IF	O item SLNS – solicitação de LICPINO de Número de Serviço sofreu alteração com relação ao documento OR-0043-2001-0000 para o documento OR-0043-2001-0100 . Foi: Localidade - passou de tamanho variável, máximo 5 caracteres alfanuméricos para 6 caracteres numéricos.
10	7	Fluxo de Entrada	IF	O item SLNS – solicitação de LICPINO de Número de Serviço sofreu alteração com relação ao documento OR-0043-2001-0000 para o documento OR-0043-2001-0100 . Foi: Estação telefônica - passou de tamanho variável, máximo 5 caracteres alfanuméricos para 11 caracteres alfanuméricos.
11	7	Fluxo de Entrada	IF	O item SLNS – solicitação de LICPINO de Número de Serviço sofreu alteração com relação ao documento OR-0043-2001-0000 para o documento OR-0043-2001-0100 . Foi: Prefixo - passou de tamanho variável, máximo 5 (1 – 99999) para 4 (1 – 9999).
12	7	Fluxo de Entrada	IF	O item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço sofreu alteração com relação ao documento OR-0043-2001-0000 para o documento OR-0043-2001-0100 . Foi: Localidade - passou de tamanho variável, máximo 5 caracteres alfanuméricos para 6 caracteres numéricos.
13	7	Fluxo de Entrada	IF	O item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço sofreu alteração com relação ao documento OR-0043-2001-0000 para o documento OR-0043-2001-0100 . Foi: Estação telefônica - passou de tamanho variável, máximo 5 caracteres alfanuméricos para 11 caracteres alfanuméricos.
14	7	Fluxo de Entrada	IF	O item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço sofreu alteração com relação ao documento OR-0043-2001-0000 para o documento OR-0043-2001-0100 . Foi: Prefixo - passou de tamanho variável, máximo 5 (1 – 99999) para 4 (1 – 9999).
15	7	Fluxo de Entrada	IF	O item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro sofreu alteração com relação ao documento OR-0043-2001-0000 para o documento OR-0043-2001-0100 . Foi: Localidade - passou de tamanho variável, máximo 5 caracteres alfanuméricos para 6 caracteres numéricos.

Defeitos Históricos				
Defeito #	Pág.	Req. #	Tipo	Descrição
16	7	Fluxo de Entrada	IF	O item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro sofreu alteração com relação ao documento OR-0043-2001-0000 para o documento OR-0043-2001-0100 . Foi: Estação telefônica - passou de tamanho variável, máximo 5 caracteres alfanuméricos para 11 caracteres alfanuméricos;
17	7	Fluxo de Entrada	IF	O item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro sofreu alteração com relação ao documento OR-0043-2001-0000 para o documento OR-0043-2001-0100 . Foi: Prefixo - passou de tamanho variável, máximo 5 (1 – 99999) para 4 (1 – 9999).
18	2	Índice	MD	O item Processo de importação de LICPINO do documento OR-0043-2001-0000 foi diluído nos Requisitos do Produto (item 4) do documento OR-0043-2001-0100
19	Todo Doc.	Todo Doc.	MD	Em todo documento OR-0043-2001-0100 a denominação SAGRE e SAGRE/Oper foi substituída no documento OR-0043-2001-0000 por SAGRE/Oper – FCT
20	Todo Doc.	Todo Doc.	MD	Em todo documento OR-0043-2001-0100 a denominação LICPINO foi substituída no documento OR-0043-2001-0000 por EQN
Defeitos Novos				
21	6	3.1	O	Não é especificado no documento o termo 'DG' utilizado no requisito.
22	8	4.2	O	Não é descrito como cancelar a solicitação e se é possível cancelar.
23	7	4.1.2	O	Não é especificado o formato do parâmetro de retorno "Diagnóstico".
24	7	4.1.1 4.1.2 4.1.3	O	Não existe descrição da sigla MCDU, utilizada para descrição de informação de I/O.
25	8	4.2	O	Não é descrito o processo de "Mapear contagens" que consta do fluxo SLNS. Ela influencia a importação de licpinos?
26	8	4.2	A	Na expressão "O número de serviço que não constar seu respectivo LICPINO deverá ficar em branco" deixa dúvida sobre o que deve ficar em branco no relatório (o número de serviço ou o LIQPINO?)
27	8	4.2	A	São utilizados os termos "FCT" e "folha de corte" para designar o resultado.
28	8	4.2 e 3.1	IF	A folha de corte impressa sem a informação dos LICPINOS não atende ao objetivo do sistema de associar/importar esse dado da rede interna (no caso de falha de comunicação não faz sentido imprimir sem o LICPINO).
29	8	4.2	O	Não é especificado se existe número máx de números por FCT.
30	8	4.2	O	"vir em branco" através de qual fluxo? RSLNS?
31	8	4.2	EI	Informação de como é feita a associação à folha de corte está omissa.
32	8	4.2	O	Não é especificado o que deve retornar se o sistema localizar mais de um LICPINO para o mesmo número de serviço. Não é especificado se existe correspondência 1 para 1 entre os números de serviços e os LICPINOS.

Defeitos Históricos				
Defeito #	Pág.	Req. #	Tipo	Descrição
16	7	Fluxo de Entrada	IF	O item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro sofreu alteração com relação ao documento OR-0043-2001-0000 para o documento OR-0043-2001-0100 . Foi: Estação telefônica - passou de tamanho variável, máximo 5 caracteres alfanuméricos para 11 caracteres alfanuméricos;
17	7	Fluxo de Entrada	IF	O item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro sofreu alteração com relação ao documento OR-0043-2001-0000 para o documento OR-0043-2001-0100 . Foi: Prefixo - passou de tamanho variável, máximo 5 (1 – 99999) para 4 (1 – 9999).
18	2	Índice	MD	O item Processo de importação de LICPINO do documento OR-0043-2001-0000 foi diluído nos Requisitos do Produto (item 4) do documento OR-0043-2001-0100
19	Todo Doc.	Todo Doc.	MD	Em todo documento OR-0043-2001-0100 a denominação SAGRE e SAGRE/Oper foi substituída no documento OR-0043-2001-0000 por SAGRE/Oper – FCT
20	Todo Doc.	Todo Doc.	MD	Em todo documento OR-0043-2001-0100 a denominação LICPINO foi substituída no documento OR-0043-2001-0000 por EQN
Defeitos Novos				
21	6	3.1	O	Não é especificado no documento o termo 'DG' utilizado no requisito.
22	8	4.2	O	Não é descrito como cancelar a solicitação e se é possível cancelar.
23	7	4.1.2	O	Não é especificado o formato do parâmetro de retorno "Diagnóstico".
24	7	4.1.1 4.1.2 4.1.3	O	Não existe descrição da sigla MCDU, utilizada para descrição de informação de I/O.
25	8	4.2	O	Não é descrito o processo de "Mapear contagens" que consta do fluxo SLNS. Ela influencia a importação de licpinos?
26	8	4.2	A	Na expressão "O número de serviço que não constar seu respectivo LICPINO deverá ficar em branco" deixa dúvida sobre o que deve ficar em branco no relatório (o número de serviço ou o LIQPINO?)
27	8	4.2	A	São utilizados os termos "FCT" e "folha de corte" para designar o resultado.
28	8	4.2 e 3.1	IF	A folha de corte impressa sem a informação dos LICPINOS não atende ao objetivo do sistema de associar/importar esse dado da rede interna (no caso de falha de comunicação não faz sentido imprimir sem o LICPINO).
29	8	4.2	O	Não é especificado se existe número máx de números por FCT.
30	8	4.2	O	"vir em branco" através de qual fluxo? RSLNS?
31	8	4.2	EI	Informação de como é feita a associação à folha de corte está omissa.
32	8	4.2	O	Não é especificado o que deve retornar se o sistema localizar mais de um LICPINO para o mesmo número de serviço. Não é especificado se existe correspondência 1 para 1 entre os números de serviços e os LICPINOS.

Defeitos Novos				
Defeito #	Pág.	Req. #	Tipo	Descrição
33	8	4.2	O	A inclusão manual não é especificada como inclusão no relatório ou no sistema para atualização da folha de corte.
34	8	4.2	O	Não descreve o que acontece quando um número de serviço que não existe, ou já presente em outro FCT, é inserido numa folha de corte.
35	8	4.2	O	Se uma SLNS é enviada com algum campo em branco, a resposta é uma RSLNS(erro) ou um número de serviço em branco?
36	8	4.2	O	Como o sistema reconhece uma informação incorreta? Como diferenciar de uma não localizada?
37	8	4.2	O	Em que consiste a preparação para a impressão da folha de corte?
38	8	4.2	O	Associar manualmente um LICPINO inválido a um nº de serviço
39	8	4.2	AI	Não especifica se a recepção do LICPINO se dá na mesma seq. das solicitações o que gera dúvidas na ação de associação.
40	7	4.1	O	Não é especificado qual deve ser a ordem do envio dos parâmetros SLNS.

4ª Lista de Defeitos

Na quarta lista os defeitos foram organizados e agrupados seguindo a taxonomia e origem dos defeitos (defeitos históricos e defeitos novos). Devido a isso os defeitos receberam nova numeração.

Tabela de Correlação entre as Listas 3 e 4

3ª Lista	4ª Lista	3ª Lista	4ª Lista
1	1-O	21	7-O
2	2-O	22	8-O
3	3-O	23	9-O
4	4-O	24	10-O
5	5-O	25	11-O
6	6-O	26	1-IA
7	1-II	27	2-IA
8	2-II	28	10-FI
9	1-FI	29	12-O
10	2-FI	30	13-O
11	3-FI	31	1-IE
12	4-FI	32	14-O
13	5-FI	33	15-O
14	6-FI	34	16-O
15	7-FI	35	17-O
16	8-FI	36	18-O
17	9-FI	37	19-O
18	1-DD	38	20-O
19	2-DD	39	3-IA
20	3-DD	40	21-O

Além das alterações mencionadas acima, as seguintes modificações foram efetuadas:

- A redação do defeito 2 da terceira lista (defeito 2-O na quarta lista) foi modificada para caracterizar o defeito e não a solução dada.
- Os defeitos 3, 4, 5 e 6 (3-O, 4-O, 5-O e 6-O, respectivamente), defeitos 9, 10, 11, 12, 13, 14, 15, 16 e 17 (1-FI, 2-FI, 3-FI, 4-FI, 5-FI, 6-FI, 7-FI, 8-FI e 9-FI, respectivamente) e defeitos 18, 19 e 20 da terceira lista (1-DD, 2-DD e 3-DD, respectivamente) também tiveram suas redações modificadas para caracterizar o problema e não a solução adotada.
- Foram acrescentados 3 novos defeitos históricos (4-IA, 3-II e 4-DD), originados da comparação entre as versões intermediárias do documento de especificação de requisitos.

