

UNIVERSIDADE FEDERAL DO PARÁ  
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

UM ESTUDO ETNOGRÁFICO SOBRE A ENGENHARIA DE REQUISITOS EM UM  
PROJETO DE DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE

ALLINE DE MELO LEMOS

DM 05/2011

UFPA / ICEN / PPGCC  
Campus Universitário do Guamá  
Belém-Pará-Brasil  
2011



UNIVERSIDADE FEDERAL DO PARÁ  
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ALLINE DE MELO LEMOS

UM ESTUDO ETNOGRÁFICO SOBRE A ENGENHARIA DE REQUISITOS EM UM  
PROJETO DE DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE

DM 05/2011

UFPA / ICEN / PPGCC  
Campus Universitário do Guamá  
Belém-Pará-Brasil  
2011

UNIVERSIDADE FEDERAL DO PARÁ  
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ALLINE DE MELO LEMOS

UM ESTUDO ETNOGRÁFICO SOBRE A ENGENHARIA DE REQUISITOS EM UM  
PROJETO DE DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE

Dissertação submetida à Banca Examinadora do Programa de Pós-Graduação em Ciência da Computação da UFPA para a obtenção do Grau de Mestre em Ciência da Computação.

UFPA / ICEN / PPGCC  
Campus Universitário do Guamá  
Belém-Pará-Brasil  
2011

Lemos, Alline de Melo

Um Estudo Etnográfico sobre a Engenharia de Requisitos em um Projeto de Desenvolvimento de Software / (Alline de Melo Lemos); orientador, Cleidson Ronald Botelho de Souza. – 2011.

160f. il. 28cm.

Dissertação (Mestrado) – Universidade Federal do Pará. Instituto de Ciência Exatas e Naturais. Programa de Pós-Graduação em Ciência da Computação. Belém, 2011.

1. Engenharia de Software. I. Souza, Cleidson R. B., orient. II. Universidade Federal do Pará, Instituto de Ciências Exatas e Naturais, Programa de Pós-Graduação em Ciência da Computação. III. Título.

CDD 22. ed. 005.1

**FOLHA PRONTA E ASSINADA. PEGAR EM CASA E SCANNEAR!**

All of us are watchers – of television, of clocks, of traffic – but few of us are observers.

Everyone is looking; not many are seeing”.

Peter M. Leschak

## **AGRADECIMENTOS**

A Deus, pela vida.

Aos meus pais Ana e Roberto e ao meu irmão Roberto Allan, pelo amor incondicional.

Ao meu orientador, Prof. Dr. Cleidson, pelos sábios conselhos, pela enorme paciência e pelos valiosos ensinamentos.

Aos professores do PPGCC, por contribuírem para o meu crescimento acadêmico.

À banca, Prof. Dr. Rafael Prikladnicki e Profa. Dra. Tayana Conte, por aceitarem o convite para avaliação deste trabalho.

Ao meu namorado, Marcelo, pela paciência, cumplicidade, compreensão e amor dedicados (e pelos jantares preparados com tanto carinho nos últimos dias).

À minha família brasiliense, tia Stela, tio Sérgio, Lorena e Serginho, por me acolherem com tanto carinho em sua casa a cada ida à Brasília para coleta de dados.

Aos amigos, por tornarem essa jornada um pouco menos difícil. Agradecimentos especiais aos amigos Adailton, Jadielly, Joana, Kuroki, Murilo, Patrícia e Pedro pela amizade verdadeira, por me apoiarem em tantos momentos de angústia e por entenderem meu mau humor recorrente nos últimos dias. Aos amigos do LABES, pela convivência agradável desde a graduação até os dias atuais. Agradeço especialmente aos amigos do projeto SIGAP – sinto falta de trabalhar com vocês.

Aos amigos da Cobra Tecnologia, por me proporcionarem um ambiente de trabalho tão divertido. E em especial, à minha coordenadora (e grande amiga), Thayssa Carvalho, por me apoiar principalmente na última semana de escrita da dissertação.

Aos funcionários e clientes da empresa Alpha, por dedicarem seu tempo às entrevistas e pela paciência durante as observações locais e remotas.

# SUMÁRIO

<b>1</b>	<b><u>INTRODUÇÃO .....</u></b>	<b><u>17</u></b>
1.1	JUSTIFICATIVA .....	19
1.2	QUESTÃO DE PESQUISA .....	20
1.3	OBJETIVOS .....	21
1.4	ORGANIZAÇÃO DO TRABALHO .....	21
<b>2</b>	<b><u>FUNDAMENTAÇÃO TEÓRICA.....</u></b>	<b><u>23</u></b>
2.1	ENGENHARIA DE REQUISITOS.....	23
2.1.1	PROCESSO DE ENGENHARIA DE REQUISITOS .....	24
2.2	DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE.....	27
2.3	ENGENHARIA DE REQUISITOS E DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE.....	28
2.3.1	DIVERSIDADE DE CULTURA E DE NEGÓCIO DOS CLIENTES .....	29
2.3.2	POUCA PARTICIPAÇÃO DOS USUÁRIOS DO SISTEMA COM A EQUIPE TÉCNICA .....	29
2.3.3	FALTA DE COMUNICAÇÃO INFORMAL E POUCA PERCEPÇÃO DO CONTEXTO LOCAL DE TRABALHO.....	30
2.3.4	REDUÇÃO DO NÍVEL DE CONFIANÇA.....	30
2.3.5	DIFICULDADE NA GERÊNCIA DE CONFLITOS E NA OCORRÊNCIA DE DISCUSSÕES ENTRE OS INTEGRANTES DA EQUIPE	30
2.3.6	DIFICULDADE NA OBTENÇÃO DE UMA INTERPRETAÇÃO COMUM DOS REQUISITOS.....	31
2.3.7	INEFICIÊNCIA DE REUNIÕES PARA TOMADA DE DECISÃO.....	31
2.3.8	DEMORA NA EXECUÇÃO DAS TAREFAS.....	31
2.4	TRABALHOS CORRELATOS.....	32
2.4.1	METODOLOGIAS .....	32
2.4.2	FERRAMENTAS .....	34
2.4.3	PROCESSOS DE ENGENHARIA DE REQUISITOS.....	36

<b><u>3</u></b>	<b><u>MÉTODO DE PESQUISA</u></b>	<b><u>42</u></b>
<b>3.1</b>	<b>ETNOGRAFIA</b>	<b>42</b>
3.1.1	MOTIVAÇÕES	42
3.1.2	DEFINIÇÃO	44
3.1.3	UM BREVE HISTÓRICO	45
3.1.4	ETNOGRAFIA EM ENGENHARIA DE SOFTWARE	46
3.1.5	TIPOS DE ETNOGRAFIA NA ENGENHARIA DE SOFTWARE	48
<b>3.2</b>	<b>ENTREVISTAS E OBSERVAÇÕES</b>	<b>50</b>
3.2.1	ENTREVISTAS	50
3.2.2	OBSERVAÇÕES	51
<b>3.3</b>	<b>A TEORIA FUNDAMENTADA EM DADOS</b>	<b>52</b>
3.3.1	MOTIVAÇÕES	52
3.3.2	DEFINIÇÃO	52
<b>3.4</b>	<b>O CONTEXTO ORGANIZACIONAL DO ESTUDO ETNOGRÁFICO</b>	<b>54</b>
<b>3.5</b>	<b>COLETA E ANÁLISE DE DADOS</b>	<b>55</b>
<b><u>4</u></b>	<b><u>UM ESTUDO ETNOGRÁFICO NA EMPRESA ALPHA</u></b>	<b><u>62</u></b>
<b>4.1</b>	<b>PONTOS NEGATIVOS</b>	<b>62</b>
4.1.1	PROBLEMAS ENFRENTADOS DURANTE O PROCESSO DE ER	62
4.1.2	PROBLEMAS TRANSVERSAIS AO PROCESSO DE ER	83
<b>4.2</b>	<b>PONTOS POSITIVOS</b>	<b>95</b>
4.2.1	SOLUÇÕES METODOLÓGICAS	96
4.2.2	SOLUÇÕES ORGANIZACIONAIS	113
4.2.3	SOLUÇÕES TECNOLÓGICAS	116
<b><u>5</u></b>	<b><u>DISCUSSÃO</u></b>	<b><u>120</u></b>
<b>5.1</b>	<b>CONSIDERAÇÕES INICIAIS</b>	<b>120</b>
<b>5.2</b>	<b>PONTOS EM QUE O ESTUDO ETNOGRÁFICO CORROBORA OS MODELOS ANALISADOS</b>	<b>122</b>
5.2.1	PLANEJAMENTO DE REUNIÕES DE ELICITAÇÃO	122
5.2.2	ELABORAÇÃO DE PROTÓTIPOS	123
5.2.3	DISPONIBILIZAÇÃO DE INFORMAÇÕES DE CONTEXTO	123
5.2.4	RECONHECIMENTO DA GC COMO UM DESAFIO PARA A ER DISTRIBUÍDA	124

<b>5.3</b>	<b>PONTOS EM QUE O ESTUDO ETNOGRÁFICO CORROBORA AS FERRAMENTAS ANALISADAS</b> .....	<b>125</b>
<b>5.4</b>	<b>PONTOS QUE FICARAM EM ABERTO NOS TRABALHOS ANALISADOS</b> .....	<b>126</b>
5.4.1	FORMAS DE VALIDAÇÃO DE REQUISITOS EM DDS .....	126
5.4.2	TÉCNICAS PARA GERENCIAR REQUISITOS.....	127
5.4.3	REPRESENTAÇÃO VISUAL DOS REQUISITOS .....	128
<b>5.5</b>	<b>OPORTUNIDADES DE MELHORIA EM DDS.....</b>	<b>129</b>
	ADMINISTRADORES DE DADOS (ADs) EM POUCA QUANTIDADE, DISTANTES GEOGRAFICAMENTE DO PÓLO DE DESENVOLVIMENTO E COM TAREFAS DESCOORDENADAS DOS DESENVOLVEDORES. ....	129
	ATENDIMENTO DE DEMANDAS MUITO LONGAS. ....	130
	REALIZAR A ELICITAÇÃO DE REQUISITOS NO PÓLO DE DESENVOLVIMENTO. ....	130
	REALIZAR A HOMOLOGAÇÃO NO PÓLO DE DESENVOLVIMENTO.....	130
	ADOÇÃO DE PONTOS DE CONTROLE COMO PRÉ-HOMOLOGAÇÕES. ....	131
	REALIZAR <i>WORKSHOPS</i> DE REQUISITOS.....	132
	FORMALIZAÇÃO DO PAPEL DE <i>KNOWLEDGE BROKER</i> . ....	133
<b>6</b>	<b><u>CONSIDERAÇÕES FINAIS</u></b> .....	<b>134</b>
6.1	PRINCIPAIS CONTRIBUIÇÕES.....	135
6.2	LIMITAÇÕES DO ESTUDO .....	135
6.3	TRABALHOS FUTUROS .....	136
	<b><u>REFERÊNCIAS BIBLIOGRÁFICAS</u></b> .....	<b>137</b>
	<b><u>APÊNDICE I – GUIAS DE ENTREVISTAS</u></b> .....	<b>144</b>
	<b><u>APÊNDICE II – CÓDIGOS</u></b> .....	<b>154</b>