De acordo com a taxonomia, os defeitos da segunda lista ficaram distribuídos da seguinte forma:

Omissão	Informação Inconsistente	Fato Incorreto	Informação Estranha	Ambigüidade	Defeitos Diversos
21	3	10	1	4	4

4ª Lista

Defeito #	Pág.	Req. #	Tipo	Descrição
Defeitos Históricos - Omissão				
1-O	07	6	O	Não foi explicitado no diagrama de Requisitos do Produto do documento OR-0043-2001-0000 a delimitação da Emissão de FCT
2-O	08	6.1.2	O	Nada foi apresentado no documento sobre o que acontece caso o sistema de controle da rede interna não localize o EQN de um determinado número de serviço.
3-O	08	6.1.2	O	Nada foi apresentado no documento sobre o que acontece no caso do SAGRE não verificar a informação da central no Banco de Dados.
4-O	08	6.1.3	O	Nada foi apresentado no documento sobre o que acontece no caso das informações enviadas pelo SAGRE/Oper-FCT não estiverem corretas, se deverá ser enviada alguma informação com o erro ocorrido.
5-O	08	6.1.3	O	Os Códigos de erro não foram especificados.
6-O	08	6.1.3	O	Nada foi apresentado no item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro sobre o que deverá ocorrer no caso de uma falha de comunicação.
Defeitos Novos – Omissão				
7-O	6	4	O	Não é especificado no documento o termo 'DG' utilizado no requisito.
8-O	8	6.2	O	Não é descrito como cancelar a solicitação e se é possível cancelar.
9-O	8	6.1.2	O	Não é especificado o formato do parâmetro de retorno "Diagnóstico".
10-O	8/9	6.1.1 6.1.2 6.1.3	O	Não existe descrição da sigla MCDU, utilizada para descrição de informação de I/O.
11-O	8	6.2	O	Não é descrito o processo de "Mapear contagens" que consta do fluxo SLNS. Ela influencia a importação de licpinos?
12-O	9	6.2	O	Não é especificado se existe número máx de números por FCT.
13-O	9	6.2	O	"vir em branco" através de qual fluxo? RSLNS?
14-O	9	6.2	O	Não é especificado o que deve retornar se o sistema localizar mais de um LICPINO para o mesmo número de serviço. Não é especificado se existe correspondência 1 para 1 entre os números de serviços e os LICPINOS.
15-O	9	6.2	O	A inclusão manual não é especificada como inclusão no relatório ou no sistema para atualização da folha de corte.
16-O	9	6.2	O	Não descreve o que acontece quando um número de serviço que não existe, ou já presente em outro FCT, é inserido numa folha de corte.
17-O	8	6.2	O	Se uma SLNS é enviada com algum campo em branco, a resposta é uma RSLNS(erro) ou um número de serviço em branco?
18-O	8	6.2	O	Como o sistema reconhece uma informação incorreta? Como diferenciar de uma não localizada?
19-O	8	6.2	O	Em que consiste a preparação para a impressão da folha de corte?
20-O	8	6.2	O	Associar manualmente um LICPINO inválido a um n° de serviço

Defeito #	Pág.	Req. #	Tipo	Descrição
Defeitos Novos – Informação Ambígua				
1-IA	9	6.2	AI	Na expressão "O número de serviço que não constar seu respectivo LICPINO deverá ficar em branco" deixa dúvida sobre o que deve ficar em branco no relatório (o número de serviço ou o LICPINO?)
2-IA	8	6.2	AI	São utilizados os termos "FCT" e "folha de corte" para designar o resultado.
3-IA	9	6.2	AI	Não especifica se a recepção do LICPINO se dá na mesma seqüência das solicitações o que gera dúvidas na ação de associação.
4-IA	9	6.2	AI	No processo de importação de LICPINO a frase "nenhuma inclusão será permitida em uma área que esteja em folha de corte exceto para pares PARA (destino)." deixa dúvida no que significa pares PARA.
Defeitos Históricos – Informação Inconsistente				
1-II	2/8	Índice/ 6.1.2	II	O item RSLNS – Resposta da solicitação de LICPINO de número de serviço do documento OR-0043-2001-0000 está colocado como um item de entrada, e o próprio nome diz que o item é de resposta
2-II	2/8	Índice/ 6.1.3	II	O item RSLNS (ERRO) – Resposta da solicitação de LICPINO de número de serviço com erro do documento OR-0043-2001-0000 está colocado como um item de entrada, e o próprio nome diz que o item é de resposta.
3-II	8	6.1.3	II	O campo prefixo apresenta tamanho variável, máximo 5 caracteres e apresenta como máscara 9999.
Defeitos Históricos – Defeitos Diversos				
1-DD	2	Índice	MD	As informações apresentadas no item Processo de importação de LICPINO do documento OR-0043-2001-0000 poderiam ser diluídas nos requisitos.
2-DD	Todo Doc.	Todo Doc.	MD	As denominações SAGRE e SAGRE/Oper foram usadas em todo o documento para descrever a mesma coisa.
3-DD	Todo Doc.	Todo Doc.	MD	As denominações LICPINO e EQN foram utilizadas em todo o documento para descrever a mesma coisa.
4-DD	6	4	MD	Na documentação existe controvérsia entre a utilização de LIQPINO (com Q) ou LICPINO (com C).

Defeito #	Pág.	Req. #	Tipo	Descrição
Defeitos Novos – Omissão				
21-O	7	6.1.1	O	Não é especificado qual deve ser a ordem do envio dos parâmetros SLNS.
Defeitos Históricos – Fato Incorreto				
1-FI	7	6.1.1	IF	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Localidade deverá conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
2-FI	7	6.1.1	IF	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Estação telefônica deverá conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos
3-FI	7	6.1.1	IF	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Prefixo deveria conter tamanho variável 4 (1 – 9999) ao invés de tamanho variável, máximo 5 (1 – 99999).
4-FI	8	6.1.2	IF	No item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço, o Localidade deverá conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
5-FI	8	6.1.2	IF	No item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço, o campo Estação telefônica deverá conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
6-FI	8	6.1.2	IF	No item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço, o campo Prefixo deverá conter 4 (1 – 9999) ao invés tamanho variável, máximo 5 (1 – 99999).
7-FI	8	6.1.3	IF	No item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Localidade deverá conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
8-FI	8	6.1.3	IF	No item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Estação Telefônica deverá conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
9-FI	8	6.1.3	IF	No item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Prefixo deverá conter 4 (1 – 9999) ao invés tamanho variável, máximo 5 (1 – 99999).
Defeitos Novos – Fato Incorreto				
10-FI	9	4 e 6.2	IF	A folha de corte impressa sem a informação dos LICPINOS não atende ao objetivo do sistema de associar/importar esse dado da rede interna (no caso de falha de comunicação não faz sentido imprimir sem o LICPINO).
Defeitos Novos – Informação Estranha				
1-IE	8	6.2	EI	Informação de como é feita a associação à folha de corte está omissa.

5ª Lista de Defeitos

As seguintes modificações foram executadas originando a quinta lista de defeitos, a qual possui 41 defeitos, sendo 21 históricos e 20 novos.

- No defeito 2-O, o termo EQN foi substituído por LICPINO para manter a consistência.
- A redação do defeito 3-O foi modificada visando maior clareza.
- Os defeitos 4-O e 5-O da quarta lista foram unidos em um só (4-O na quinta lista), recebendo nova redação, por tratarem do mesmo problema, além disso, o defeito 4-O da quarta lista era inconsistente com o item 6.2 do documento.
- A redação do defeito 6-O da quarta lista foi alterada (5-O na quinta lista), pois contradizia com um parágrafo do item 6.2 do documento de requisitos. O tipo do defeito também foi alterado de Omissão para Defeitos Diversos.
- O defeito 11 da quarta lista foi retirado por não pertencer ao escopo do documento de requisitos.
- O defeito 19 da quarta lista também não pertencia ao escopo do documento de requisitos, então teve sua redação alterada para ser considerado defeito (17-O na quinta lista). Sua classificação na taxonomia passou de Omissão para Informação Estranha.
- A redação do defeito 1-IE da quarta lista também foi adaptada para que permanecesse como defeito.

Tabela de Correlação entre as Listas 4 e 5

4ª Lista	5ª Lista	4ª Lista	5ª Lista
1-O	1-O	2-FI	2-FI
2-O	2-O	3-FI	3-FI
3-O	3-O	4-FI	4-FI
4-O	4-O	5-FI	5-FI
5-O	4-O	6-FI	6-FI
6-O	5-O	7-FI	7-FI
7-O	6-O	8-FI	8-FI
8-O	7-O	9-FI	9-FI
9-O	8-O	10-FI	10-FI
10-O	9-O	1-IE	1-IE
11-O	Retirado	1-IA	1-IA
12-O	10-O	2-IA	2-IA
13-O	11-O	3-IA	3-IA
14-O	12-O	4-IA	4-IA
15-O	13-O	1-II	1-II
16-O	14-O	2-II	2-II
17-O	15-O	3-II	3-II
18-O	16-O	1-DD	1-DD
19-O	17-O	2-DD	2-DD
20-O	18-O	3-DD	3-DD
21-O	19-O	4-DD	4-DD
1-FI	1-FI		

De acordo com a taxonomia, os defeitos da segunda lista ficaram distribuídos da seguinte forma:

Omissão	Informação Inconsistente	Fato Incorreto	Informação Estranha	Ambigüidade	Defeitos Diversos
19	10	3	1	4	4

5ª Lista

Defeito #	Pág.	Req. #	Tipo	Descrição
Defeitos Históricos – Omissão				
1-O	7	6	O	Não foi explicitado no diagrama de Requisitos do Produto do documento OR-0043-2001-0000 a delimitação da Emissão de FCT
2-O	8	6.1.2	O	Nada foi apresentado no documento sobre o que acontece caso o sistema de controle da rede interna não localize o LICPINO de um determinado número de serviço.
3-O	8	6.1.2	O	Não foi apresentado no documento o fato de que o SAGRE/Oper-FCT não deve verificar a informação da central no Banco de Dados.
4-O	8	6.1.3	O	Não é informado como o erro vem identificado no diagnóstico de falha.
5-O	9	6.2	MD	O procedimento para quando ocorrer falha de comunicação na solicitação de LICPINO deveria estar especificado no item "RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro".
Defeitos Novos – Omissão				
6-O	6	4	O	Não é especificado no documento o termo 'DG' utilizado no requisito.
7-O	8	6.2	O	Não é descrito como cancelar a solicitação e se é possível cancelar.
8-O	8	6.1.2	O	Não é especificado o formato do parâmetro de retorno "Diagnóstico".
9-O	8/9	6.1.1 6.1.2 6.1.3	O	Não existe descrição da sigla MCDU, utilizada para descrição de informação de I/O.
10-O	9	6.2	O	Não é especificado se existe número máximo de números de serviço por FCT.
11-O	9	6.2	O	"vir em branco" através de qual fluxo? RSLNS?
12-O	9	6.2	O	Não é especificado o que deve retornar se o sistema localizar mais de um LICPINO para o mesmo número de serviço. Não é especificado se existe correspondência 1 para 1 entre os números de serviços e os LICPINOS.
13-O	9	6.2	O	A inclusão manual não é especificada como inclusão no relatório ou no sistema para atualização da folha de corte.
14-O	9	6.2	O	Não descreve o procedimento para quando um número de serviço que não existe, ou já presente em outro FCT, é inserido numa folha de corte, através da inclusão manual.
15-O	8	6.2	O	Se uma SLNS é enviada com algum campo em branco, a resposta é uma RSLNS(erro) ou um número de serviço em branco?
16-O	8	6.2	O	Como o sistema reconhece uma informação incorreta? Como diferenciar de uma não localizada?
17-O	8	6.2	IE	A preparação para a impressão folha de corte está fora do escopo dessa especificação.
18-O	8	6.2	O	Associar manualmente um LICPINO inválido a um nº de serviço
19-O	7	6.1.1	O	Não é especificado qual deve ser a ordem do envio dos parâmetros SLNS.
Defeitos Históricos – Fato Incorreto				
1-FI	7	6.1.1	IF	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Localidade deverá conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.

Defeito #	Pág.	Req. #	Tipo	Descrição
Defeitos Históricos – Fato Incorreto				
2-FI	7	6.1.1	IF	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Estação telefônica deverá conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos
3-FI	7	6.1.1	IF	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Prefixo deveria conter tamanho variável 4 (1 – 9999) ao invés de tamanho variável, máximo 5 (1 – 99999).
4-FI	8	6.1.2	IF	No item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço, o Localidade deverá conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
5-FI	8	6.1.2	IF	No item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço, o campo Estação telefônica deverá conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
6-FI	8	6.1.2	IF	No item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço, o campo Prefixo deverá conter 4 (1 – 9999) ao invés tamanho variável, máximo 5 (1 – 99999).
7-FI	8	6.1.3	IF	No item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Localidade deverá conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
8-FI	8	6.1.3	IF	No item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Estação Telefônica deverá conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
9-FI	8	6.1.3	IF	No item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Prefixo deverá conter 4 (1 – 9999) ao invés tamanho variável, máximo 5 (1 – 99999).
Defeitos Novos – Fato Incorreto				
10-FI	9	4 e 6.2	IF	A folha de corte impressa sem a informação dos LICPINOS não atende ao objetivo do sistema de associar/importar esse dado da rede interna (no caso de falha de comunicação não faz sentido imprimir sem o LICPINO).
Defeitos Novos – Informação Estranha				
1-IE	8	6.2	EI	A associação dos LICPINOS à folha de corte está fora do escopo dessa especificação
Defeitos Novos – Informação Ambígua				
1-IA	9	6.2	AI	Na expressão "O número de serviço que não constar seu respectivo LICPINO deverá ficar em branco" deixa dúvida sobre o que deve ficar em branco no relatório (o número de serviço ou o LICPINO?)
2-IA	8	6.2	AI	São utilizados os termos "FCT" e "folha de corte" para designar o resultado.
3-IA	9	6.2	AI	Não especifica se a recepção do LICPINO se dá na mesma seqüência das solicitações o que gera dúvidas na ação de associação.

Defeito #	Pág.	Req. #	Tipo	Descrição
Defeitos Novos – Informação Ambígua				
4-IA	9	6.2	AI	No processo de importação de LICPINO a frase "nenhuma inclusão será permitida em uma área que esteja em folha de corte exceto em pares PARA (destino)." deixa dúvida no que significa pares PARA.
Defeitos Históricos – Informação Inconsistente				
1-II	2/8	Índice/ 6.1.2	II	O item RSLNS – Resposta da solicitação de LICPINO de número de serviço do documento OR-0043-2001-0000 está colocado como um item de entrada, e o próprio nome diz que o item é de resposta
2-II	2/8	Índice/ 6.1.3	II	O item RSLNS (ERRO) – Resposta da solicitação de LICPINO de número de serviço com erro do documento OR-0043-2001-0000 está colocado como um item de entrada, e o próprio nome diz que o item é de resposta.
3-II	7/8	6.1.1 / 6.1.3	II	O campo prefixo apresenta tamanho variável, máximo 5 caracteres e apresenta como máscara 9999.
Defeitos Históricos – Defeitos Diversos				
1-DD	2	6.2	MD	As informações apresentadas no item Processo de importação de LICPINO do documento OR-0043-2001-0000 poderiam ser diluídas nos requisitos.
2-DD	Todo Doc.	Todo Doc.	MD	As denominações SAGRE e SAGRE/Oper foram usadas em todo o documento para descrever a mesma coisa.
3-DD	Todo Doc.	Todo Doc.	MD	As denominações LICPINO e EQN foram utilizadas em todo o documento para descrever a mesma coisa.
4-DD	6	4	MD	Na documentação existe controvérsia entre a utilização de LIQPINO (com Q) ou LICPINO (com C).

6ª Lista de Defeitos

As alterações que originaram a sexta lista de defeitos foram as seguintes:

- O defeito 2-O da quinta lista mudou para o tipo Defeitos Diversos.
- O defeito 5-O da quinta lista passou a ser o 5-DD na sexta lista, pois já estava classificado como tipo MD, mas ainda permanecia na lista de Omissão.
- As redações dos defeitos 11-O e 16-O, da quinta lista, foram modificadas visando maior clareza.
- O defeito 17-O da quinta lista passou a ser o 2-IE na sexta lista, pois já estava classificado como IE, mas ainda permanecia entre os defeitos de Omissão.
- O defeito 3-IA da quinta lista mudou de classificação, passando a ser do tipo Omissão.
- O defeito 4-IA da quinta lista mudou de numeração porque estava entre os novos defeitos e é um defeito histórico.
- O defeito 3-II da quinta lista foi desmembrado em dois defeitos, porque ocorre em dois requisitos distintos.
- O defeito 4-DD da quinta lista foi retirado por se tratar de problemas de grafia.

A sexta lista consta de 41 defeitos, sendo 22 defeitos históricos e 19 defeitos novos.

Tabela de Correlação entre as Listas 5 e 6

5ª Lista	6ª Lista	5ª Lista	6ª Lista
1-O	1-O	3-FI	3-FI
2-O	4-DD	4-FI	4-FI
3-O	2-O	5-FI	5-FI
4-O	3-O	6-FI	6-FI
5-O	5-DD	7-FI	7-FI
6-O	4-O	8-FI	8-FI
7-O	5-O	9-FI	9-FI
8-O	6-O	10-FI	10-FI
9-O	7-O	1-IE	1-IE
10-O	8-O	1-IA	2-IA
11-O	9-O	2-IA	3-IA
12-O	10-O	3-IA	17-O
13-O	11-O	4-IA	1-IA
14-O	12-O	1-II	1-II
15-O	13-O	2-II	2-II
16-O	14-O	3-II	3-II / 4-II
17-O	2-IE	1-DD	1-DD
18-O	15-O	2-DD	2-DD
19-O	16-O	3-DD	3-DD
1-FI	1-FI	4-DD	Retirado
2-FI	2-FI		

6ª Lista

Defeito #	Pág.	Req. #	Tipo	Descrição
Defeitos Históricos – Omissão				
1-O	07	6	O	Não foi explicitada no diagrama de Requisitos do Produto do documento OR-0043-2001-0000 a delimitação da Emissão de FCT
2-O	08	6.1.2	O	Não foi apresentado no documento o fato de que o SAGRE/Oper-FCT não deve verificar a informação da central no Banco de Dados.
3-O	08	6.1.3	O	Não é informado como o erro vem identificado no diagnóstico de falha.
Defeitos Novos – Omissão				
4-O	6	4	O	Não é especificado no documento o termo 'DG'.
5-O	8	6.2	O	Não é descrito como cancelar a solicitação e se é possível cancelar.
6-O	8	6.1.2	O	Não é especificado o formato do parâmetro de retorno "Diagnóstico".
7-O	8/9	6.1.1 6.1.2 6.1.3	O	Não existe descrição da sigla MCDU.
8-O	9	6.2	O	Não é especificado se existe número máximo de números de serviço por FCT.
9-O	9	6.2	O	Não é especificado através de qual dos dois fluxos de retorno ocorre o "vir em branco".
10-O	9	6.2	O	Não é especificado o que deve retornar se o sistema localizar mais de um LICPINO para o mesmo número de serviço.
11-O	9	6.2	O	A inclusão manual não é especificada como inclusão no relatório ou no sistema para atualização da folha de corte.
12-O	9	6.2	O	Não descreve o procedimento para quando um número de serviço que não existe, ou que já esteja presente em outra FCT, seja inserido numa folha de corte, através da inclusão manual.
13-O	8	6.2	O	Se uma SLNS é enviada com algum campo em branco, e por isso o LICPINO não é localizado não fica claro se a resposta é uma RSLNS(erro) ou um número de serviço em branco.
14-O	8	6.2	O	Não fica claro qual a diferença entre uma informação incorreta e uma não localizada.
15-O	8	6.2	O	A associação manual pode possibilitar que um LICPINO inválido seja associado a um nº de serviço.
16-O	7	6.1.1	O	Não é especificado qual deve ser a ordem do envio dos parâmetros SLNS.
17-O	9	6.2	O	Não especifica se a recepção do LICPINO se dá na mesma seqüência das solicitações o que gera dúvidas na ação de associação.
Defeitos Históricos – Fato Incorreto				
1-FI	7	6.1.1	IF	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Localidade deveria conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
2-FI	7	6.1.1	IF	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Estação telefônica deveria conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos

Defeito #	Pág.	Req. #	Tipo	Descrição
Defeitos Históricos – Fato Incorreto				
3-FI	7	6.1.1	IF	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Prefixo deveria conter tamanho variável 4 (1 – 9999) ao invés de tamanho variável, máximo 5 (1 – 99999).
4-FI	8	6.1.2	IF	No item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço, o Localidade deveria conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
5-FI	8	6.1.2	IF	No item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço, o campo Estação telefônica deveria conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
6-FI	8	6.1.2	IF	No item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço, o campo Prefixo deveria conter 4 (1 – 9999) ao invés tamanho variável, máximo 5 (1 – 99999).
7-FI	8	6.1.3	IF	No item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Localidade deveria conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
8-FI	8	6.1.3	IF	No item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Estação Telefônica deveria conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
9-FI	8	6.1.3	IF	No item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Prefixo deveria conter 4 (1 – 9999) ao invés tamanho variável, máximo 5 (1 – 99999).
Defeitos Novos – Fato Incorreto				
10-FI	9	4 e 6.2	IF	A folha de corte impressa sem a informação dos LICPINOS não atende ao objetivo do sistema de associar/importar esse dado da rede interna (no caso de falha de comunicação não faz sentido imprimir sem o LICPINO).
Defeitos Novos – Informação Estranha				
1-IE	8	6.2	EI	A associação dos LICPINOS à folha de corte está fora do escopo dessa especificação
2-IE	8	6.2	EI	A preparação para a impressão folha de corte está fora do escopo dessa especificação.
Defeitos Históricos – Informação Ambígua				
1-IA	9	6.2	A	No processo de importação de LICPINO a frase "nenhuma inclusão será permitida em uma área que esteja em folha de corte exceto em pares PARA (destino)." deixa dúvida no que significa pares PARA.
Defeitos Novos – Informação Ambígua				
2-IA	9	6.2	A	Na expressão "O número de serviço que não constar seu respectivo LICPINO deverá ficar em branco" deixa dúvida sobre o que deve ficar em branco no relatório (o número de serviço ou o LICPINO?)
3-IA	8	6.2	A	São utilizados os termos "FCT" e "folha de corte" para designar a mesma coisa.

Defeito #	Pág.	Req. #	Tipo	Descrição
Defeitos Históricos – Informação Inconsistente				
1-II	2/8	Índice/ 6.1.2	II	O item RSLNS – Resposta da solicitação de LICPINO de número de serviço do documento OR-0043-2001-0000 está colocado como um item de entrada, mas o próprio nome diz que o item é de resposta.
2-II	2/8	Índice/ 6.1.3	II	O item RSLNS (ERRO) – Resposta da solicitação de LICPINO de número de serviço com erro do documento OR-0043-2001-0000 está colocado como um item de entrada, mas o próprio nome diz que o item é de resposta.
3-II	7	6.1.1	II	O campo prefixo do item 6.1.1 está descrito como tendo tamanho variável, máximo 5 caracteres e apresenta como máscara 9999.
4-II	8	6.1.3	II	O campo prefixo do item 6.1.3 está descrito como tendo tamanho variável, máximo 5 caracteres e apresenta como máscara 9999.
Defeitos Históricos – Defeitos Diversos				
1-DD	2	6.2	MD	As informações apresentadas no item Processo de importação de LICPINO do documento OR-0043-2001-0000 deveriam estar diluídas nos requisitos.
2-DD	Todo Doc.	Todo Doc.	MD	As denominações SAGRE e SAGRE/Oper foram usadas em todo o documento para descrever a mesma coisa.
3-DD	Todo Doc.	Todo Doc.	MD	As denominações LICPINO e EQN foram utilizadas em todo o documento para descrever a mesma coisa.
4-DD	08	6.2	MD	O procedimento sobre o que acontece caso o sistema de controle da rede interna não localize o LICPINO de um determinado número de serviço deveria estar no item "RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro".
5-DD	9	6.2	MD	O procedimento para quando ocorrer falha de comunicação na solicitação de LICPINO deveria estar especificado no item "RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro".