## LISTA DE ILUSTRAÇÕES

Figura 1 – Processo genérico de Engenharia de Requisitos. ....	24
Figura 2 – Exemplo de Matriz de Rastreabilidade (Pressman, 2004). ....	26
Figura 3 – Modelo de impacto dos desafios e das atividades de ER afetadas em virtude de problemas causados pela diversidade cultural, comunicação inadequada, gerência de conhecimento e diferença de tempo (Adaptado de Damian e Zowghi, 2002). ....	29
Figura 4 – Categorias e subcategorias do modelo Felder – Silverman (Adaptado de Aranda et al, 2006). ....	33
Figura 5 – Distribuição dos engenheiros de requisitos no exemplo de cenário (Lopes, 2004). ....	36
Figura 6 – Processo preliminar de Engenharia de Requisitos para ambientes de DDS (Lopes, 2004). ....	37
Figura 7 – Modelo de processo de Engenharia de Requisitos em DDS (Lopes, 2004). ....	38
Figura 8 – Fluxo principal do processo colaborativo (Brito, 2006). ....	39
Figura 9 – Iteração de Elicitação de Requisitos (Medeiros, 2007). ....	40
Figura 10 – Iteração de Documentação dos requisitos (Medeiros, 2007). ....	40
Figura 11 – Iteração de Validação dos Requisitos (Medeiros, 2007). ....	41
Figura 12 – Tipos de entrevistas (Adaptado de Dewalt e Dewalt, 2002). ....	50
Figura 13 – Passos da teoria fundamentada em dados. ....	53
Figura 14 – Clientes e analistas da empresa Alpha estão separados por uma distância aérea de 1593 km. ....	55
Figura 15 – Utilização do MAXqda2 durante as etapas de codificação da <i>Grounded Theory</i> . ....	59

## **LISTA DE TABELAS**

Tabela 1 – Atividades realizadas durante a coleta de dados na empresa Alpha.....	57
Tabela 2 – Organização dos códigos, categorias e principais categorias. ....	60

## LISTA DE ABREVIATURAS E SIGLAS

AD	Administrador de Dados
BEL	Belém
BSB	Brasília
DDS	Desenvolvimento Distribuído de Software
DRS	Documento de Especificação de Requisitos
DV	Documento de Visão
ER	Engenharia de Requisitos
ES	Engenharia de Software
GC	Gerência do Conhecimento
GT	<i>Grounded Theory</i>
PPGCC	Programa de Pós-Graduação em Ciência da Computação
RJ	Rio de Janeiro
RUP	<i>Rational Unified Process</i>
SP	São Paulo
SW-CMM	<i>Software Capability Maturity Model</i>
UFPA	Universidade Federal do Pará
UML	<i>Unified Modeling Language</i>

## RESUMO

A indústria de software é uma das várias indústrias que tem sofrido alterações com o advento da globalização. A crescente demanda por software e a escassa mão-de-obra qualificada motivaram o surgimento do Desenvolvimento Distribuído de Software. O Desenvolvimento Distribuído de Software caracteriza-se pela distância cultural, geográfica e/ou temporal entre alguns *stakeholders* (cliente, usuário e desenvolvedores, por exemplo) envolvidos no processo de desenvolvimento. Dentre os principais desafios a serem vencidos em projetos distribuídos de software, estão aqueles relacionados à Engenharia de Requisitos.

Este trabalho apresenta os resultados de um estudo etnográfico realizado em uma organização de desenvolvimento de software que elicitava, analisa e gerencia requisitos de clientes situados em locais geograficamente distantes. O estudo foi conduzido utilizando entrevistas não e semi-estruturadas e observações diretas durante um período de um ano, de maneira intercalada. A análise dos dados foi realizada utilizando práticas da Teoria Fundamentada em Dados. Considerando as atividades de requisitos realizadas na empresa estudada, esta dissertação identificou as principais dificuldades enfrentadas pela empresa, bem como as boas práticas executadas e oportunidades de melhorias sugeridas por seus analistas para amenizar o efeito da distância. Além disso, outro resultado desta dissertação é uma discussão que compara os resultados do estudo etnográfico com alguns trabalhos da Engenharia de Requisitos em contextos de Desenvolvimento Distribuído de Software (em termos de processos, metodologias e ferramentas). Esta discussão sugere áreas de pesquisa que podem ser mais bem estudadas e avaliadas para que se tornem eficazes em seu auxílio aos engenheiros de requisitos envolvidos em projetos distribuídos.

**PALAVRAS-CHAVE:** Desenvolvimento Distribuído de Software, Engenharia de Requisitos, Engenharia de Software, Etnografia, Teoria Fundamentada em Dados.

## **ABSTRACT**

Software industry is one of many industries which have been influenced by globalization. Low qualified labor and software increasing demand have motivated the appearance of Global Software Development. Global Software Development is characterized by the cultural, geographic and/or temporal distance among some of the stakeholders (e.g.: customer, user and developers) involved in software development process. One of the main challenges of global software projects is Requirements Engineering.

This work presents the results of an empiric study conducted in a software organization which elicits, analyzes and manages requirements with customers located in many places. The study was conducted using and semi-structured and unstructured interviews and direct observations during one year, interchangeably. Practices from Grounded Theory were used to analyze the collected data. Considering the requirements activities the organization adopts, this thesis identified the main difficulties faced by the organization as well as some of the good practices performed by its analysts in order to diminish the effects of distance. In addition, another result of this thesis is a discussion about the results from this ethnographic study with some studies in Requirements Engineering in global software development contexts. This discussion suggests research areas that require further study and evaluation in order to provide efficient support to requirements engineers in distributed projects.

**KEYWORDS:** Global Software Development, Requirements Engineering, Software Engineering, Ethnography, Grounded Theory.

# 1 INTRODUÇÃO

A indústria de software é uma das várias indústrias que têm sofrido os efeitos da globalização (Herbsleb e Moitra, 2001)(Damian e Moitra, 2006). Percebe-se uma crescente demanda por software de um lado e uma conseqüente escassez de mão-de-obra qualificada de outro. Isto forçou as empresas a buscarem soluções em outros países especialmente, Brasil, Irlanda, Rússia, Índia, Israel e China. Dentro deste contexto, surge o conceito de Desenvolvimento Distribuído de Software (DDS), uma nova situação na qual software é desenvolvido. Um projeto de DDS está caracterizado sempre que pelo menos um dos *stakeholders* envolvidos no processo de desenvolvimento (clientes, usuários ou desenvolvedores) se encontra fisicamente distante dos demais (Audy e Prikladnicki, 2007).

De uma maneira geral, os seguintes fatores podem ser citados como motivações para o DDS (Mockus e Herbsleb, 2001): 1) a necessidade de uma empresa estar próxima de seus clientes; 2) o surgimento de nações como participantes ativas no mercado mundial de software; 3) os benefícios econômicos encontrados nestas nações (mão-de-obra qualificada e de 10 a 30% mais barata), e; 4) a possibilidade de desenvolvimento *follow-the-sun*<sup>1</sup>. Em DDS, entretanto, existe uma série de desafios a serem vencidos, sendo o maior deles a Engenharia de Requisitos (Audy e Prikladnicki, 2007). Segundo Damian e Zowghi (2003), a Engenharia de Requisitos (ou simplesmente, ER) já envolve atividades suficientemente complicadas quando realizadas de forma local. Em ambientes distribuídos, estas atividades se tornam ainda

---

<sup>1</sup> O desenvolvimento *follow-the-sun* ou *round-the-clock* é caracterizado pela distribuição das tarefas de desenvolvimento de software entre regiões com fusos-horário diferentes, de modo a permitir que quando um dia de trabalho se encerra em um local (por exemplo, no Brasil), ele se inicia em outro (por exemplo, no Japão). Desta forma, o desenvolvimento de software nunca pararia.

mais problemáticas, sobretudo em virtude das diferenças culturais, temporais e geográficas enfrentadas pelos integrantes das equipes envolvidas (Damian e Zowghi, 2003).

De forma mais específica, a globalização gera dois grandes desafios à comunidade de pesquisa em Engenharia de Requisitos (Cheng e Atlee, 2007). O primeiro desafio diz respeito à criação ou aprimoramento de técnicas de ER que apoiem as demais atividades envolvidas no desenvolvimento distribuído de software, como a codificação ou o teste do sistema. Isto é necessário porque a distância agrava a lacuna existente entre os analistas de requisitos e os desenvolvedores, sobretudo se são indivíduos que trabalham em organizações situadas em países diferentes. Esta lacuna também pode existir entre analistas de requisitos e seus clientes, exigindo que os primeiros se desloquem ao ambiente de trabalho dos segundos e vice-versa. Uma vez que a comunicação informal entre os stakeholders diminui devido à distância e, desta forma, torna a coordenação de atividades mais difícil (Kraut et al, 1990), aumentam-se as chances de os requisitos serem mal interpretados e, conseqüentemente, de pouca utilidade ao cliente final. O segundo desafio consiste em permitir que as etapas da ER sejam realizadas de forma *eficiente* em ambientes distribuídos. A tendência é que analistas de requisitos lidem cada vez mais com stakeholders distribuídos, assim, estes profissionais necessitam de apoio através de técnicas, ferramentas, metodologias, etc, que facilitem atividades distribuídas como elicitação, documentação e validação de requisitos (Cheng e Atlee, 2007).

Este trabalho descreve a realização de um estudo etnográfico em uma empresa (chamada simplesmente de *Alpha* para preservar sua identidade) em que algumas tarefas da ER são realizadas à distância. Contatos iniciais com a empresa sugeriram que a mesma enfrenta problemas durante o processo de Engenharia de Requisitos porque se encontra localizada distante de seu cliente. A empresa em questão é caracterizada pela prática das etapas da Engenharia de Requisitos com clientes geograficamente separados da equipe responsável pelo desenvolvimento do sistema. Esta dissertação de mestrado considera os dois desafios citados anteriormente: espera-se que o resultado desta pesquisa não só diminua a lacuna existente entre clientes e analistas (quando estes estiverem distantes um do outro), mas também possibilite a realização mais eficiente das tarefas de análise de requisitos (quando realizadas em ambientes de DDS).

Estudos etnográficos requerem que eles sejam conduzidos no ambiente de trabalho das equipes estudadas (McGrath, 1994). Assim, o estudo aqui apresentado foi conduzido no ambiente da própria empresa Alpha. Além disso, dado o contexto distribuído das atividades

de ER, existiu também a necessidade de coletar dados no domínio do cliente desta empresa, localizado em outra cidade. Isto foi um importante aspecto do estudo porque evitou uma das limitações observadas em diversos estudos anteriores: a coleta de dados em apenas um dos sítios. Resultados extraídos de apenas um contexto de trabalho possuem problemas de validade, uma vez que descrevem problemas a partir de uma única ótica: a da equipe responsável pela construção do software (Damian e Zowghi, 2002). Desta forma, a realização de estudos etnográficos que incluam coletas de dados no contexto do cliente é importante porque auxilia na obtenção de resultados com maior validade.

O estudo etnográfico descrito nesta dissertação possuiu, em um primeiro momento, caráter exploratório. Seu principal objetivo era identificar os problemas enfrentados por engenheiros de software envolvidos em projetos de DDS na empresa Alpha. Além das dificuldades, o estudo também identificou as “boas práticas” de trabalho realizadas pela empresa, isto é, as estratégias já utilizadas pelas equipes para minimizar os efeitos causados pela distância. Finalmente, este trabalho confronta os resultados do estudo etnográfico com os trabalhos elencados durante a pesquisa bibliográfica, visando apontar suas limitações e sugerir como os trabalhos pesquisados sobre ER em contextos distribuídos podem avançar a partir dos resultados obtidos.

Os próximos tópicos apresentam a justificativa, o problema tratado e os objetivos deste trabalho.