7ª Lista de Defeitos

A sétima lista de defeitos é composta de 33 defeitos, sendo 21 defeitos históricos e 12 defeitos novos. Foram criadas também uma lista de falsos-positivos (todos os relatos que o revisor é levado a relatar devido a um defeito verdadeiro) e uma lista de não defeitos (defeitos retirados da lista de defeitos, após um consenso do grupo, acompanhados de uma justificativa para a retirada).

As alterações efetuadas para a criação da sétima lista foram:

- Os defeitos 2-O, 16-O e 17-O da sexta lista foram retirados.
- Os defeitos 8-O, 11-O, 12-O, 15-O, 10-FI, 1-IE e 2-IE foram retirados da lista e acrescentados na lista de falsos-positivos (FP1, FP2, FP3, FP4, FP6, FP7 e FP8, respectivamente).
- Foram acrescentados dois novos defeitos que haviam sido detectados no Projeto Piloto 1 (11-O e 12-O).
- Os defeitos 2-DD e 3-DD, da sexta lista, passaram a ser do tipo II (5-II e 6-II).

Tabela de Correlação entre as Listas 6 e 7

6ª Lista	7ª Lista	6ª Lista	7ª Lista
1-O	1-O	4-FI	4-FI
2-O	Retirado	5-FI	5-FI
3-O	2-O	6-FI	6-FI
4-O	3-O	7-FI	7-FI
5-O	4-O	8-FI	8-FI
6-O	5-O	9-FI	9-FI
7-O	6-O	10-FI	FP6
8-O	FP1	1-IE	FP7
9-O	7-O	2-IE	FP8
10-O	8-O	1-IA	1-IA
11-O	FP2	2-IA	2-IA
12-O	FP3	3-IA	3-IA
13-O	9-O	1-II	1-II
14-O	10-O	2-II	2-II
15-O	FP4	3-II	3-II
16-O	Retirado	4-II	4-II
17-O	Retirado	1-DD	1-DD
1-FI	1-FI	2-DD	5-II
2-FI	2-FI	3-DD	6-II
3-FI	3-FI	4-DD	2-DD
		5-DD	3-DD

7ª Lista

Defeito #	Pág.	Req. #	Tipo	Descrição
Defeitos Históricos – Omissão				
1-O	07	6	O	Não foi explicitada no diagrama de Requisitos do Produto do documento OR-0043-2001-0000 a delimitação da Emissão de FCT
2-O	08	6.1.3	O	Não é informado como o erro vem identificado no diagnóstico de falha.
Defeitos Novos – Omissão				
3-O	6	4	O	Não é especificado no documento o termo 'DG'.
4-O	8	6.2	O	Não é descrito como cancelar a solicitação e se é possível cancelar.
5-O	8	6.1.2	O	Não é especificado o formato do parâmetro de retorno "Diagnóstico".
6-O	8/9	6.1.1 6.1.2 6.1.3	O	Não existe descrição da sigla MCDU.
7-O	9	6.2	O	Não é especificado através de qual dos dois fluxos de retorno ocorre o "vir em branco".
8-O	9	6.2	O	Não é especificado o que deve retornar se o sistema localizar mais de um LICPINO para o mesmo número de serviço.
9-O	8	6.2	O	Se uma SLNS é enviada com algum campo em branco, e por isso o LICPINO não é localizado não fica claro se a resposta é uma RSLNS(erro) ou um número de serviço em branco.
10-O	8	6.2	O	Não fica claro qual a diferença entre uma informação incorreta e uma não localizada.
11-O	8	6.1.3	O	Não é especificado o formato do parâmetro "Diagnóstico de Falha".
12-O	9	6.2	O	Não é descrito como identificar uma falha de comunicação.
Defeitos Históricos – Fato Incorreto				
1-FI	7	6.1.1	IF	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Localidade deveria conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
2-FI	7	6.1.1	IF	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Estação telefônica deveria conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos
3-FI	7	6.1.1	IF	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Prefixo deveria conter tamanho variável 4 (1 – 9999) ao invés de tamanho variável, máximo 5 (1 – 99999).
4-FI	8	6.1.2	IF	No item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço, o Localidade deveria conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
5-FI	8	6.1.2	IF	No item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço, o campo Estação telefônica deveria conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
6-FI	8	6.1.2	IF	No item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço, o campo Prefixo deveria conter 4 (1 – 9999) ao invés tamanho variável, máximo 5 (1 – 99999).

Defeito #	Pág.	Req. #	Tipo	Descrição
Defeitos Históricos – Fato Incorreto				
7-FI	8	6.1.3	IF	No item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Localidade deveria conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
8-FI	8	6.1.3	IF	No item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Estação Telefônica deveria conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
9-FI	8	6.1.3	IF	No item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Prefixo deveria conter 4 (1 – 9999) ao invés tamanho variável; máximo 5 (1 – 99999).
Defeitos Históricos – Informação Ambígua				
1-IA	9	6.2	A	No processo de importação de LICPINO a frase "nenhuma inclusão será permitida em uma área que esteja em folha de corte exceto em pares PARA (destino)." deixa dúvida no que significa pares PARA.
Defeitos Novos – Informação Ambígua				
2-IA	9	6.2	A	Na expressão "O número de serviço que não constar seu respectivo LICPINO deverá ficar em branco" deixa dúvida sobre o que deve ficar em branco no relatório (o número de serviço ou o LICPINO?)
3-IA	8	6.2	A	São utilizados os termos "FCT" e "folha de corte" para designar a mesma coisa.
Defeitos Históricos – Informação Inconsistente				
1-II	2/8	Índice/ 6.1.2	II	O item RSLNS – Resposta da solicitação de LICPINO de número de serviço do documento OR-0043-2001-0000 está colocado como um item de entrada, mas o próprio nome diz que o item é de resposta.
2-II	2/8	Índice/ 6.1.3	II	O item RSLNS (ERRO) – Resposta da solicitação de LICPINO de número de serviço com erro do documento OR-0043-2001-0000 está colocado como um item de entrada, mas o próprio nome diz que o item é de resposta.
3-II	7/8	6.1.1	II	O campo prefixo do item 6.1.1 está descrito como tendo tamanho variável, máximo 5 caracteres e apresenta como máscara 9999.
4-II	7/8	6.1.3	II	O campo prefixo do item 6.1.3 está descrito como tendo tamanho variável, máximo 5 caracteres e apresenta como máscara 9999.
5-II	Todo Doc.	Todo Doc.	II	As denominações SAGRE e SAGRE/Oper foram usadas em todo o documento para descrever a mesma coisa.
6-II	Todo Doc.	Todo Doc.	II	As denominações LICPINO e EQN foram utilizadas em todo o documento para descrever a mesma coisa.
Defeitos Históricos – Defeitos Diversos				
1-DD	2	6.2	MD	As informações apresentadas no item Processo de importação de LICPINO do documento OR-0043-2001-0000 deveriam estar diluídas nos requisitos.

Defeito #	Pág.	Req. #	Tipo	Descrição
Defeitos Históricos – Defeitos Diversos				
2-DD	08	6.2	MD	O procedimento sobre o que acontece caso o sistema de controle da rede interna não localize o LICPINO de um determinado número de serviço deveria estar no item "RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro".
3-DD	9	6.2	MD	O procedimento para quando ocorrer falha de comunicação na solicitação de LICPINO deveria estar especificado no item "RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro".

LISTA DE FALSOS-POSITIVOS

Documento de Especificação de Requisitos de Importação de LICPINO na FCT

FP #	Pág.	Req. #	Tipo	Descrição	Justificativa
1	9	6.2	O	Não é especificado se existe número máximo de números de serviço por FCT.	É relatado como defeito devido à falta de delimitação no diagrama.
2	9	6.2	O	A inclusão manual não é especificada como inclusão no relatório ou no sistema para atualização da folha de corte.	É relatado como defeito devido à falta de delimitação no diagrama.
3	9	6.2	O	Não descreve o procedimento para quando um número de serviço que não existe, ou que já esteja presente em outra FCT, seja inserido numa folha de corte, através da inclusão manual.	É relatado como defeito devido à falta de delimitação no diagrama.
4	8	6.2	O	A associação manual pode possibilitar que um LICPINO inválido seja associado a um nº de serviço.	É relatado como defeito devido à falta de delimitação no diagrama.
5	9	6.2	O	Não é descrito o processo "Mapear Contagem" que consta do fluxo SLNS.	É relatado como defeito devido à falta de delimitação no diagrama.
6	9	4 e 6.2	IF	A folha de corte impressa sem a informação dos LICPINOS não atende ao objetivo do sistema de associar/importar esse dado da rede interna (no caso de falha de comunicação não faz sentido imprimir sem o LICPINO).	É relatado como defeito devido à falta de delimitação no diagrama.
7	8	6.2	O	A informação de como é feita a associação dos LICPINOS à folha de corte está omissa.	É relatado como defeito devido à falta de delimitação no diagrama.
8	8	6.2	O	Não especifica em que consiste a preparação para a impressão da folha de corte.	É relatado como defeito devido à falta de delimitação no diagrama.

Retirados da Lista de Defeitos

#	Pag.	Req. #	Tipo	Descrição	Justificativa
2-O	08	6.1.2	O	Não foi apresentado no documento o fato de que o SAGRE/Oper-FCT não deve verificar a informação da central no Banco de Dados.	Foi retirado porque em nenhum lugar do documento é dito que o BD deve ser verificado.
16-O	7	6.1.1	O	Não é especificado qual deve ser a ordem do envio dos parâmetros SLNS.	No item 2.2 está dito que não será detalhada a interface com outros sistemas.
17-O	9	6.2	O	Não especifica se a recepção do LICPINO se dá na mesma seqüência das solicitações o que gera dúvidas na ação de associação.	No item 2.2 está dito que não será detalhada a interface com outros sistemas.

8ª Lista de Defeitos

A oitava lista de defeitos é composta de 44 defeitos, sendo 23 defeitos históricos e 21 defeitos novos.

Foram acrescentados doze defeitos novos (13-O ao 21-O, 7-II, 4-DD e 5-DD) e dois falsos-positivos (FP9 e FP10) detectados no Projeto Piloto 2.

As alterações efetuadas para a criação da oitava lista foram:

- Foram acrescentados doze defeitos novos (13-O ao 21-O, 7-II, 4-DD e 5-DD) e dois falsos-positivos (FP9 e FP10) detectados no Projeto Piloto 2.
- O defeito 2-IA da sétima lista foi retirado.
O defeito 3-IA (sétima lista) foi classificado como Informação Inconsistente (6-II), o 5-II foi classificado como Fato Incorreto (10-FI) e o 3-DD como Informação Inconsistente (5-II).