## **1.1 Justificativa**

A Engenharia de Software é desempenhada por seres humanos. Por este motivo, estudar os aspectos humanos e cooperativos do desenvolvimento de software é considerado tão importante para entender como metodologias e ferramentas são utilizadas por estas pessoas (de Souza et al, 2009). Dentro da Engenharia de software, a disciplina de Engenharia de Requisitos inclui atividades de natureza intensamente comunicativa e colaborativa uma vez que envolve a interação entre diversos *stakeholders* (clientes, usuários e analistas) (Curtis et al, 1988)(Kraut e Streeter, 1995). Estas atividades já são suficientemente críticas em ambientes co-localizados: problemas associados com a ER são apontados como uma das grandes causas para falhas em projetos de software (Pfleegler, 2004), bem como um dos principais problemas em desenvolvimento de software em geral (Curtis et al, 1988). Isto acontece porque a ER é responsável por entender o que o cliente deseja, analisar suas

necessidades, avaliar a exequibilidade das possíveis soluções, negociar uma solução razoável, especificar a solução de maneira não-ambígua, validar a especificação e administrar os requisitos à medida que eles são transformados num sistema em operação (Pressman, 2004). Se os requisitos não são especificados adequadamente, eles serão erroneamente passados de uma etapa para outra ao longo do processo de desenvolvimento de software e resultarão em um sistema que não satisfaz as necessidades de seus usuários. Em poucas palavras, todo o esforço gasto durante o projeto poderá ser desperdiçado, por isso toda a etapa de ER deve ser executada de forma bastante cautelosa.

Com o surgimento do DDS, caracterizado pela distância geográfica entre os *stakeholders* envolvidos no projeto, a ER se tornou uma etapa ainda mais crítica. Decisivamente, a distância geográfica exerce impactos sobre as tarefas de requisitos (Zowghi, 2002), sobretudo no que diz respeito à comunicação. A distância geográfica restringe o contato informal entre os *stakeholders* (Kraut et al, 1990), tornando-o escasso – muitas vezes inexistente – e, com isso: (a) compromete a coordenação dos membros da equipe e desacelera o processo de ER, isto é, os envolvidos demoram mais a chegar ao entendimento comum dos requisitos; (b) permite que o grande volume de conhecimento tratado nesta etapa permaneça tácito e não-documentado; e, (c) dificulta a construção do sentimento de confiança (*trust*) e “de equipe” (*teamness*) entre os membros (Zowghi, 2002).

Os desafios listados acima sugerem que soluções tecnológicas e/ou metodológicas precisam ser estudadas com o objetivo de mitigar os efeitos causados pela distância em ambientes distribuídos de desenvolvimento de software. Dentro deste contexto, Zowghi (2002) recomenda a adaptação do processo tradicional de ER para um processo que atenda as peculiaridades encontradas em contextos de DDS.

## 1.2 Questão de pesquisa

Considerando a contextualização apresentada na seção anterior, o principal objetivo desta dissertação é realizar um estudo qualitativo em uma empresa local que realiza atividades de ER de forma distribuída para, a partir deste, indicar novas direções de pesquisa a partir da avaliação de abordagens existentes com dados reais e identificar limitações e boas práticas para a execução da ER em contextos distribuídos.

Particularmente, a empresa Alpha é caracterizada pela prática de etapas de Engenharia de Requisitos com clientes geograficamente separados dos desenvolvedores do sistema.

Portanto, a questão de pesquisa a ser tratada no decorrer deste trabalho é a seguinte: *“Como a distância geográfica influencia as atividades de Engenharia de Requisitos quando estas são desempenhadas por equipes geograficamente distribuídas?”*

### **1.3 Objetivos**

O objetivo principal deste trabalho é realizar um estudo etnográfico em uma empresa local que realiza atividades de ER de forma distribuída para, a partir deste, analisar alguns trabalhos sobre a ER em ambientes de DDS, apontar possíveis limitações e sugerir como esta área de pesquisa pode avançar a partir dos resultados encontrados. Este objetivo principal está subdividido em objetivos específicos. Dentre eles estão:

- Identificar os problemas enfrentados pelos engenheiros de software durante a realização de suas tarefas de Engenharia de Requisitos;
- Identificar as boas práticas utilizadas pelos profissionais da empresa sobre como realizar ER em contextos distribuídos; e
- Avançar o conhecimento na Engenharia de Requisitos em contextos de Desenvolvimento Distribuído de Software através da avaliação das abordagens existentes a partir dos dados do estudo etnográfico conduzido.

### **1.4 Organização do trabalho**

Este trabalho está organizado em seis capítulos. O capítulo 2 apresenta a teoria sobre a qual este trabalho está fundamentado. Esta revisão bibliográfica inclui trabalhos nas linhas de pesquisa de Engenharia de Requisitos e Desenvolvimento Distribuído de Software que são correlatos ao trabalho aqui proposto.

O capítulo 3 mostra a metodologia utilizada no decorrer deste trabalho, apresentando conceitos como: Etnografia, entrevistas e observações, e os métodos da Teoria Fundamentada em Dados que foram utilizados para a análise dos dados coletados durante o estudo etnográfico. Além disso, são apresentados também informações sobre o contexto no qual o estudo etnográfico foi realizado, bem como a descrição da coleta e da análise dos dados.

O próximo capítulo ilustra o estudo etnográfico realizado, apresentando os resultados obtidos a partir da análise dos dados coletados e classificando-os em pontos negativos e pontos positivos.

O capítulo 5 propõe uma discussão em que se relacionam os resultados obtidos do estudo etnográfico ilustrado no capítulo 4 e a revisão bibliográfica apresentada no capítulo 2 de modo a analisar alguns trabalhos sobre a ER em contextos de DDS e como os resultados encontrados podem auxiliar em seu avanço.

Finalmente, o capítulo 6 aborda considerações como as principais contribuições do trabalho, suas limitações e possibilidades de estudos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 Engenharia de Requisitos

Segundo Nuseibeh e Easterbrook (2000), a Engenharia de Requisitos é a parte da Engenharia de Software que se preocupa em descobrir o propósito de um software através da identificação dos clientes e usuários finais e de suas necessidades, bem como da documentação desta informação de forma a ser facilmente analisada, compreendida e implementada nas fases seguintes do desenvolvimento de software.

Um projeto de desenvolvimento de software, antes de iniciado, deve ser planejado. Finkelstein categoriza este planejamento como contexto e alicerces, isto é, um projeto de desenvolvimento de software deve ser planejado de acordo com o *contexto* no qual ocorrerá, considerando *fundamentos* importantes da Engenharia de Requisitos (Finkelstein, 1993).

Atualmente, a ER é realizada em uma multiplicidade de *contextos*. Há casos em que os softwares são desenvolvidos sob encomenda para um cliente específico. Ou, o desenvolvimento do produto pode ser orientado ao mercado, isto é, o software é desenvolvido de acordo com uma necessidade do mercado. Ou ainda, o produto é construído para atender as necessidades de um cliente em específico, mas já com a intenção de disponibilizá-lo ao mercado. O tipo do software a ser produzido também influencia as atividades de ER. Por exemplo, a Engenharia de Requisitos realizada para desenvolver sistemas de informação é diferente daquela para sistemas embarcados que, por sua vez, é diferente em se tratando de serviços genéricos como redes ou sistemas operacionais. Diante disto, a ER deve ser adaptada de acordo com o contexto em que o software será utilizado e construído e o tipo do software a ser desenvolvido.

Além do contexto, a preparação do projeto deve considerar alguns *fundamentos* sobre os quais a ER deve estar apoiada, são eles: (a) avaliar a viabilidade de um projeto e dos riscos a ele associados são atividades apoiadas pela ER; (b) estimativas de custo, de cronograma e de viabilidade técnica podem ser obtidas a partir de um documento preciso de especificação de requisitos; e, finalmente, (c) a disciplina de requisitos deve ser apoiada por um *processo de ER adequado* e, para isto, métodos e técnicas devem ser selecionados para apoiar as diversas atividades de ER.

No contexto da Engenharia de Software, o termo processo de software significa um arcabouço para as atividades necessárias à construção de software com qualidade (Pressman, 2004). De forma mais detalhada, um processo de desenvolvimento de software define: (a) quais atividades devem ser realizadas; (b) em que ordem estas atividades devem ocorrer; (c) quais artefatos são consumidos e/ou gerados a partir delas; (d) quem são as pessoas responsáveis por executar estas tarefas, e, finalmente; (e) quais técnicas e/ou ferramentas apóiam estas atividades (Pressman, 2004). Restringindo o conceito para a área de Engenharia de Requisitos, um processo de ER define as características acima e que são necessárias a identificação, análise, negociação, documentação, validação e manutenção dos requisitos.

### 2.1.1 Processo de Engenharia de Requisitos

Segundo Pressman (2004), um processo genérico de Engenharia de Requisitos (Figura 1) normalmente é composto pelas atividades de elicitação, modelagem, negociação, documentação, validação e gestão de requisitos. Esta última é considerada uma atividade guarda-chuva já que é realizada de forma transversal às demais atividades. Estas atividades serão descritas nos parágrafos seguintes (Pressman, 2004).

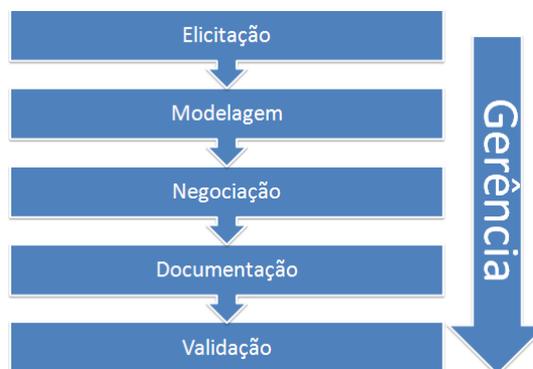


Figura 1 – Processo genérico de Engenharia de Requisitos.

Normalmente, a *elicitação de requisitos* é a primeira etapa de um processo de ER. Nesta fase ocorrem as conversas entre analistas e clientes, usuários e outros interessados para descobrir quais são as funcionalidades do sistema, os problemas que devem ser solucionados pelo sistema e quais restrições este sistema possui. Existem inúmeras técnicas para elicitação de requisitos: técnicas tradicionais (questionários, entrevistas, análise de documentação da organização), técnicas em grupo (sessões de *brainstorming*, discussões de grupo – do inglês, *focus group*), técnicas contextuais (etnografia), de prototipagem, entre outras (Nuseibeh e Easterbrook, 2000).

Uma vez que os requisitos são coletados, o analista deve organizá-los na forma de um modelo de análise. A etapa de *modelagem dos requisitos* no desenvolvimento orientado a objetos trata da construção de diagramas UML e permite que os analistas visualizem o sistema como um todo para que, assim, sejam capazes de avaliar e relacionar os componentes uns com os outros.

Freqüentemente, alguns dos requisitos coletados e analisados certamente serão conflitantes. Isto acontece ou porque o cliente pede mais do que a equipe pode desenvolver, isto é, sem levar em conta as restrições do projeto, ou porque os próprios clientes e usuários do sistema não estão de acordo quanto aos requisitos do sistema. Neste momento, os analistas devem ser capazes de conciliar estes requisitos, ou seja, negociá-los. A *negociação* consiste em organizar os requisitos por ordem de prioridade. Feito isso, a equipe de desenvolvimento, clientes e usuários chegam juntos a um consenso, excluindo, combinando e/ou modificando requisitos. O importante é que cada parte saia razoavelmente satisfeita com o resultado obtido do processo de negociação.

Depois que os requisitos são negociados, o analista registra estes requisitos na forma de um documento de especificação. Este documento descreve os requisitos tanto em linguagem natural, quanto gráfica, podendo esta última parte estar relacionada aos próprios diagramas previamente elaborados. A *especificação de requisitos* contém as funções (requisitos funcionais) e os aspectos de qualidade (requisitos não-funcionais) do sistema. Este artefato serve ainda para estabelecer de forma precisa o que será implementado pela equipe de desenvolvimento e o que não será desenvolvido em função de alguma restrição tecnológica, por exemplo. É importante que isto seja documentado para que clientes e analistas definam claramente qual o escopo do sistema, estejam cientes da responsabilidade um do outro e, assim, possam trabalhar de forma harmoniosa.