Tabela de Correlação entre as Listas 7 e 8

7ª Lista	8ª Lista	7ª Lista	8ª Lista
1-O	1-O	6-FI	6-FI
2-O	2-O	7-FI	7-FI
3-O	3-O	8-FI	8-FI
4-O	4-O	9-FI	9-FI
5-O	5-O	1-IA	1-IA
6-O	6-O	2-IA	Retirado
7-O	7-O	3-IA	6-II
8-O	8-O	1-II	1-II
9-O	9-O	2-II	2-II
10-O	10-O	3-II	3-II
11-O	11-O	4-II	4-II
12-O	12-O	5-II	10-FI
1-FI	1-FI	6-II	5-II
2-FI	2-FI	1-DD	1-DD
3-FI	3-FI	2-DD	2-DD
4-FI	4-FI	3-DD	3-DD
5-FI	5-FI		

8ª Lista

Defeito #	Pág.	Req. #	Tipo	Descrição
Defeitos Históricos – Omissão				
1-O	07	6	O	Não foi explicitada no diagrama de Requisitos do Produto do documento OR-0043-2001-0000 a delimitação da Emissão de FCT
2-O	08	6.1.3	O	Não é informado como o erro vem identificado no diagnóstico de falha.
Defeitos Novos – Omissão				
3-O	6	4	O	Não é especificado no documento o termo 'DG'.
4-O	8	6.2	O	Não é descrito como cancelar a solicitação e se é possível cancelar.
5-O	8	6.1.2	O	Não é especificado o formato do parâmetro de retorno "Diagnóstico".
6-O	8/9	6.1.1 6.1.2 6.1.3	O	Não existe descrição da sigla MCDU.
7-O	9	6.2	O	Não é especificado através de qual dos dois fluxos de retorno ocorre o "vir em branco".
8-O	9	6.2	O	Não é especificado o que deve retornar se o sistema localizar mais de um LICPINO para o mesmo número de serviço.
9-O	8	6.2	O	Se uma SLNS é enviada com algum campo em branco, e por isso o LICPINO não é localizado não fica claro se a resposta é uma RSLNS (ERRO) ou um número de serviço em branco.
10-O	8	6.2	O	Não fica claro qual a diferença entre uma informação incorreta e uma não localizada.
11-O	8	6.1.3	O	Não é especificado o formato do parâmetro "Diagnóstico de Falha".
12-O	9	6.2	O	Não é descrito como identificar uma falha de comunicação.
13-O	Todo Doc.	Todo Doc.	O	Não existe descrição da sigla LICPINO
14-O	Todo Doc.	Todo Doc.	O	Não existe descrição da sigla SAGRE/Oper
15-O	5.	2.2	O	Não existe descrição da sigla SAGRE
16-O	5	2.3.3	O	Não existe descrição da sigla SASREQ
17-O	8	6.1.3	O	Não existe descrição da sigla EQN
18-O	8	6.2	O	Não existe descrição da sigla SAGRE/Oper-FCT.
19-O	8	6.2	O	Não existe descrição da sigla FCT.
20-O	9	6.2	O	Não existe descrição da sigla TBD.
21-O	9	6.2 § 4	O	Não especifica a funcionalidade para "tratar a resposta RLSN (ERRO)".
Defeitos Históricos – Fato Incorreto				
1-FI	7	6.1.1	IF	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Localidade deveria conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
2-FI	7	6.1.1	IF	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Estação telefônica deveria conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos
3-FI	7	6.1.1	IF	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Prefixo deveria conter tamanho variável 4 (1 – 9999) ao invés de tamanho variável, máximo 5 (1 – 99999).

Defeito #	Pág.	Req. #	Tipo	Descrição
Defeitos Históricos – Fato Incorreto				
4-FI	8	6.1.2	IF	No item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço, o Localidade deveria conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
5-FI	8	6.1.2	IF	No item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço, o campo Estação telefônica deveria conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
6-FI	8	6.1.2	IF	No item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço, o campo Prefixo deveria conter 4 (1 – 9999) ao invés tamanho variável, máximo 5 (1 – 99999).
7-FI	8	6.1.3	IF	No item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Localidade deveria conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
8-FI	8	6.1.3	IF	No item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Estação Telefônica deveria conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
9-FI	8	6.1.3	IF	No item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Prefixo deveria conter 4 (1 – 9999) ao invés tamanho variável, máximo 5 (1 – 99999).
10-FI	Todo Doc.	Todo Doc.	IF	As denominações SAGRE e SAGRE/Oper foram usadas em todo o documento ao invés de SAGRE/Oper-FCT.
Defeitos Históricos – Informação Ambígua				
1-IA	9	6.2	A	No processo de importação de LICPINO a frase "nenhuma inclusão será permitida em uma área que esteja em folha de corte exceto em pares PARA (destino)" deixa dúvida no que significa pares PARA.
Defeitos Históricos – Informação Inconsistente				
1-II	2/8	Índice/ 6.1.2	II	O item RSLNS – Resposta da solicitação de LICPINO de número de serviço do documento OR-0043-2001-0000 está colocado como um item de entrada, mas o próprio nome diz que o item é de resposta.
2-II	2/8	Índice/ 6.1.3	II	O item RSLNS (ERRO) – Resposta da solicitação de LICPINO de número de serviço com erro do documento OR-0043-2001-0000 está colocado como um item de entrada, mas o próprio nome diz que o item é de resposta.
3-II	7/8	6.1.1	II	O campo prefixo do item 6.1.1 está descrito como tendo tamanho variável, máximo 5 caracteres e apresenta como máscara 9999.
4-II	7/8	6.1.3	II	O campo prefixo do item 6.1.3 está descrito como tendo tamanho variável, máximo 5 caracteres e apresenta como máscara 9999.
5-II	Todo Doc.	Todo Doc.	II	As denominações LICPINO e EQN foram utilizadas em todo o documento para descrever a mesma coisa.
6-II	8	6.2	II	São utilizados os termos "FCT" e "folha de corte" para designar a mesma coisa.

Defeito #	Pág.	Req. #	Tipo	Descrição
Defeitos Novos – Informação Inconsistente				
7-II	8	6.1.1/6.1.2	II	Há uma inconsistência no tamanho do campo prefixo do requisito 6.1.2 com o requisito 6.1.1.
Defeitos Históricos – Defeitos Diversos				
1-DD	2	6.2	MD	As informações apresentadas no item Processo de importação de LICPINO do documento OR-0043-2001-0000 deveriam estar diluídas nos requisitos.
2-DD	08	6.2	MD	O procedimento sobre o que acontece caso o sistema de controle da rede interna não localize o LICPINO de um determinado número de serviço deveria estar no item "RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro".
3-DD	9	6.2	MD	O procedimento para quando ocorrer falha de comunicação na solicitação de LICPINO deveria estar especificado no item "RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro".
Defeitos Novos – Defeitos Diversos				
4-DD	9	6.2	MD	O procedimento para quando o número de serviço não possuir seu respectivo LICPINO deveria estar especificado em requisito específico.
5-DD	9	6.2	MD	O parágrafo 8 "O número de serviço que não constar seu respectivo LICPINO deverá ficar em branco." está redundante com o parágrafo 6.

LISTA DE FALSOS-POSITIVOS

Documento de Especificação de Requisitos de Importação de LICPINO na FCT

FP #	Pág.	Req. #	Tipo	Descrição	Justificativa
1	9	6.2	O	Não é especificado se existe número máximo de números de serviço por FCT.	É relatado como defeito devido à falta de delimitação no diagrama.
2	9	6.2	O	A inclusão manual não é especificada como inclusão no relatório ou no sistema para atualização da folha de corte.	É relatado como defeito devido à falta de delimitação no diagrama.
3	9	6.2	O	Não descreve o procedimento para quando um número de serviço que não existe, ou que já esteja presente em outra FCT, seja inserido numa folha de corte, através da inclusão manual.	É relatado como defeito devido à falta de delimitação no diagrama.
4	8	6.2	O	A associação manual pode possibilitar que um LICPINO inválido seja associado a um nº de serviço.	É relatado como defeito devido à falta de delimitação no diagrama.
5	9	6.2	O	Não é descrito o processo "Mapear Contagem" que consta do fluxo SLNS.	É relatado como defeito devido à falta de delimitação no diagrama.
6	9	4 e 6.2	IF	A folha de corte impressa sem a informação dos LICPINOS não atende ao objetivo do sistema de associar/importar esse dado da rede interna (no caso de falha de comunicação não faz sentido imprimir sem o LICPINO).	É relatado como defeito devido à falta de delimitação no diagrama.

FP #	Pág.	Req. #	Tipo	Descrição	Justificativa
7	8	6.2	O	A informação de como é feita a associação dos LICPINOS à folha de corte está omissa.	É relatado como defeito devido à falta de delimitação no diagrama.
8	8	6.2	O	Não especifica em que consiste a preparação para a impressão da folha de corte.	É relatado como defeito devido à falta de delimitação no diagrama.
9	9	6.2	MD	O último parágrafo da seção 6.2 deveria estar descrito em requisito específico.	É relatado como defeito devido à falta de delimitação no diagrama.
10	5	Geral	O	Não há definição sobre folha de corte, mapear contagens, detalhar facilidades. Apenas aparecem no diagrama.	É relatado como defeito devido à falta de delimitação no diagrama.

Retirados da Lista de Defeitos

#	Pág.	Req. #	Tipo	Descrição	Justificativa
2-O	08	6.1.2	O	Não foi apresentado no documento o fato de que o SAGRE/Oper-FCT não deve verificar a informação da central no Banco de Dados.	Foi retirado porque em nenhum lugar do documento é dito que o BD deve ser verificado.
16-O	7	6.1.1	O	Não é especificado qual deve ser a ordem do envio dos parâmetros SLNS.	No item 2.2 está dito que não será detalhada a interface com outros sistemas.
17-O	9	6.2	O	Não especifica se a recepção do LICPINO se dá na mesma seqüência das solicitações o que gera dúvidas na ação de associação.	No item 2.2 está dito que não será detalhada a interface com outros sistemas.
2-IA	9	6.2	A	Na expressão "O número de serviço que não constar seu respectivo LICPINO deverá ficar em branco" deixa dúvida sobre o que deve ficar em branco no relatório (o número de serviço ou o LICPINO?)	A redação é clara quanto ao sujeito da oração.

9ª Lista de Defeitos

A nona lista de defeitos é composta de 43 defeitos, sendo 20 defeitos históricos e 23 defeitos novos.

Foi acrescentada uma coluna caracterizando o projeto piloto em que cada novo defeito foi detectado.

As alterações efetuadas para a criação da nona lista foram:

- O defeito 10-FI da oitava lista foi classificado como Informação Inconsistente, passando a ser o 6-II da nona lista.
- O defeito 1-IA da oitava lista foi retirado.
- O defeito 6-II da oitava lista passou a ser o 8-II na nona, pois foi um defeito encontrado no Projeto Piloto 1 e estava classificado como defeito histórico.

9ª Lista

Defeito #	Proj. Piloto	Pág.	Req. #	Tipo	Descrição
Defeitos Históricos – Omissão					
1-O	-	07	6	O	Não foi explicitada no diagrama de Requisitos do Produto do documento OR-0043-2001-0000 a delimitação da Emissão de FCT
2-O	-	08	6.1.3	O	Não é informado como o erro vem identificado no diagnóstico de falha.
Defeitos Novos – Omissão					
3-O	P1	6	4	O	Não é especificado no documento o termo 'DG'.
4-O	P1	8	6.2	O	Não é informado se é possível cancelar a solicitação.
5-O	P1	8	6.1.2	O	Não é especificado o formato do parâmetro de retorno "Diagnóstico".
6-O	P1	8/9	6.1.1 6.1.2 6.1.3	O	Não existe descrição da sigla MCDU.
7-O	P1	9	6.2	O	Não é especificado através de qual dos dois fluxos de retorno ocorre o "vir em branco".
8-O	P1	9	6.2	O	Não é especificado o que deve retornar se o sistema localizar mais de um LICPINO para o mesmo número de serviço.
9-O	P1	8	6.2	O	Se uma SLNS é enviada com algum campo em branco, não fica claro se a resposta é uma RSLNS (ERRO) ou um número de serviço em branco.
10-O	P1	8	6.2	O	Não fica claro qual a diferença entre uma informação incorreta e uma não localizada.
11-O	P1	8	6.1.3	O	Não é especificado o formato do parâmetro "Diagnóstico de Falha".
12-O	P1	9	6.2	O	Não é descrito como identificar uma falha de comunicação.
13-O	P2	Todo Doc.	Todo Doc.	O	Não existe descrição da sigla LICPINO
14-O	-	Todo Doc.	Todo Doc.	O	Não existe descrição da sigla SAGRE/Oper
15-O	-	5	2.2	O	Não existe descrição da sigla SAGRE
16-O	-	5	2.3.3	O	Não existe descrição da sigla SASREQ
17-O	P2	8	6.1.3	O	Não existe descrição da sigla EQN.
18-O	-	8	6.2	O	Não existe descrição da sigla SAGRE/Oper-FCT.
19-O	P2	8	6.2	O	Não existe descrição da sigla FCT.
20-O	-	9	6.2	O	Não existe descrição da sigla TBD.

Defeito #	Proj. Piloto	Pág.	Req. #	Tipo	Descrição
Defeitos Novos – Omissão					
21-O	P2	9	6.2 § 4	O	Não especifica a funcionalidade para "tratar a resposta RLSN (ERRO)".
Defeitos Históricos – Fato Incorreto					
1-FI	-	7	6.1.1	IF	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Localidade deveria conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
2-FI	-	7	6.1.1	IF	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Estação telefônica deveria conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos
3-FI	-	7	6.1.1	IF	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Prefixo deveria conter tamanho variável 4 (1 – 9999) ao invés de tamanho variável, máximo 5 (1 – 99999).
4-FI	-	8	6.1.2	IF	No item RLSN – Resposta da solicitação de LICPINO de Número de Serviço, o Localidade deveria conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
5-FI	-	8	6.1.2	IF	No item RLSN – Resposta da solicitação de LICPINO de Número de Serviço, o campo Estação telefônica deveria conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
6-FI	-	8	6.1.2	IF	No item RLSN – Resposta da solicitação de LICPINO de Número de Serviço, o campo Prefixo deveria conter 4 (1 – 9999) ao invés tamanho variável, máximo 5 (1 – 99999).
7-FI	-	8	6.1.3	IF	No item RLSN (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Localidade deveria conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
8-FI	-	8	6.1.3	IF	No item RLSN (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Estação Telefônica deveria conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
9-FI	-	8	6.1.3	IF	No item RLSN (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Prefixo deveria conter 4 (1 – 9999) ao invés tamanho variável, máximo 5 (1 – 99999).
Defeitos Históricos – Informação Inconsistente					
1-II	-	2/8	Índice/ 6.1.2	II	O item RLSN – Resposta da solicitação de LICPINO de número de serviço do documento OR-0043-2001-0000 está colocado como um item de entrada, mas o próprio nome diz que o item é de resposta.
2-II	-	2/8	Índice/ 6.1.3	II	O item RLSN (ERRO) – Resposta da solicitação de LICPINO de número de serviço com erro do documento OR-0043-2001-0000 está colocado como um item de entrada, mas o próprio nome diz que o item é de resposta.

Defeito #	Proj. Piloto	Pág.	Req. #	Tipo	Descrição
Defeitos Históricos – Informação Inconsistente					
3-II	-	7	6.1.1	II	O campo prefixo do item 6.1.1 está descrito como tendo tamanho variável, máximo 5 caracteres e apresenta como máscara 9999.
4-II	-	8	6.1.3	II	O campo prefixo do item 6.1.3 está descrito como tendo tamanho variável, máximo 5 caracteres e apresenta como máscara 9999.
5-II	-	Todo Doc.	Todo Doc.	II	As denominações LICPINO e EQN foram utilizadas em todo o documento para descrever a mesma coisa.
6-II	-	Todo Doc.	Todo Doc.	II	As denominações SAGRE, SAGRE/Oper e SAGRE/Oper-FCT foram usadas indistintamente em todo o documento.
Defeitos Novos – Informação Inconsistente					
7-II	P2	8	6.1.1/6.1.2	II	Há uma inconsistência no tamanho do campo prefixo do requisito 6.1.2 com o requisito 6.1.1.
8-II	P1	8	6.2	II	São utilizados os termos "FCT" e "folha de corte" para designar a mesma coisa.
Defeitos Históricos – Defeitos Diversos					
1-DD	-	8/9	6.2	MD	As informações apresentadas no item Processo de importação de LICPINO do documento OR-0043-2001-0000 deveriam estar diluídas nos requisitos.
2-DD	-	9	6.2 § 6	MD	O procedimento sobre o que acontece caso o sistema de controle da rede interna não localize o LICPINO de um determinado número de serviço deveria estar no item "RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro".
3-DD	-	9	6.2 § 9	MD	O procedimento para quando ocorrer falha de comunicação na solicitação de LICPINO deveria estar especificado no item "RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro".
Defeitos Novos – Defeitos Diversos					
4-DD	P2	9	6.2 § 8	MD	O procedimento para quando o número de serviço não possuir seu respectivo LICPINO deveria estar especificado em requisito específico.
5-DD	P2	9	6.2 § 6 / § 8	MD	O parágrafo 8 "O número de serviço que não constar seu respectivo LICPINO deverá ficar em branco." está redundante com o parágrafo 6.

LISTA DE FALSOS-POSITIVOS
Documento de Especificação de Requisitos de Importação de LICPINO na FCT

FP #	Proj	Pág.	Req. #	Tipo	Descrição	Justificativa
1	P1	9	6.2	O	Não é especificado se existe número máximo de números de serviço por FCT.	É relatado como defeito devido à falta de delimitação no diagrama.
2	P1	9	6.2 § 10	O	A inclusão manual não é especificada como inclusão no relatório ou no sistema para atualização da folha de corte.	É relatado como defeito devido à falta de delimitação no diagrama.
3	P1	9	6.2 § 10	O	Não descreve o procedimento para quando um número de serviço que não existe, ou que já esteja presente em outra FCT, seja inserido numa folha de corte, através da inclusão manual.	É relatado como defeito devido à falta de delimitação no diagrama.
4	P1	9	6.2 § 10	O	A associação manual pode possibilitar que um LICPINO inválido seja associado a um nº de serviço.	É relatado como defeito devido à falta de delimitação no diagrama.
5	P1	9	6.2	O	Não é descrito o processo "Mapear Contagem" que consta do fluxo SLNS.	É relatado como defeito devido à falta de delimitação no diagrama.
6	P1	9	4 e 6.2 § 9	IF	A folha de corte impressa sem a informação dos LICPINOS não atende ao objetivo do sistema de associar/importar esse dado da rede interna (no caso de falha de comunicação não faz sentido imprimir sem o LICPINO).	É relatado como defeito devido à falta de delimitação no diagrama.
7	P1	9	6.2 § 7	O	A informação de como é feita a associação dos LICPINOS à folha de corte está omissa.	É relatado como defeito devido à falta de delimitação no diagrama.
8	P1	9	6.2 § 7	O	Não especifica em que consiste a preparação para a impressão da folha de corte.	É relatado como defeito devido à falta de delimitação no diagrama.
9	P2	9	6.2 § 10	MD	O último parágrafo da seção 6.2 deveria estar descrito em requisito específico.	É relatado como defeito devido à falta de delimitação no diagrama.
10	P2	5	Geral	O	Não há definição sobre folha de corte, mapear contagens, detalhar facilidades. Apenas aparecem no diagrama.	É relatado como defeito devido à falta de delimitação no diagrama.

Retirados da Lista de Defeitos

#	Pág.	Req. #	Tipo	Descrição	Justificativa
2-O	08	6.1.2	O	Não foi apresentado no documento o fato de que o SAGRE/Oper-FCT não deve verificar a informação da central no Banco de Dados.	Foi retirado porque em nenhum lugar do documento é dito que o BD deve ser verificado.
16-O	7	6.1.1	O	Não é especificado qual deve ser a ordem do envio dos parâmetros SLNS.	No item 2.2 está dito que não será detalhada a interface com outros sistemas.
17-O	9	6.2	O	Não especifica se a recepção do LICPINO se dá na mesma seqüência das solicitações o que gera dúvidas na ação de associação.	No item 2.2 está dito que não será detalhada a interface com outros sistemas.
2-IA	9	6.2 § 8	A	Na expressão "O número de serviço que não constar seu respectivo LICPINO deverá ficar em branco" deixa dúvida sobre o que deve ficar em branco no relatório (o número de serviço ou o LICPINO?)	A redação é clara quanto ao sujeito da oração.

10ª Lista de Defeitos

Após a última reunião, na qual foi revisada, junto com a equipe do CPqD, a fase de levantamento dos defeitos do Projeto Piloto 2, foram acrescentados 4 novos defeitos e 2 novos falsos-positivos.

A décima lista de defeitos é composta de 47 defeitos, sendo 20 defeitos históricos e 27 defeitos novos.

10ª Lista

Defeito #	Proj. Piloto	Pág.	Req. #	Tipo	Descrição
Defeitos Históricos – Omissão					
1-O	-	07	6	O	Não foi explicitada no diagrama de Requisitos do Produto do documento OR-0043-2001-0000 a delimitação da Emissão de FCT
2-O	-	08	6.1.3	O	Não são informados quais os tipos de erros que podem ser identificados no diagnóstico de falha.
Defeitos Novos – Omissão					
3-O	P1	6	4	O	Não é especificado no documento o termo 'DG'.
4-O	P1	8	6.2	O	Não é informado se é possível cancelar a solicitação.
5-O	P1	8	6.1.2	O	Não é especificado o formato do parâmetro de retorno "Diagnóstico".
6-O	P1	8/9	6.1.1 6.1.2 6.1.3	O	Não existe descrição da sigla MCDU.
7-O	P1	9	6.2	O	Não é especificado através de qual dos dois fluxos de retorno ocorre o "vir em branco".
8-O	P1	9	6.2	O	Não é especificado o que deve retornar se o sistema localizar mais de um LICPINO para o mesmo número de serviço.
9-O	P1	8	6.2	O	Se uma SLNS é enviada com algum campo em branco, não fica claro se a resposta é uma RSLNS (ERRO) ou um número de serviço em branco.
10-O	P1	8	6.2	O	Não fica claro qual a diferença entre uma informação incorreta e uma não localizada.
11-O	P1	8	6.1.3	O	Não é especificado o formato do parâmetro "Diagnóstico de Falha".
12-O	P1	9	6.2	O	Não é descrito como identificar uma falha de comunicação.
13-O	P2	Todo Doc.	Todo Doc.	O	Não existe descrição da sigla LICPINO
14-O	-	Todo Doc.	Todo Doc.	O	Não existe descrição da sigla SAGRE/Oper
15-O	-	5	2.2	O	Não existe descrição da sigla SAGRE
16-O	-	5	2.3.3	O	Não existe descrição da sigla SASREQ
17-O	P2	8	6.1.3	O	Não existe descrição da sigla EQN.
18-O	-	8	6.2	O	Não existe descrição da sigla SAGRE/Oper-FCT.
19-O	P2	8	6.2	O	Não existe descrição da sigla FCT.
20-O	-	9	6.2	O	Não existe descrição da sigla TBD.
21-O	P2	9	6.2 § 4	O	Não especifica a funcionalidade para "tratar a resposta RSLSN (ERRO)".
22-O	P2	5	2.2	O	Não foram descritas as funcionalidades do sistema.
23-O	P2	6	3.1	O	Não é mencionado quais são os dados do LICPINO.