A próxima etapa de um processo genérico de ER é a *validação dos requisitos*. Os engenheiros de requisitos, junto aos clientes e aos usuários, avaliam o documento elaborado com o intuito de detectar problemas de ambigüidade, incompletude, inconsistência, erros e omissões. Neste momento, os *stakeholders* se preocupam novamente em averiguar se existem requisitos conflitantes ou ainda inatingíveis. Todo este cuidado é tomado para que a equipe de desenvolvimento construa o sistema de acordo com as reais necessidades do cliente.

O processo de ER não acaba quando a especificação de requisitos é validada e entregue ao cliente. Pode-se entender que o processo de ER é iterativo e reinicia cada vez que o cliente pede alteração ou inserção de um novo requisito.

Finalmente, de forma transversal a todas as etapas anteriormente descritas, os analistas devem realizar ainda atividades de gestão de requisitos, sendo elas: identificar, controlar e rastrear requisitos e modificações de requisitos. Em outras palavras, a *gerência de requisitos* consiste em identificar de forma única os requisitos coletados e reuni-los em uma “matriz de rastreabilidade” (Figura 2). Esta matriz relaciona os requisitos com uma série de outros aspectos do sistema. Alguns exemplos de matrizes são: a matriz de rastreamento de dependências que mostra como os requisitos se relacionam uns com os outros; e a matriz de rastreamento de interface que mostra a relação entre requisitos e as interfaces de um sistema. A matriz de rastreabilidade é um mecanismo importante, pois permite que o engenheiro de requisitos visualize, por exemplo, quais telas do sistema serão afetadas quando um determinado requisito for alterado, mantendo, assim, a consistência entre os requisitos e todas as entidades deles dependentes. Este conjunto de atividades também é conhecido como análise de impacto.

Requisito	Aspecto específico do sistema ou do seu ambiente					
	A01	A02	A03	A04	A05	Aii
R01			✓		✓	
R02	✓		✓			
R03	✓			✓		✓
R04		✓			✓	
R05	✓	✓		✓		✓
Rnn	✓		✓			

Figura 2 – Exemplo de Matriz de Rastreabilidade (Pressman, 2004).

Como visto nos parágrafos anteriores, o processo de ER é fundamentado na comunicação entre os envolvidos, sobretudo, durante as atividades de elicitação, negociação, especificação,

validação e gerência de requisitos, caracterizadas pela ocorrência de diversas conversas entre clientes/usuários e analistas. Em outras palavras, para que estas atividades sejam realizadas com sucesso, é essencial que as partes envolvidas se comuniquem adequadamente. Não é à toa que os maiores desafios enfrentados pelos engenheiros de requisitos dizem respeito à falta de comunicação (Macaulay, 1993); isto acontece porque o processo de desenvolvimento de software, sobretudo o de ER, é primeiramente um processo social que depende muito do tipo e do nível de comunicação existente entre seus participantes. A distância geográfica afeta de forma decisiva as tarefas de ER, sobretudo no que diz respeito à comunicação entre os envolvidos nestas atividades (Zowghi, 2002).

## 2.2 Desenvolvimento Distribuído de Software

O processo de Desenvolvimento Distribuído de Software, como o próprio nome sugere, envolve equipes de um projeto que se encontram separadas geograficamente e que, no entanto, cooperam no sentido de alcançar um mesmo objetivo: o desenvolvimento de um software. Tais equipes enfrentam diversas barreiras: organizacionais, culturais, técnicas e temporais, todas causadas pela distância geográfica que as separa (Carmel, 2000). Os parágrafos seguintes caracterizam cada tipo de obstáculo causado pela distância, segundo Carmel (2000).

Dentre as *barreiras organizacionais*, podem ser citadas: (a) a dificuldade na alocação de tarefas, isto é, “quem fará o quê quando e como?” e (b) a resistência da organização ao DDS: funcionários se sentem ameaçados com a possibilidade de transferência para outras cidades ou países e com a necessidade de viagens longas e frequentes.

A cooperação entre indivíduos de culturas diferentes representa o maior *desafio cultural*. Por exemplo, enquanto gerentes norte-americanos preferem tomar decisões com a participação de toda a equipe (supervisão baseada em consenso), os asiáticos – conhecidos por respeitar o conceito de hierarquia – preferem gerentes enérgicos, determinados e que não esperam pelo *feedback* de sua equipe (supervisão baseada no autoritarismo) (Carmel, 2000). Há situações em que a diversidade cultural também causa problemas de compreensão, pois as equipes envolvidas não estão situadas no mesmo país e, portanto, são obrigadas a se comunicar em um idioma que não o seu. Nestes casos, uma simples expressão, se interpretada equivocadamente, pode fazer toda diferença.

Um dos *obstáculos técnicos* de DDS é a dificuldade de se implementar a Gerência do Conhecimento em contextos distribuídos, o que acarreta a má disseminação das informações de um projeto às equipes distribuídas (De Souza et al, 2006). Diante de um problema, uma pessoa não é capaz de localizar o especialista, embora ele exista e esteja disponível na empresa. Além disso, as equipes continuam a cometer erros repetidos, pois desperdiçam oportunidades de reuso que, do contrário, poderiam economizar tempo e custo no projeto.

Os *desafios temporais* surgem quando as equipes estão situadas em lugares cujos fusos-horário são diferentes, pois o tempo de *overlap* diminui (Carmel, 2000). O tempo de *overlap* é a fração de tempo em que duas equipes distantes fisicamente são capazes de se comunicar de forma síncrona. Quanto menor for o tempo de *overlap*, maior será, por exemplo, o tempo de resposta a um e-mail, causando na equipe uma sensação de perda de controle sobre seu trabalho uma vez que dependerá e esperará por outras equipes para dar continuidade às suas tarefas.

Finalmente, a distância geográfica dificulta o estabelecimento do sentimento de confiança e de equipe (do inglês, *teamness*) entre as pessoas. Segundo Kotlarsky e Oshiri (2005), indivíduos que trabalham em locais diferentes não se vêem como uma mesma equipe; a interação face-a-face inspira confiança, respeito e comprometimento entre colegas de trabalho.

### **2.3 Engenharia de Requisitos e Desenvolvimento Distribuído de Software**

Damian e Zowghi (2002) definem os desafios enfrentados durante as atividades relacionadas à Engenharia de Requisitos e caracterizam os quatro maiores problemas decorrentes da distribuição geográfica dos *stakeholders*: 1) comunicação inadequada; 2) gerência ineficiente de conhecimento; 3) diversidade cultural; e 4) diferença de fuso-horário (ver Figura 3). Estes problemas, por sua vez, adicionam uma série de dificuldades específicas à área de ER, identificadas por meio de um estudo de campo realizado pelos autores: 1) diversidade de cultura e de negócio dos clientes; 2) pouca participação dos usuários do sistema com o corpo técnico (desenvolvedores, gerentes, analistas, etc); 3) falta de comunicação informal e pouca percepção do contexto local de trabalho; 4) redução do nível de confiança; 5) dificuldade na gerência de conflitos e na ocorrência de discussões entre os integrantes da equipe; 6) dificuldade na obtenção de uma interpretação comum dos requisitos

a todos; 7) ineficiência de reuniões para tomada de decisão e, finalmente, 8) demora na execução das tarefas.

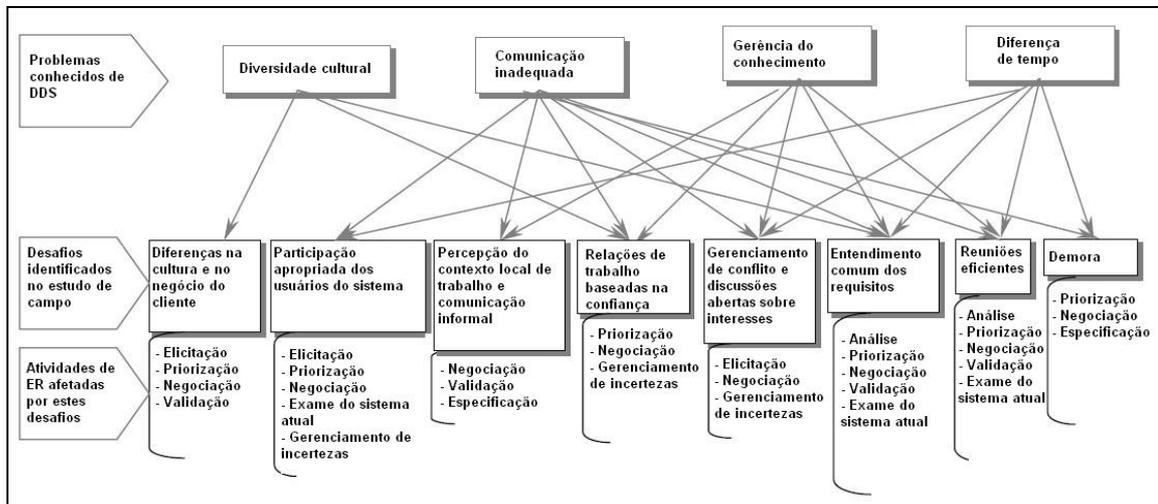


Figura 3 – Modelo de impacto dos desafios e das atividades de ER afetadas em virtude de problemas causados pela diversidade cultural, comunicação inadequada, gerência de conhecimento e diferença de tempo (Adaptado de Damian e Zowghi, 2002).

Cada um destes desafios é brevemente comentado nas subseções seguintes, segundo o trabalho de Damian e Zowghi (2002).

### 2.3.1 Diversidade de cultura e de negócio dos clientes

Quando um sistema é desenvolvido para clientes de culturas diferentes, o idioma pode se tornar um agravante, sobretudo, na fase de elicitación e validação de requisitos. Isto acontece porque a falta de domínio sobre o idioma poderá dificultar a comunicação e a compreensão dos requisitos, contribuindo para a elaboração de um documento de especificação incorreto, incompleto e até mesmo ambíguo. Além disso, as tendências de mercado mudam de um lugar para o outro, causando situações em que um sistema pode ser muito bem aceito em um determinado país, mas totalmente inadequado para outro.

### 2.3.2 Pouca participação dos usuários do sistema com a equipe técnica

A distância geográfica também provoca barreiras entre engenheiros de requisitos e usuários finais do sistema. Em ambientes distribuídos, os clientes não se envolvem diretamente com o processo de ER, ou seja, não existe contato face-a-face entre clientes e equipe de desenvolvimento. Muitas vezes, este processo é realizado por intermédio de meios síncronos (e até assíncronos) de comunicação. No caso da elicitación de requisitos, os mecanismos síncronos (e.g.: teleconferência) costumam dar resultados de maior qualidade

que os assíncronos (e.g.: questionários enviados por e-mail) (Lloyd et al, 2002), mas ainda assim ambos podem esconder as nuances de um contato pessoal, normalmente enriquecido por gestos, tons de voz e expressões faciais (Damian et al, 2001). Quando o meio de comunicação utilizado pelos *stakeholders* prejudica a qualidade da comunicação entre os mesmos, corre-se o risco de a especificação de requisitos elaborada ser de baixa qualidade.

### **2.3.3 Falta de comunicação informal e pouca percepção do contexto local de trabalho**

Entre equipes distribuídas não há comunicação informal (também conhecida por “conversas de corredor”) (Kraut e Streeter, 1995), o que contribui para que os membros de uma equipe tenham pouca noção do contexto de trabalho dos membros remotos. Isto significa que problemas relacionados a requisitos são propagados e resolvidos com mais rapidez em equipes que trabalham de forma co-localizada. Além disso, equipes distribuídas frequentemente, não conhecem a estrutura e a cultura organizacional umas das outras, dificultando a identificação correta dos papéis responsáveis com o processo de ER.

### **2.3.4 Redução do nível de confiança**

Para que as atividades de definição, negociação e validação de requisitos sejam realizadas com sucesso, é necessário que as pessoas envolvidas confiem umas nas outras. Entretanto, a falta de comunicação informal entre equipes distribuídas dificulta que as mesmas construam relações baseadas em confiança (Carmel, 2000). O sentimento de confiança pode ser adquirido, mas isto requer tempo: equipes distribuídas desenvolvem confiança na medida em que percebem que o trabalho está evoluindo de forma colaborativa, isto é, quando compreendem que todos estão contribuindo de forma positiva para o andamento do projeto.

### **2.3.5 Dificuldade na gerência de conflitos e na ocorrência de discussões entre os integrantes da equipe**

Lidar com conflitos de interesses já é problemático na ER tradicional (Nuseibeh e Easterbrook, 2000), mas se tornou ainda mais crítico com o surgimento do DDS. A distância diminui a qualidade da comunicação bem como a Gerência de Conhecimento existente em equipes distribuídas (De Souza et al, 2006), dificultando, portanto, a negociação de desvantagens entre os envolvidos. O resultado disto pode permitir que alguns requisitos não sejam informados publicamente, por exemplo, por meio de um fórum aberto, mas através de mensagens particulares de e-mails ou telefonemas.

### **2.3.6 Dificuldade na obtenção de uma interpretação comum dos requisitos**

Equipes distribuídas também enfrentam problemas em adquirir conhecimento comum sobre os requisitos de seus clientes, ou seja, os *stakeholders* não interpretam os requisitos da mesma forma. A razão para isto pode estar no fato de a especificação de requisitos muitas vezes agir como o único meio de comunicação entre clientes e desenvolvedores, forçando que estes últimos tentem adivinhar quais decisões de projeto (*rationale*) estão por trás de cada requisito.

### **2.3.7 Ineficiência de reuniões para tomada de decisão**

Realizar reuniões para tomada de decisão cujos participantes estejam distribuídos não é uma tarefa trivial. Primeiramente, estas reuniões acontecem por intermédio de ferramentas de tele ou videoconferência, portanto os participantes podem se sentir confortáveis para entrar depois de a reunião ter iniciado. Se a reunião não for mediada por um bom facilitador, a entrada destas pessoas pode passar despercebida. Isto dificultaria a identificação dos presentes (sobretudo se o meio de comunicação não possuir recursos visuais), por exemplo, da pessoa mais indicada para tratar de problemas relacionados a um determinado requisito. Dificuldades também são relatadas no que diz respeito à ausência de ferramentas síncronas de criação e compartilhamento de documentos: isto torna complicada a tarefa de manusear uma extensa especificação de requisitos e discutir suas modificações apenas por meio da numeração de páginas. Finalmente, podem acontecer situações em que questões já discutidas anteriormente sejam discutidas novamente em virtude da ausência de Gerência de Conhecimento.

### **2.3.8 Demora na execução das tarefas**

Finalmente, a diferença de fuso horário, embora possa funcionar como agente facilitador no processo de DDS (Mockus e Herbsleb, 2001), também prejudica o processo de ER em ambientes distribuídos uma vez que dificulta a utilização de mecanismos síncronos entre os participantes envolvidos no projeto. Isto acontece porque o tempo de *overlap* diminui, ou seja, dependendo do lugar onde as equipes estejam situadas, dificilmente haverá (ou não haverá, de fato) um intervalo de tempo satisfatório em que ambas estejam trabalhando em seus escritórios e possam se comunicar de forma síncrona. Um exemplo deste problema pode ser evidenciado quando a equipe de desenvolvimento se depara com um requisito que não está claro o suficiente. Se a diferença de tempo que separa estas equipes for muito grande, será necessário utilizar mecanismos assíncronos de comunicação (e.g.: e-mails). Dentro deste

contexto, o que era uma simples dúvida de requisitos pode levar dias para ser resolvida por e-mail ou até mesmo se tornar uma questão ainda mais complicada.

## 2.4 Trabalhos correlatos

A seção anterior apresentou os desafios enfrentados pelo processo distribuído de ER. Na tentativa de minimizar estes problemas, a comunidade acadêmica tem proposto diversas soluções para aperfeiçoar o processo de ER em ambientes distribuídos. Dentre as sugestões, podem-se citar metodologias, ferramentas e processos de Engenharia de Requisitos. Algumas destas propostas serão descritas em detalhes a seguir.

### 2.4.1 Metodologias

Várias metodologias já foram propostas no sentido de facilitar o processo de ER em ambientes distribuídos. Mullick e colegas (2006) realizaram um estudo experimental com o intuito de avaliar ferramentas e abordagens utilizadas em um processo de DDS. Como resultado, os autores produziram o que se pode considerar um guia de recomendações para a Engenharia de Requisitos distribuída. Por exemplo, o uso de um *wiki* contendo uma matriz de rastreabilidade a partir da qual existem *links* para descrições em alto nível, casos de uso, e quaisquer outros diagramas relevantes e casos de teste relacionados a um determinado requisito. Outro exemplo é o uso de um fórum de discussão como repositório de perguntas e respostas: todos os e-mails recebidos pela gerência contendo dúvidas de equipes distribuídas (por exemplo, sobre algum requisito) passam a ser respondidos por meio do fórum, disponibilizando a dúvida a todas as outras equipes e evitando que os gerentes percam tempo respondendo duas ou mais vezes a mesma pergunta. O estudo propõe também a elaboração de especificações de requisitos menos baseadas em diagramas UML (uma vez que as equipes remotas não se mostraram capazes de compreender com facilidade o que o sistema deveria fazer a partir do modelo de requisitos) e mais baseadas em descrições textuais.

A EasyWinWin (Gruenbacher, 2000) é uma metodologia que apóia as atividades de coleta, elaboração, priorização e negociação de requisitos a partir da integração de técnicas de grupo (como a realização de sessões de *brainstorming*) e ferramentas de colaboração. O uso da metodologia permite que uma equipe de *stakeholders* negocie os requisitos e chegue a um consenso mais detalhado e em menos tempo quando comparado com a utilização de técnicas tradicionais, como entrevistas. Estudos de caso conduzidos com esta metodologia sugerem

que o uso da mesma aprimora o envolvimento dos *stakeholders* e a interação entre eles durante o processo de ER, pois prioriza condições de ganho-mútuo<sup>2</sup> (Berenbach e Gall, 2006).

A metodologia proposta por Aranda e colegas (2006) é fundamentada na psicologia cognitiva e consiste em definir quais ferramentas de colaboração e técnicas de elicitación de requisitos melhor se ajustam a um determinado tipo de pessoa, segundo o modelo de classificação Felder – Silverman (F-S). O modelo F-S, ilustrado na Figura 4, apresenta quatro categorias (percepção, absorção, processamento e compreensão), estando cada uma delas dividida em duas subcategorias (sensível/intuitivo, visual/verbal, ativo/reflexivo, seqüencial/global).

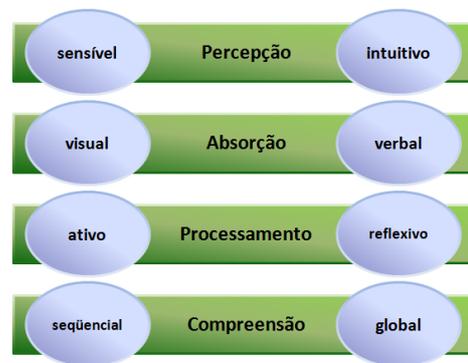


Figura 4 – Categorias e subcategorias do modelo Felder – Silverman (Adaptado de Aranda et al, 2006).

Segundo Aranda e colegas, durante o processo de ER, se apenas uma ferramenta ou técnica de elicitación é utilizada por uma organização, eventualmente pessoas que não se sintam confortáveis em utilizá-las, podem se mostrar desmotivadas a realizarem suas tarefas, o que resultaria na elaboração de um documento de especificação de qualidade duvidosa. A maior contribuição desta metodologia para a ER é sua capacidade de sugerir que tipo de ferramentas e técnicas de elicitación de requisitos podem ser mais adequadas a uma pessoa segundo suas características pessoais. A metodologia também considera fatores inerentes ao DDS, como a distância temporal, a diferença de idiomas, entre outros.

A Abordagem Unificada (do inglês, *Unified Approach*), proposta por Berenbach (2006), consiste em uma extensão da UML para suportar a integração entre características de um

<sup>2</sup> O objetivo da negociação do tipo ganha-ganha (do inglês, *win-win*) é encontrar uma solução que seja aceita por ambas as partes (por exemplo, clientes e desenvolvedores) e que, além disso, seja capaz de deixá-las com a sensação de satisfação – neste caso, satisfação mútua.

sistema, casos de uso, requisitos funcionais e não-funcionais, e riscos em um único modelo UML. Isto é possível a partir da adição de estereótipos que representam novos tipos de relacionamentos, por exemplo: um novo estereótipo pode se chamar “*mitigate*”, assim um possível relacionamento pode existir de forma que um requisito funcional mitigue um determinado risco. O requisito “cancela de pedágio pára quando qualquer resistência é detectada” seria responsável por mitigar o risco “cancela de pedágio pode descer e machucar uma pessoa”. Para o processo distribuído de ER, esta abordagem é interessante porque se trata de um mecanismo gráfico que reduz problemas culturais (idiomas diferentes) e de comunicação, presentes entre equipes separadas geograficamente e facilita, portanto, a compreensão correta dos requisitos.

Entretanto, as conclusões deste trabalho entram em conflito com os resultados apresentados por Mullick e colegas (2006), o que possivelmente sugere que a literatura sobre este assunto ainda não está suficientemente madura.

#### **2.4.2 Ferramentas**

O *Econference* (Calefato e Lanubile, 2005) é uma aplicação de conferência que possibilita a comunicação síncrona e estruturada entre equipes distribuídas durante a realização de atividades de ER. Dentre as funcionalidades da ferramenta estão: *chat* integrado à agenda, quadro de comunicação (*whiteboard*) e suporte ao conceito de percepção (*awareness*). O *Econference* permite que os *stakeholders*: a) se comuniquem uns com os outros de forma síncrona por meio do chat durante, por exemplo, uma elicitação de requisitos; b) organizem decisões tomadas em uma reunião através do *whiteboard* e c) exponham a colegas remotos qual papel exercem em seu contexto local de trabalho, bem como seu status atual (conectado ou desconectado) (*awareness*).

O CRETA (Cooperative Requirements Engineering Support Tool) (Togneri et al, 2002) é uma ferramenta que integra ferramentas de colaboração com o intuito de apoiar as principais atividades do processo de ER uma vez que promove mecanismos para o compartilhamento de informação, facilita comunicação, coordenação e cooperação entre as pessoas, bem como o conceito de *awareness* e a Gerência de Conhecimento. O sistema apóia a cooperação na medida em que permite a coordenação das atividades e o compartilhamento de conhecimento. A ferramenta apóia também o conceito de *awareness*, pois torna as pessoas cientes das

atividades umas das outras. Finalmente, esta ferramenta também possibilita o registro, o compartilhamento e o reuso da memória organizacional da empresa.

A ferramenta de código aberto CODIPSE-Req (Brito e Vasconcelos, 2006) foi construída e incorporada ao e-Groupware<sup>3</sup> e, juntos, permitem o registro e a manutenção do *rationale* sobre os requisitos, aumentam a compreensão destes em ambientes de DDS, e aprimoram a colaboração e a coordenação entre os *stakeholders* durante as atividades da ER. Dentre as funcionalidades oferecidas pela ferramenta, estão: a) o agendamento de reuniões através da agenda de eventos; b) o registro das decisões tomadas durante as reuniões; c) a manutenção das alterações de requisitos através de links com quaisquer interações que tenham causado esta alteração; d) o cadastro de informações sobre os casos de uso, como descrição, atores, pré e pós-condições e fluxos principal e alternativo; e) a exportação destas informações para o Microsoft Word 2003, entre outras.

Assim como o *Econference*, o CRETA e o CODIPSE-Req foram ferramentas construídas diretamente com o propósito de apoiar a ER distribuída, aplicações de propósito geral também podem ser adaptadas para minimizar os desafios impostos pela distância, como é o caso de *TeamWave*, Microsoft NetMeeting e Centra Symposium.