Defeito #	Proj Piloto	Pág.	Req. #	Tipo	Descrição
Defeitos Históricos – Fato Incorreto					
1-FI	-	7	6.1.1	IF	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Localidade deveria conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
2-FI	-	7	6.1.1	IF	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Estação telefônica deveria conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos
3-FI	-	7	6.1.1	IF	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Prefixo deveria conter tamanho variável 4 (1 – 9999) ao invés de tamanho variável, máximo 5 (1 – 99999).
4-FI	-	8	6.1.2	IF	No item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço, o Localidade deveria conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
5-FI	-	8	6.1.2	IF	No item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço, o campo Estação telefônica deveria conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
6-FI	-	8	6.1.2	IF	No item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço, o campo Prefixo deveria conter 4 (1 – 9999) ao invés tamanho variável, máximo 5 (1 – 99999).
7-FI	-	8	6.1.3	IF	No item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Localidade deveria conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
8-FI	-	8	6.1.3	IF	No item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Estação Telefônica deveria conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.
9-FI	-	8	6.1.3	IF	No item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Prefixo deveria conter 4 (1 – 9999) ao invés tamanho variável, máximo 5 (1 – 99999).
Defeitos Novos – Fato Incorreto					
10-FI	P2	7	6	IF	Nem todos os sistemas de rede interna existentes em Empresas Operadoras têm LICPINO.
11-FI	P2	4	Geral	IF	O documento cita que o cliente é telefônica, mas a lista de distribuição é GVT.
Defeitos Históricos – Informação Inconsistente					
1-II	-	2/8	Índice/ 6.1.2	II	O item RSLNS – Resposta da solicitação de LICPINO de número de serviço do documento OR-0043-2001-0000 está colocado como um item de entrada, mas o próprio nome diz que o item é de resposta.

Defeito #	Proj Piloto	Pág.	Req. #	Tipo	Descrição
Defeitos Históricos – Informação Inconsistente					
2-II	-	2/8	Índice/ 6.1.3	II	O item RSLNS (ERRO) – Resposta da solicitação de LICPINO de número de serviço com erro do documento OR-0043-2001-0000 está colocado como um item de entrada, mas o próprio nome diz que o item é de resposta.
3-II	-	7	6.1.1	II	O campo prefixo do item 6.1.1 está descrito como tendo tamanho variável, máximo 5 caracteres e apresenta como máscara 9999.
4-II	-	8	6.1.3	II	O campo prefixo do item 6.1.3 está descrito como tendo tamanho variável, máximo 5 caracteres e apresenta como máscara 9999.
5-II	-	Todo Doc.	Todo Doc.	II	As denominações LICPINO e EQN foram utilizadas em todo o documento para descrever a mesma coisa.
6-II	-	Todo Doc.	Todo Doc.	II	As denominações SAGRE, SAGRE/Oper e SAGRE/Oper-FCT foram usadas indistintamente em todo o documento.
Defeitos Novos – Informação Inconsistente					
7-II	P2	8	6.1.1/6.1.2	II	Há uma inconsistência no tamanho do campo prefixo do requisito 6.1.2 com o requisito 6.1.1.
8-II	P1	8	6.2	II	São utilizados os termos "FCT" e "folha de corte" para designar a mesma coisa.
Defeitos Históricos – Defeitos Diversos					
1-DD	-	8/9	6.2	MD	As informações apresentadas no item Processo de importação de LICPINO do documento OR-0043-2001-0000 deveriam estar diluídas nos requisitos.
2-DD	-	9	6.2 § 6	MD	O procedimento sobre o que acontece caso o sistema de controle da rede interna não localize o LICPINO de um determinado número de serviço deveria estar no item "RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro".
3-DD	-	9	6.2 § 9	MD	O procedimento para quando ocorrer falha de comunicação na solicitação de LICPINO deveria estar especificado no item "RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço com erro".
Defeitos Novos – Defeitos Diversos					
4-DD	P2	9	6.2 § 8	MD	O procedimento para quando o número de serviço não possuir seu respectivo LICPINO deveria estar especificado em requisito específico.
5-DD	P2	9	6.2 § 6 / § 8	MD	O parágrafo 8 "O número de serviço que não constar seu respectivo LICPINO deverá ficar em branco." esta redundante com o parágrafo 6.

LISTA DE FALSOS-POSITIVOS
Documento de Especificação de Requisitos de Importação de LICPINO na FCT

FP #	Proj	Pag.	Req. #	Tipo	Descrição	Justificativa
1	P1	9	6.2	O	Não é especificado se existe número máximo de números de serviço por FCT.	É relatado como defeito devido à falta de delimitação no diagrama.
2	P1	9	6.2 § 10	O	A inclusão manual não é especificada como inclusão no relatório ou no sistema para atualização da folha de corte.	É relatado como defeito devido à falta de delimitação no diagrama.
3	P1	9	6.2 § 10	O	Não descreve o procedimento para quando um número de serviço que não existe, ou que já esteja presente em outra FCT, seja inserido numa folha de corte, através da inclusão manual.	É relatado como defeito devido à falta de delimitação no diagrama.
4	P1	9	6.2 § 10	O	A associação manual pode possibilitar que um LICPINO inválido seja associado a um nº de serviço.	É relatado como defeito devido à falta de delimitação no diagrama.
5	P1	9	6.2	O	Não é descrito o processo "Mapear Contagem" que consta do fluxo SLNS.	É relatado como defeito devido à falta de delimitação no diagrama.
6	P1	9	4 e 6.2 § 9	IF	A folha de corte impressa sem a informação dos LICPINOS não atende ao objetivo do sistema de associar/importar esse dado da rede interna (no caso de falha de comunicação não faz sentido imprimir sem o LICPINO).	É relatado como defeito devido à falta de delimitação no diagrama.
7	P1	9	6.2 § 7	O	A informação de como é feita a associação dos LICPINOS à folha de corte está omissa.	É relatado como defeito devido à falta de delimitação no diagrama.
8	P1	9	6.2 § 7	O	Não especifica em que consiste a preparação para a impressão da folha de corte.	É relatado como defeito devido à falta de delimitação no diagrama.
9	P2	9	6.2 § 10	MD	O último parágrafo da seção 6.2 deveria estar descrito em requisito específico.	É relatado como defeito devido à falta de delimitação no diagrama.
10	P2	5	Geral	O	Não há definição sobre folha de corte, mapear contagens, detalhar facilidades. Apenas aparecem no diagrama.	É relatado como defeito devido à falta de delimitação no diagrama.
11	P2	5	2.3.2	O	Falta definir no item de Conceitos o assunto folha de corte.	É relatado como defeito devido à falta de delimitação no diagrama.
12	P2	9	6.2	O	A descrição do requisito não está de acordo com o diagrama. Não há um fluxo representando "falha na comunicação".	

Retirados da Lista de Defeitos

#	Pág.	Req. #	Tipo	Descrição	Justificativa
2-O	08	6.1.2	O	Não foi apresentado no documento o fato de que o SAGRE/Oper-FCT não deve verificar a informação da central no Banco de Dados.	Foi retirado porque em nenhum lugar do documento é dito que o BD deve ser verificado.
16-O	7	6.1.1	O	Não é especificado qual deve ser a ordem do envio dos parâmetros SLNS.	No item 2.2 está dito que não será detalhada a interface com outros sistemas.
17-O	9	6.2	O	Não especifica se a recepção do LICPINO se dá na mesma seqüência das solicitações o que gera dúvidas na ação de associação.	No item 2.2 está dito que não será detalhada a interface com outros sistemas.
2-IA	9	6.2 § 8	A	Na expressão "O número de serviço que não constar seu respectivo LICPINO deverá ficar em branco" deixa dúvida sobre o que deve ficar em branco no relatório (o número de serviço ou o LICPINO?)	A redação é clara quanto ao sujeito da oração.

Caracterização dos Defeitos

Relação entre questões e erros encontrados por tipo

Legenda

TDD - Totalmente dependente do domínio
 DD - Dependente do domínio
 PDD - Pouco dependente do domínio
 ID - independente do domínio

MD - Muito difícil
 D - Difícil
 MF - Muito fácil
 F - Fácil

MP - Muito Provável
 P - Provável
 PP - Pouco Provável
 I - Improvável

C - Facilidade para pessoas com conhecimento do domínio; S - Facilidade para as pessoas sem conhecimento do domínio.

OMISSÃO

	Questões - visão User													Domínio				Facilidade				Probabilidade				
	1.1	1.2	1.3	1.4	2.1	2.2	2.3	2.4	2.5	3.1	3.2	3.3	3.4	TDD	DD	PDD	ID	MD	D	F	MF	MP	P	PP	I	
	Defeitos Históricos													Defeitos Históricos												
1-0					X					X				X				S		C					X	
2-0					X					X						X				SC		X			X	
	Defeitos Novos													Defeitos Novos												
3-0					X												X				SC	X				
4-0								X		X					X					S	C				X	X
5-0					X												X				SC				X	
6-0					X												X				SC	X			X	
7-0										X				X						SC					X	
8-0								X		X				X				S	C						X	
9-0					X									X						SC					X	
10-0					X			X									X			SC					X	X
11-0					X					X					X					SC					X	
12-0					X					X							X				SC				X	
13-0					X												X				SC	X			X	

14-O					X											X		SC	X	X		
15-O					X											X		SC	X	X		
16-O					X											X		SC	X	X		
17-O					X											X		SC	X	X		
18-O					X											X		SC	X	X		
19-O					X											X		SC	X	X		
20-O					X											X		SC	X	X		
21-O					X											X		SC	X	X		
																						X

Questões - visão Tester

Domínio

Facilidade

Probabilidade

1.1	1.2	1.3	1.4	1.5	2.1	2.2	2.3	2.4	2.5	3.1	4.1	4.2	4.3	TDD	DD	PDD	ID	MD	D	F	MF	MP	P	PP	I
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	----	-----	----	----	---	---	----	----	---	----	---

Defeitos Históricos

Defeitos Históricos

1-O			X			X								X				S		C			X		X	
2-O			X			X										X					SC		X		X	

Defeitos Novos

Defeitos Novos

3-O								X									X				SC	X			
4-O			X									X											X		
5-O			X						X								X				SC	X			
6-O			X							X							X				SC	X			
7-O			X								X				X						SC	X			X
8-O			X									X		X				X		C			X		
9-O								X							X						SC	X			X

Omissão

FATO INCORRETO																										
Fato Incorreto	Questões - visão User													Domínio				Facilidade				Probabilidade				
	1.1	1.2	1.3	1.4	2.1	2.2	2.3	2.4	2.5	3.1	3.2	3.3	3.4	TDD	DD	PDD	ID	MD	D	F	MF	MP	P	PP	I	
	Defeitos Históricos													Defeitos Históricos												
1-FI												X		X				S			C			X		
2-FI												X		X				S			C			X		
3-FI												X		X				S			C			X		
4-FI												X		X				S			C			X		
5-FI												X		X				S			C			X		
6-FI												X		X				S			C			X		
7-FI												X		X				S			C			X		
8-FI												X		X				S			C			X		
9-FI												X		X				S			C			X		
Fato Incorreto	Questões - visão Tester													Domínio				Facilidade				Probabilidade				
	1.1	1.2	1.3	1.4	1.5	2.1	2.2	2.3	2.4	2.5	3.1	4.1	4.2	4.3	TDD	DD	PDD	ID	MD	D	F	MF	MP	P	PP	I
	Defeitos Históricos													Defeitos Históricos												
1-FI						X									X			S			C			X		
2-FI						X									X			S			C			X		
3-FI						X									X			S			C			X		
4-FI						X									X			S			C			X		
5-FI						X									X			S			C			X		
6-FI						X									X			S			C			X		
7-FI						X									X			S			C			X		
8-FI						X									X			S			C			X		
9-FI						X									X			S			C			X		