O *TeamWave* trata-se de uma ferramenta de colaboração, que “simula” uma sala de reuniões equipada com uma série de ferramentas de colaboração, tais como: quadro de avisos, visualizador de arquivos, chat, lista de participantes, entre outros. Herlea e Greenberg (1998) mostram como as funcionalidades da *TeamWave* podem ser utilizadas para apoiar ao processo de ER distribuída. Algumas delas são: a) a criação de um espaço de reuniões; b) a facilidade de negociar requisitos, uma vez que as pessoas envolvidas com a fase de elaboração do escopo podem trocar informações acerca dos requisitos do sistema, mostrando umas às outras suas perspectivas em um ambiente de trabalho compartilhado, e; c) a manutenção dos requisitos por meio de suas ferramentas de versionamento.

Outras ferramentas de propósito geral também podem auxiliar tarefas distribuídas de ER como o *Microsoft NetMeeting* (Damian et al, 2000) e o *Centra Symposium* (Lloyd et al, 2002). Lloyd (2001) apresenta um estudo empírico no qual são analisadas diversas ferramentas de colaboração bem como o suporte que estes oferecem ao processo de ER em

---

<sup>3</sup> O eGroupWare é um software colaborativo de código aberto, voltado para negócios, permitindo gerenciamento de contatos, projetos e listas “to-do” (eGroupware, 2000).

ambientes distribuídos. Dentre as ferramentas estudadas estão *Microsoft NetMeeting*, *Centra Symposium 99*, *Centra eMeeting*, entre outras. O *Centra Symposium* possui funcionalidades de grande utilidade para os engenheiros de requisitos: agenda/slide, teleconferência e *chat*. Os resultados da pesquisa revelam que o *chat* da ferramenta *Centra Symposium* se mostrou útil para clientes que não costumam participar ativamente de reuniões virtuais, enquanto que o *whiteboard*, a apresentação de slides e da agenda, e o compartilhamento do navegador são funcionalidades eficazes para clientes que participam de forma efetiva de reuniões virtuais. O estudo reúne informações de diversas ferramentas, suas funcionalidades e como elas podem minimizar o efeito da distância no processo de ER. Os dados apresentados no estudo foram coletados a partir de um *survey* aplicado a seis equipes de engenheiros de requisitos que deveriam seguir um processo básico para elicitar, definir, refinar e especificar requisitos para um sistema de agendamento para uma empresa.

Embora existam estudos que sugiram a supremacia das ferramentas síncronas em relação às assíncronas (Lloyd et al, 2002), alguns trabalhos argumentam que tanto canais de comunicação síncronos quanto assíncronos são necessários para aprimorar as atividades relacionadas ao processo distribuído de ER (Damian et al, 2006).

## 2.4.3 Processos de Engenharia de Requisitos

### 2.4.3.1 Lopes (2004)

Lopes e colegas (Lopes et al, 2005) propõem uma adaptação ao processo de ER tradicional visando minimizar os principais desafios oriundos de ambientes de desenvolvimento de software distribuídos. A adaptação proposta considera que existem engenheiros de requisitos (ER) próximos tanto dos clientes (C) e dos usuários (U), quanto da equipe de desenvolvimento (D), estando clientes/usuários e desenvolvedores distantes um do outro, tal como mostra a Figura 5.

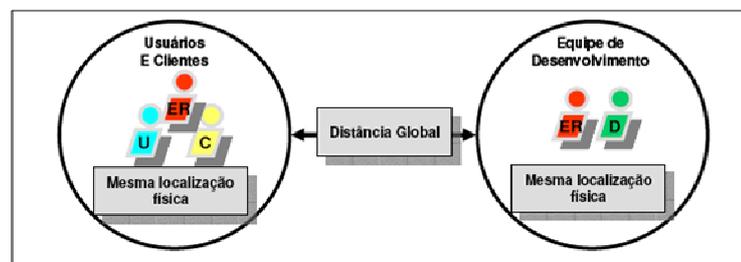


Figura 5 – Distribuição dos engenheiros de requisitos no exemplo de cenário (Lopes, 2004).

Inicialmente, os autores propuseram um processo de requisitos preliminar composto por cinco passos (Figura 6): 1) o documento de especificação de requisitos (SRS) é concluído e enviado à equipe de desenvolvedores; 2) a SRS é analisada e adaptada pela equipe de desenvolvimento; 3) a análise é concluída e a SRS é enviada de volta à equipe de especificação para ser validada; 4) a SRS é validada e aprovada pelos engenheiros de requisitos, e, finalmente; 5) uma versão final da SRS é elaborada.

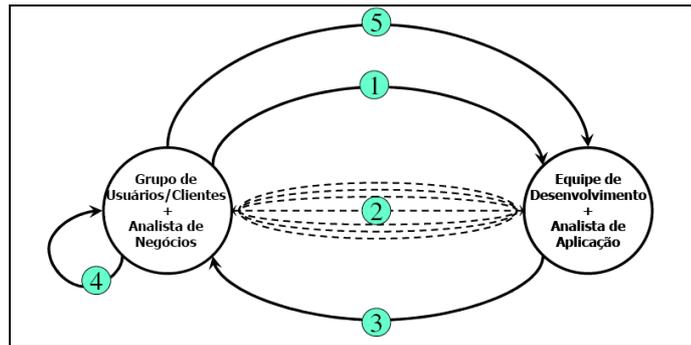


Figura 6 – Processo preliminar de Engenharia de Requisitos para ambientes de DDS (Lopes, 2004).

O objetivo desta proposta era minimizar, sobretudo, problemas relacionados à comunicação entre os *stakeholders* (causados principalmente por questões de idioma), assim como a ocorrência de ambigüidade e falta de clareza na descrição dos requisitos. Dois estudos de caso foram realizados durante a execução deste trabalho: o primeiro serviu para avaliar o processo proposto preliminarmente, enquanto que o segundo foi realizado com o intuito de se identificar os problemas que a distância tende a acentuar durante as atividades de requisitos.

Após a análise dos dados obtidos com os estudos de caso o processo proposto sofreu modificações. Desta forma, o processo final (Figura 7) passou a contar com as seguintes etapas: 1) estabelecimento da infra-estrutura a ser utilizada durante as atividades de requisitos, isto é, definição da linguagem de especificação, dos meios de comunicação, da equipe e de suas responsabilidades, etc; 2) mapeamento do contexto no qual o software será inserido (base de conceitos do universo de informações, base de informações culturais, etc); 3) criação da especificação de requisitos; 4) gerência de requisitos, e atividades de suporte para manutenção das bases citadas anteriormente.

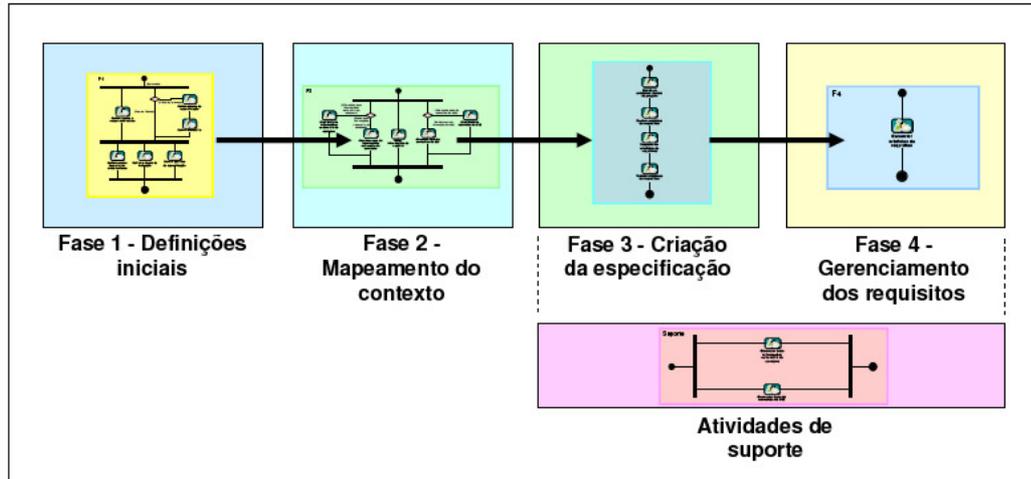


Figura 7 – Modelo de processo de Engenharia de Requisitos em DDS (Lopes, 2004).

#### 2.4.3.2 Brito (2006)

O processo proposto por Brito e Vasconcelos (2006) assume que os participantes do projeto estão totalmente distribuídos e baseia-se no RUP e no processo proposto por Lopes (2004). A proposta reúne alguns conceitos de ferramentas de colaboração, como coordenação e memória organizacional através da construção de bases de conhecimento que podem ser consultadas pelas pessoas para aprimorar o entendimento dos requisitos e apoiar a comunicação e a coordenação entre elas.

Além de atividades, os autores definem os papéis, responsabilidades e artefatos que existem no modelo de processo proposto. Dentre as atividades previstas no processo (Figura 8) estão: 1) Planejar as atividades (definição do idioma em que a especificação será elaborada e padrões para as atividades de gerência de requisitos); 2) Organizar conhecimento (definições iniciais do sistema, como o documento de escopo ou de visão e distribuição destas informações aos *stakeholders*); 3) Elicitar requisitos (definição dos requisitos do sistema através de interações à distância com as partes interessadas); 4) Analisar e negociar requisitos (refinamento dos requisitos com o intuito de retirar problemas de inconsistência e ambigüidade); 5) Documentar requisitos (transformação dos requisitos em versões finais, prontas para serem validadas pelo cliente); 6) Validar requisitos (clientes verificam se os requisitos estão de acordo com seus padrões); 7) Gerenciar requisitos (manutenção das dependências entre os requisitos e as informações que originaram quaisquer mudanças nestes, importantes para a manutenção do *rationale*), e; 8) Gerenciar informações e conhecimento (criação e atualização das bases de conhecimento formal e informal e de informações de negócio).

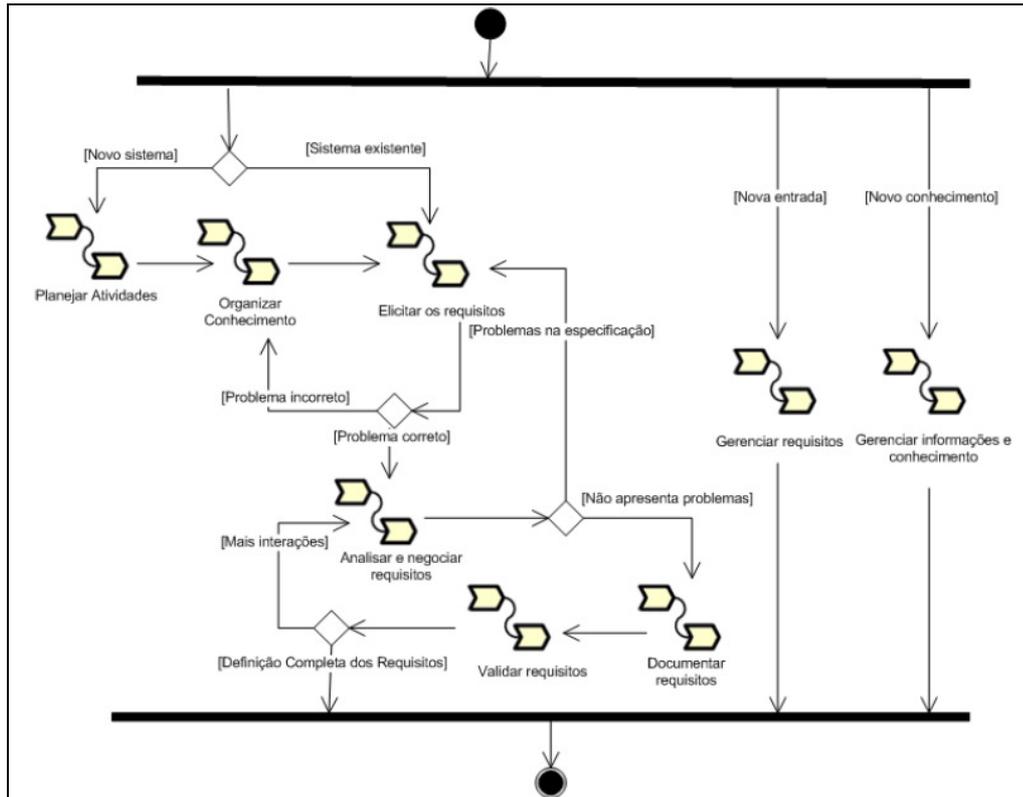


Figura 8 – Fluxo principal do processo colaborativo (Brito, 2006).