INFORMAÇÃO INCONSISTENTE																										
Fato Incorreto	Questões - visão User													Domínio				Facilidade				Probabilidade				
	1.1	1.2	1.3	1.4	2.1	2.2	2.3	2.4	2.5	3.1	3.2	3.3	3.4	TDD	DD	PDD	ID	MD	D	F	MF	MP	P	PP	I	
	Defeitos Históricos													Defeitos Históricos												
1-II			X													X					SC			X		
						X																		X		
2-II			X													X						SC			X	
						X																		X		
3-II			X													X						SC			X	
			X																					X		
4-II						X										X								X		
																								X		
5-II	X															X					SC		X		X	
	X																						X		X	
6-II																X					SC		X		X	

Informação Inconsis

Defeitos Novos															Defeitos Novos										
Questões - visão Tester															Dominio				Facilidade				Probabilidade		
1.1	1.2	1.3	1.4	1.5	2.1	2.2	2.3	2.4	2.5	3.1	4.1	4.2	4.3	TDD	DD	PDD	ID	MD	D	F	MF	MP	P	PP	I
Defeitos Históricos															Defeitos Históricos										
1-II				X					X			X				X				SC		X			
2-II				X					X			X				X				SC		X			
3-II				X					X			X				X				SC		X			
4-II				X					X			X				X				SC		X			
5-II				X					X			X				X				SC		X			
6-II				X					X			X				X				SC		X			
Defeitos Novos															Defeitos Novos										
7-II				X					X			X				X				SC		X			
8-II				X					X			X				X				SC		X			

FALSO POSITIVO

Questões - visão User					Domínio	Facilidade	Probabilidade
FP5				X	X	SC	X
							X
FP1	X				X	SC	X
				X			X
FP2			X		X	S C	X
				X			X
FP3			X		X	S C	X
				X			X
FP8			X		X	SC	X
				X			X
FP4					X	C S	X
				X			X
FP6		X			X		X
				X			X
FP7	X		X		X	C S	X
				X			X
Questões - visão Tester					Domínio	Facilidade	Probabilidade
FP5			X		X	SC	X
				X			X
FP1			X		X	SC	X
				X			X
FP2			X				X
				X			X
FP3			X				X
				X			X
FP8			X				X
				X			X
FP4	X		X				X
				X			X
FP6	X				X	C S	X
				X			X
FP7			X		X	C S	X

Vínculo dos Defeitos com as Questões

TABULAÇÃO DOS RESULTADOS DO PROCESSO DE VALIDAÇÃO DE DOCUMENTOS DE REQUISITOS

COORDENADOR: Priscilla B. B. Pagliuso

DATA: 24/03/2003

OMISSÃO DE INFORMAÇÃO

Nº	Pág.	Def.	Descrição	Questões Associadas
Defeitos Históricos				
1-O	7	O	Não foi explicitada no diagrama de Requisitos do Produto do documento OR-0043-2001-0000 a delimitação da Emissão de FCT.	Q1.4, Q4.1 e Q2.1 (T) e Q2.2, Q3.1 e Q3.2 (U)
2-O	8	O	Não é informado como o erro vem identificado no diagnóstico de falha.	Q1.4, Q4.1, Q2.1 (T) e Q2.2, Q3.2 e Q3.1 (U)
Defeitos Novos				
3-O	6	O	Não é especificado no documento o termo 'DG'.	Q1.4 e Q2.4 (T) e Q2.2 (U)
4-O	8	O	Não é informado se é possível cancelar a solicitação.	Q4.1 (T) e Q3.1, Q2.5, Q3.2 (U)
5-O	8	O	Não é especificado o formato do parâmetro de retorno "Diagnóstico".	Q1.4, Q2.4, Q2.5 (T) e Q2.2, Q3.1 (U)
6-O	8/9	O	Não existe descrição da sigla MCDU.	Q1.4 e Q2.4 (T) e Q2.2 e Q3.1 (U)
7-O	9	O	Não é especificado através de qual dos dois fluxos de retorno ocorre o "vir em branco".	Q1.4, Q2.5, Q2.4 (T) e Q3.1, Q3.4 (U)
8-O	9	O	Não é especificado o que deve retornar se o sistema localizar mais de um LICPINO para o mesmo número de serviço.	Q1.4, Q2.4, Q2.5, Q4.1 (T) e Q2.5, Q3.1 (U)
9-O	8	O	Se uma SLNS é enviada com algum campo em branco, não fica claro se a resposta é uma RSLNS (ERRO) ou um número de serviço em branco.	Q2.5, Q2.4, Q4.1 (T) e Q2.2, Q2.4 (U)
10-O	8	O	Não fica claro qual a diferença entre uma informação incorreta e uma não localizada.	Q1.4, Q2.4, Q4.1 (T) e Q2.2, Q2.4, Q2.5 (U)
11-O	8	O	Não é especificado o formato do parâmetro de retorno "Diagnóstico de Falha".	Q1.4, Q2.4, Q4.1 (T) e Q2.2, Q3.1 (U)
12-O	9	O	Não é descrito como identificar uma falha de comunicação.	Q1.4, Q2.4, Q2.5, Q4.1 (T) e Q2.2, Q3.1, Q3.2 (U)
13-O	Todo Doc.	O	Não existe descrição da sigla LICPINO.	Q1.4, Q2.4 (T) e Q2.2, Q3.1 (U)
14-O	Todo Doc.	O	Não existe descrição da sigla SAGRE/Oper.	Q1.4, Q2.4 (T) e Q2.2, Q3.1 (U)

15-O	5	O	Não existe descrição da sigla SAGRE.	Q1.4, Q2.4 (T) e Q2.2, Q3.1 (U)
16-O	5	O	Não existe descrição da sigla SASREQ.	Q1.4, Q2.4 (T) e Q2.2, Q3.1 (U)
17-O	8	O	Não existe descrição da sigla EQN.	Q1.4, Q2.4 (T) e Q2.2, Q3.1 (U)
18-O	8	O	Não existe descrição da sigla SAGRE/Oper-FCT.	Q1.4, Q2.4 (T) e Q2.2, Q3.1 (U)
19-O	8	O	Não existe descrição da sigla FCT.	Q1.4, Q2.4 (T) e Q2.2, Q3.1 (U)
20-O	9	O	Não existe descrição da sigla TBD.	Q1.4, Q2.4 (T) e Q2.2, Q3.1 (U)
21-O	9	O	Não especifica a funcionalidade para "tratar a resposta RSLSN (ERRO)".	Q1.4, Q2.4, Q2.5, Q4.1 (T) e Q2.2, Q3.1, Q3.2 (U)
FATO INCORRETO				
Nº	Pág.	Def.	Descrição	Questão Utilizada
Defeitos Históricos				
1-FI	7	FI	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Localidade deveria conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.	Q2.1 (T) e Q3.4 (U)
2-FI	7	FI	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Estação telefônica deveria conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.	Q2.1 (T) e Q3.4 (U)
3-FI	7	FI	No item SLNS – solicitação de LICPINO de Número de Serviço, o campo Prefixo deveria conter tamanho variável 4 (1 – 9999) ao invés de tamanho variável, máximo 5 (1 – 99999).	Q2.1 (T) e Q3.4 (U)
4-FI	7	FI	No item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço, o Localidade deveria conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.	Q2.1 (T) e Q3.4 (U)
5-FI	7	FI	No item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço, o campo Estação telefônica deveria conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.	Q2.1 (T) e Q3.4 (U)
6-FI	7	FI	No item RSLNS – Resposta da solicitação de LICPINO de Número de Serviço, o campo Prefixo deveria conter 4 (1 – 9999) ao invés tamanho variável, máximo 5 (1 – 99999).	Q2.1 (T) e Q3.4 (U)

Nº	Pág.	Def.	Descrição	Questão Utilizada
Defeitos Históricos (cont.)				
7-FI	7	FI	No item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Localidade deveria conter 6 caracteres numéricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.	Q2.1 (T) e Q3.4 (U)
8-FI	7	FI	No item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Estação Telefônica deveria conter 11 caracteres alfanuméricos ao invés de tamanho variável, máximo 5 caracteres alfanuméricos.	Q2.1 (T) e Q3.4 (U)
9-FI	7	FI	No item RSLNS (ERRO) – Resposta da solicitação de LICPINO de Número de Serviço, o campo Prefixo deveria conter 4 (1 – 9999) ao invés tamanho variável, máximo 5 (1 – 99999).	Q2.1 (T) e Q3.4 (U)

INFORMAÇÃO INCONSISTENTE				
Nº	Pág.	Def.	Descrição	Questão Utilizada
Defeitos Históricos				
1-II	2/8	II	O item RSLNS – Resposta da solicitação de LICPINO de número de serviço do documento OR-0043-2001-0000 está colocado como um item de entrada, mas o próprio nome diz que o item é de resposta.	Q1.5,Q2.5,Q4.2,Q4.3 (T) e Q1.3,Q2.2 (U)
2-II	2/8	II	O item RSLNS (ERRO) – Resposta da solicitação de LICPINO de número de serviço com erro do documento OR-0043-2001-0000 está colocado como um item de entrada, mas o próprio nome diz que o item é de resposta.	Q1.5,Q2.5,Q4.2,Q4.3 (T) e Q1.3,Q2.2 (U)
3-II	7	II	O campo prefixo do item 6.1.1 está descrito como tendo tamanho variável, máximo 5 caracteres e apresenta como máscara 9999.	Q1.5,Q2.5,Q4.3 (T) e Q1.3,Q2.2 (U)
4-II	8	II	O campo prefixo do item 6.1.3 está descrito como tendo tamanho variável, máximo 5 caracteres e apresenta como máscara 9999.	Q1.5,Q2.5,Q4.3 (T) e Q1.3,Q2.2 (U)
5-II	Todo doc.	II	As denominações LICPINO e EQN foram utilizadas em todo o documento para descrever a mesma coisa.	Q1.5,Q2.5(T) e Q3.4,Q1.1(U)
6-II	Todo doc.	II	As denominações SAGRE, SAGRE/Oper e SAGRE/Oper-FCT foram usadas indistintamente em todo documento.	Q1.5,Q2.5(T) e Q3.4,Q1.1(U)

Defeitos Novos

7-II	8	II	Há uma inconsistência no tamanho do campo prefixo do requisito 6.1.2 com o requisito 6.1.1.	Q1.5,Q2.5,Q4.3 (T) e Q1.3,Q2.2 (U)
8-II	8	II	São utilizados os termos "FCT" e "folha de corte" para designar a mesma coisa.	Q1.5,Q2.5 (T) e Q1.1,Q3.4(U)

DEFEITOS DIVERSOS

Nº	Pág.	Def.	Descrição	Questão Utilizada
Defeitos Históricos				
1-DD	8/9	DD	As informações apresentadas no item Processo de importação de LICPINO do documento OR-0043-2001-0000 deveriam estar diluídas nos requisitos.	
2-DD	9	DD	O procedimento sobre o que acontece caso o sistema de controle da rede interna não localize o LICPINO de um determinado número de serviço deveria estar no item "RSLNS (ERRO) - Resposta da solicitação de LICPINO de Número de Serviço com erro."	
3-DD	9	DD	O procedimento para quando ocorrer falha de comunicação na solicitação de LICPINO deveria estar especificado no item "RSLNS (ERRO) - Resposta da solicitação de LICPINO de Número de Serviço com erro."	
Defeitos Novos				
4-DD	9	DD	O procedimento para quando o número de serviço não possuir seu respectivo LICPINO deveria estar especificado em requisito específico.	
5-DD	9	DD	O parágrafo 8 "O número de serviço que não constar seu respectivo LICPINO deverá ficar em branco." está redundante com o parágrafo 6.	Q2.5 (T) e Q3.3 (U)