#### 2.4.3.3 Medeiros et. al (2007)

Medeiros e colegas (2007) também propuseram um modelo de processo de Engenharia de Requisitos para ambientes de DDS, chamado DSD-REQ-Process (Medeiros, 2007). O modelo é resultado da análise de dados coletados a partir de entrevistas semi-estruturadas com gerentes de projetos e de observações participativas, realizadas em um estudo de caso. Tanto as entrevistas semi-estruturadas quanto as observações participativas auxiliaram na compreensão do trabalho de equipes envolvidas em DDS e, assim, permitiram que um processo de engenharia de requisitos para ambientes de desenvolvimento distribuído de software.

O DSD-REQ-Process é baseado no RUP, o que o torna genérico o suficiente para se adequar a diversos domínios e tipos de aplicações. Além disso, o modelo visa: a) diminuir a distância entre as equipes envolvidas com tarefas de ER; b) permitir a revisão assíncrona de artefatos, e; c) especificar requisitos com a ajuda de *storyboards*.

O modelo é dividido em duas fases: presencial e distribuída, nas quais se distribuem as etapas previstas. São elas: a) Elicitação de requisitos que é obrigatoriamente presencial para tirar vantagem da comunicação face-a-face entre os *stakeholders* (Figura 9); b) Documentação

de requisitos que ocorre à distância e é apoiada pela apresentação de *storyboards* (Figura 10), e; c) Validação de requisitos, também ocorrida à distância e caracterizada pela revisão e inspeção dos documentos produzidos durante o processo (Figura 11).

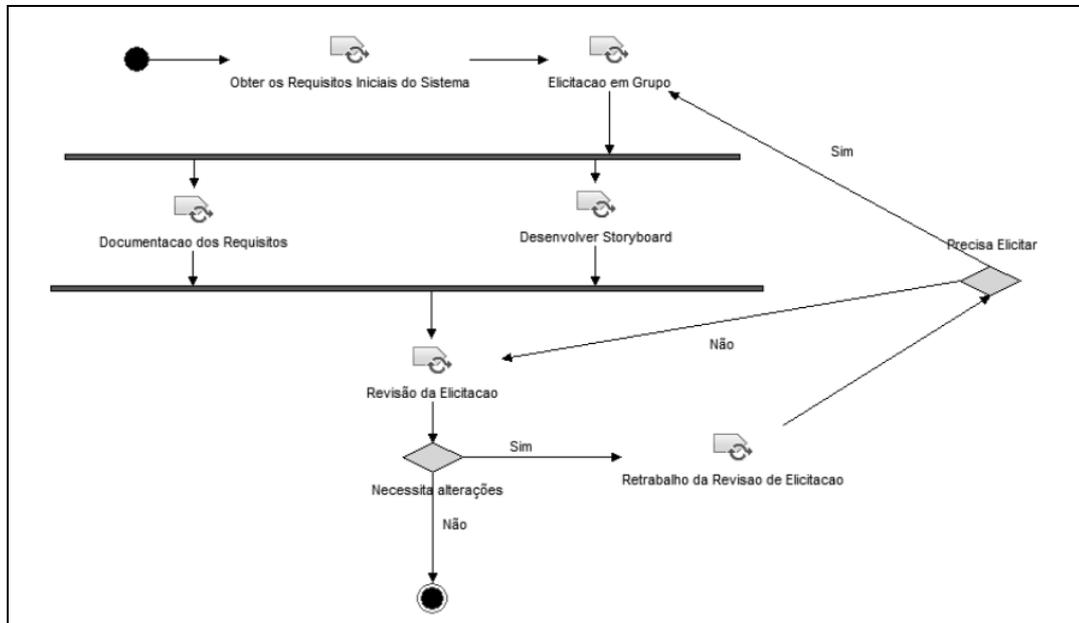


Figura 9 – Iteração de Elicitação de Requisitos (Medeiros, 2007).

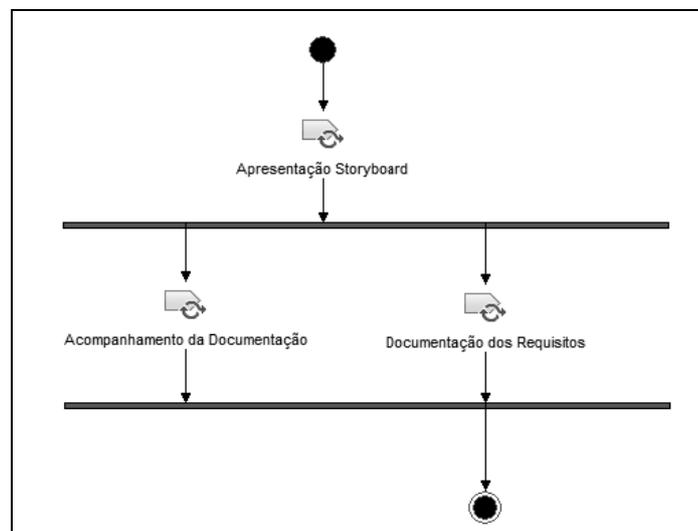


Figura 10 – Iteração de Documentação dos requisitos (Medeiros, 2007).



## 3 MÉTODO DE PESQUISA

A metodologia utilizada neste trabalho é composta por uma revisão bibliográfica e por um estudo etnográfico. Uma pesquisa bibliográfica foi realizada acerca dos temas envolvidos, isto é, livros e artigos em ER, DDS e ER em ambientes de DDS e utilizada como fonte de pesquisa. A partir deste material, foram extraídos trabalhos relacionados com os quais o estudo em questão pudesse ser comparado, além da base teórica sobre a qual este trabalho está fundamentado (consultar Capítulo 2). Além disso, um estudo etnográfico foi conduzido em uma empresa utilizando observações não-participativas e entrevistas não e semi-estruturadas para coletar dados sobre as atividades de desenvolvimento de software dos analistas, em especial as relacionadas à ER. Posteriormente, estes dados foram analisados utilizando métodos da Teoria Fundamentada em Dados (Glaser e Strauss, 1969). Cada uma destas abordagens será explicada nas próximas seções.

### 3.1 Etnografia

As próximas seções mostram as motivações pelas quais a etnografia foi utilizada como método de pesquisa, assim como a definição e um breve histórico do método etnográfico, o papel da etnografia na Engenharia de Software e os tipos de etnografias utilizados em desenvolvimento de software.

#### 3.1.1 Motivações

Muitos dos trabalhos que visam reduzir os problemas acarretados pela distância geográfica durante a fase de Engenharia de Requisitos giram em torno da realização de estudos empíricos. Os métodos empíricos utilizados nestes estudos são os mais variados. Espinosa e colegas (2007), por exemplo, realizaram um *experimento* em que 42 duplas foram

distribuídas de acordo com 4 condições de tempo de *overlap*<sup>4</sup>: a) tempo total de *overlap*; b) intervalo de tempo de *overlap* significativo; c) algum intervalo de tempo de *overlap*, e; d) nenhum intervalo de tempo de *overlap*. Os resultados deste estudo mostram que um pequeno tempo de *overlap* entre as equipes não exerce efeito sobre a precisão com que realizam uma tarefa, entretanto influencia outro fator: a velocidade com que a tarefa é executada. O estudo também revela resultados inesperados: um aumento no tempo de *overlap* não causa implicações significativas na velocidade, já a ausência de tempo de *overlap* acarreta um aumento da velocidade, ainda que não de forma significativa.

Lloyd e colegas (2002) também realizaram um *experimento* em que aplicavam questionários e realizavam observações com estudantes para avaliar como aplicações de *groupware* poderiam apoiar o processo distribuído de ER.

Um exemplo de *estudo de caso* pode ser encontrado em (Prikladnicki e Audy, 2002). Este trabalho consiste em um estudo exploratório cujo principal objetivo é analisar o processo de ER em ambientes distribuídos de software. Os dados foram coletados mediante a realização de observações e entrevistas e os resultados apresentam os problemas identificados na empresa estudada, bem como quais soluções foram propostas para adaptar a ER a ambientes de DDS.

Outro exemplo de *estudo de caso* pode ser encontrado em (Damian e Zowghi, 2002). Este trabalho investigou os desafios da ER causados pela distribuição geográfica dos *stakeholders* em uma organização distribuída em várias filiais. Os dados para este estudo de caso foram coletados a partir da inspeção de documentos, observações de reuniões e entrevistas semi-estruturadas com integrantes de vários setores de *apenas* uma das sedes da empresa estudada. Os resultados encontrados identificaram uma série de desafios durante a etapa de negociação de requisitos e sugeriram algumas recomendações para melhorar a prática de ER em ambientes distribuídos.

Illes-Seifert e colegas aplicaram e avaliaram um total de 744 *questionários* no sentido de enfatizar quão essencial é a comunicação em projetos de Desenvolvimento Distribuído de Software (Illes-Seifert et al, 2007). De maneira similar, Damian e colegas (2000) realizaram um estudo empírico em que aplicaram um *questionário* de perguntas abertas além de

---

<sup>4</sup> Tempo de *overlap* representa o intervalo de tempo em que equipes distribuídas estão trabalhando em seus escritórios e podem se comunicar de forma síncrona.

observação via vídeo e editores colaborativos para coleta dos dados com 45 estudantes que realizaram atividades de elicitação, negociação e documentação de requisitos.

De um modo geral, pode-se observar que tanto métodos quantitativos (experimentos e questionários) quanto métodos qualitativos (estudos de caso) têm sido utilizados para estudar projetos distribuídos. Entretanto, estes métodos não consideram de maneira apropriada o ponto de vista dos engenheiros de software envolvidos em projetos de DDS. Isto acontece em virtude de dois grandes motivos.

O primeiro motivo diz respeito à própria natureza dos métodos empíricos escolhidos: durante a condução de *experimentos*, as variáveis do ambiente são manipuladas visando obter uma relação de causa-efeito (Wild e Seber, 1999). O uso de *questionários*, por outro lado, não assegura que as respostas fornecidas pelos informantes correspondem de fato à realidade da organização e, além disso, os questionários geralmente são compostos por perguntas pré-definidas que limitam o grau de controle que o respondente tem sobre a conversa, conforme será discutido na seção 3.2.1. Diante disto, pode-se afirmar que em ambos os casos (experimentos e questionários), detalhes importantes do ambiente real de trabalho podem ser ocultados. Isto não é interessante para o objetivo do estudo proposto nesta dissertação.

Já no *estudo de caso*, o pesquisador consegue ter acesso a estes detalhes uma vez que utiliza os mesmos métodos de coleta de dados da etnografia (entrevistas e observações), mas ainda assim não lhe é exigida a presença em campo, isto é, o pesquisador pode realizar ótimos estudos de caso sem, no entanto, precisar estar inserido na cultura de todos os informantes (Yin, 2003). Isto pode ocasionar um problema grave, que indica o segundo motivo pelo qual os trabalhos propostos até então não consideram de maneira adequada o ponto de vista de todos os *stakeholders*: os estudos de caso existentes normalmente levam em conta apenas uma das equipes do projeto, já que as demais se encontram distribuídas, o que dificulta bastante a realização de entrevistas presenciais e observações de todas as equipes envolvidas. Em resumo, acaba-se perdendo o contexto de trabalho de um ou de vários lados.

O método etnográfico utilizado como método de pesquisa neste estudo está explicado a seguir e visa resolver estas limitações.

### 3.1.2 Definição

Etnografia (do grego, *ethno* – nação e *graphein* – escrever) é um método de pesquisa qualitativa que enfatiza a observação de pessoas em seus ambientes naturais (Randall et al,

2007). Praticantes da abordagem etnográfica, conhecidos como etnógrafos, passam uma quantidade significativa de tempo no campo de pesquisa, imersos na vida de indivíduos – alvos de sua observação.

Os estudos etnográficos também fornecem o contexto completo de uma comunidade e descrevem de maneira bastante rica os elementos que a compõem (pessoas, ambientes e interações) bem como as atividades dos membros da comunidade. Em um estudo etnográfico, as atividades são compreendidas a partir da perspectiva do informante (Blomberg, 1993). *A principal vantagem da etnografia é permitir ao pesquisador entender os aspectos que são importantes para os informantes, no caso deste trabalho, os engenheiros de software envolvidos em projetos distribuídos, ao invés de focar em aspectos relevantes para o próprio pesquisador.*

A etnografia, diferentemente de estudos quantitativos, é realizada mediante a inserção – que neste caso é exigida – de etnógrafos no ambiente de trabalho dos indivíduos, sem alterar variáveis do ambiente sob estudo (McGrath, 1994). Os etnógrafos observam práticas reais de trabalho, ao invés das chamadas “práticas prescritas”, ou seja, as práticas definidas por manuais e guias da empresa. Em outras palavras, a abordagem etnográfica proporciona um profundo entendimento acerca das *reais* atividades desempenhadas – ao invés das atividades *formais*, normalmente mencionadas em um processo tradicional de condução de entrevistas.

De um modo geral, a visão da autora deste trabalho é a de que somente através do entendimento do ponto de vista do informante é possível desenvolver ferramentas e/ou propor recomendações que considerem os reais problemas enfrentados tanto por engenheiros de software quanto por clientes e usuários envolvidos em projetos distribuídos.

### **3.1.3 Um breve histórico**

A Etnografia foi concebida por Bronislaw Malinowski (1967 apud Anderson, 1997). Seu primeiro uso foi no contexto da Primeira Guerra Mundial, quando Malinowski realizou um estudo com famílias aborígenes na costa da Nova Guiné: dava-se início à forma moderna de realizar estudos de campo<sup>5</sup> como uma ciência social investigativa, sendo a etnografia seu complemento analítico. Para Malinowski, o objetivo da etnografia era se familiarizar com um

---

<sup>5</sup> Estudo de campo: é um tipo de estratégia de pesquisa em que o pesquisador se propõe a fazer observações diretas, com o mínimo de intervenção possível, em um determinado contexto. A etnografia é um exemplo desta estratégia (McGrath, 1994).

modo de vida a partir do aprendizado da linguagem, da cultura de uma determinada comunidade e vivendo conforme o dia-a-dia do informante. Uma vez alcançado este nível de intimidade, o etnógrafo também alcançaria o entendimento necessário para integrar e correlacionar os vários tipos de dados coletados.

Entre 1930 e 1960, sociólogos de Chicago iniciaram um amplo programa de investigações da vida urbana na América contemporânea (Anderson, 1997). Durante a realização desta investigação, a etnografia foi adotada como método de pesquisa em várias sub-culturas caracterizadas como pertencentes à “sociedade marginalizada” (ex.: mendigos, alcoólatras, usuários de drogas).

A aplicação de etnografias, no entanto, se estendeu para além de sociedades marginalizadas, alcançando também ambientes de trabalho, tais como estudantes de medicina, enfermeiras, pilotos de avião, zeladores, professores, etc (Dourish, 2004). Estas pesquisas representaram um papel importante na transição da etnografia que, de método de pesquisa na Sociologia, passou a ser utilizada também durante o desenvolvimento de software. A etnografia tem sido adotada por profissionais da área de Interação Homem-Computador e Trabalho Cooperativo há décadas (Suchman, 1987) e, mais recentemente, por profissionais de Engenharia de Software (Sommerville, 2004).

### **3.1.4 Etnografia em Engenharia de Software**

A necessidade de inclusão da perspectiva social e, portanto, da etnografia à Engenharia de Software aconteceu a partir da constatação de que muitos erros surgiam em virtude de o projeto do sistema não considerar de forma adequada o contexto social das atividades que apoiava. Esta falha é freqüentemente atribuída à incapacidade e à ineficiência dos métodos existentes para elicitación de requisitos e análise de práticas de trabalho (Schmidt e Carstensen, 1993).

Blomberg e colegas (1993) citam inúmeras razões pelas quais a realização de um estudo etnográfico é relevante ao projeto de software:

- Uma vez que engenheiros de software estão freqüentemente criando sistemas para ambientes de trabalho desconhecidos, torna-se necessário que adquiram informação sobre estes ambientes para que as tecnologias desenvolvidas se ajustem aos mesmos. A abordagem etnográfica permite a aquisição deste conhecimento;

- Já que os sistemas devem apoiar as práticas de trabalho de seus usuários, é importante que a opinião dos engenheiros de software não seja imposta de forma inapropriada a estes usuários. Quando um analista de requisitos não possui conhecimento acerca de uma determinada situação, a etnografia surge como uma forte aliada para que ele adquira conhecimento, sem correr o risco de propor tecnologias que se encaixem às suas próprias necessidades ao invés de às necessidades de seus reais usuários;
- Existem situações em que o engenheiro cria tecnologias cuja utilização é desconhecida. Se este engenheiro, no entanto, possuir algum entendimento sobre o trabalho realizado pelos usuários em potencial desta tecnologia (novamente, entendimento este que pode ser obtido por meio de etnografias), torna-se mais fácil identificar alguns de seus possíveis usos e até mesmo refinar o projeto original;
- Uma vez que a experiência que um usuário possui em uma determinada tecnologia depende do contexto no qual a mesma é utilizada, torna-se vantajoso o uso de estudos etnográficos para que se alcance uma visão mais ampla sobre o uso desta tecnologia do que aquela alcançada através de testes tradicionais de operacionalidade<sup>6</sup>. Blomberg (1998) aponta que estes testes não consideram fatores sociais que estão disponíveis na maioria dos ambientes de trabalho;
- Quando sistemas inovadores são projetados, os usuários são freqüentemente incapazes de responder perguntas sobre como estes sistemas deveriam funcionar. Por este motivo, os usuários precisam visualizar e experimentar a nova tecnologia dentro do contexto de suas próprias práticas de trabalho. Etnografias auxiliam os engenheiros a conhecer adequadamente o trabalho de seus usuários para criar o contexto propício a estas discussões. Além disso, usuários e analistas devem se tornar parceiros enquanto descobrem a relação existente entre as práticas de trabalho e o sistema desenvolvido; e
- Finalmente, alguns projetos focam em uma única tarefa e, por isso, não se encaixam em ambientes onde as tarefas se integram. Focar em uma única tarefa, ou nas tarefas de um único usuário, desconsidera a forma como as atividades realizadas por um indivíduo se relacionam

---

<sup>6</sup> Em testes de operacionalidade: solicita-se que usuários em potencial de um sistema realizem tarefas em um protótipo. Este tipo de teste é realizado em laboratórios e avalia o uso do sistema em termos de tempo gasto durante a realização de uma tarefa, frequência com que uma tarefa foi realizada com êxito, etc (Blomberg, 1988).

com as realizadas por outros indivíduos de uma empresa. As etnografias não observam um indivíduo de forma isolada, mas um conjunto de indivíduos e as relações existentes entre eles.

Embora Blomberg e colegas (1993) tenham enfatizado a importância das etnografias considerando apenas a fase de concepção de novos sistemas, este tipo de abordagem também pode ser utilizado para avaliar sistemas já existentes. Segundo Nardi (1997), entender o usuário é importante em diversos momentos durante o ciclo de desenvolvimento de software, sobretudo em duas fases: antes de o projeto tomar forma, como proposto por Blomberg e colegas (1993), e ao se disponibilizar um protótipo robusto para ser testado. Em se tratando de avaliação, o etnógrafo toma o cuidado de observar atentamente aquilo que acontece após a instalação do protótipo, isto é, como usuários reais do sistema se comportam enquanto o manipulam.

### **3.1.5 Tipos de etnografia na Engenharia de Software**

Esta seção pretende mostrar o *framework* proposto por Hughes e colegas (1994), que classifica a etnografia de acordo com as diversas formas como ela pode ser utilizada no processo de desenvolvimento de software. A seção apresenta ainda o trabalho de Millen (2000) que complementa o anterior com a proposta da Etnografia Rápida.

#### *3.1.5.1 Etnografia concorrente*

A etnografia concorrente não só é o tipo mais associado a projeto de sistemas como também é o mais pesquisado. Inúmeros estudos já foram realizados utilizando este método (Heath e Luff, 1991) (Hughes et al, 1994) (Sommerville et al, 1993).

Neste caso, a investigação etnográfica acontece de forma paralela ao desenvolvimento do sistema. A vantagem deste método é que ele assegura que existirá foco no usuário durante todas as etapas do desenvolvimento do sistema. Sua principal desvantagem reside no fato de que, de todos os métodos etnográficos, trata-se daquele que mais consome tempo e senso de cooperação e coordenação da equipe.

#### *3.1.5.2 Etnografia quick and dirty*

Tradicionalmente, a etnografia é considerada uma atividade demorada, podendo levar até mesmo anos de duração. A abordagem *quick & dirty* surge como uma solução a este problema, uma vez que visa estudos curtos e bastante focados para obtenção de uma visão geral do ambiente de trabalho. Além disso, este tipo de etnografia pode ser utilizado como um método precursor para outros métodos de pesquisa etnográficos.

Sua principal vantagem está em proporcionar a diminuição do tempo gasto com observações. Isto acontece porque o etnógrafo seleciona quais amostras do ambiente são realmente relevantes ao projeto do sistema. Ainda como benefício, este tipo de etnografia pode promover o aumento da percepção de problemas de usabilidade e aceitabilidade de um sistema por parte de seus usuários. Como desvantagem, pode-se afirmar que os resultados ficam limitados a apenas uma visão geral da cultura do trabalho, desconsiderando detalhes potencialmente importantes, mas que se encontram fora do ambiente observado.

#### 3.1.5.3 *Etnografia avaliadora*

Esta é uma abordagem etnográfica semelhante à anterior no sentido de que não exige períodos longos de observações porque se atém a um foco bastante limitado: avaliar um projeto já existente. Esta abordagem etnográfica pode ser realizada com o intuito também de prever em que escala um novo projeto afetará as práticas de trabalho de uma empresa.

Sua principal vantagem é a capacidade de aprovar ou desaprovar um novo modelo de projeto, além do curto período de tempo exigido. Já a principal desvantagem aproxima-se daquela mostrada para a *quick & dirty*: estreitar demais um foco pode “cegar” o etnógrafo para informações relevantes que estão fora do domínio sob estudo.

#### 3.1.5.4 *Re-análise de estudos anteriores*

Esta abordagem consiste em “aproveitar” os estudos etnográficos já realizados em uma determinada área e aplicá-los na fase de projeto de sistemas. Isto é possível porque a etnografia, embora seja relativamente nova na engenharia de software, é um método utilizado por muitas décadas na Sociologia e na Antropologia. Estudos nestas áreas podem ser utilizados, senão como a base para projetos de sistemas, pelo menos como fonte informativa.

Como vantagem principal, pode-se afirmar que este método é o que consome menor quantidade de tempo e recursos financeiros. Sua maior desvantagem está na dificuldade de se encontrar um estudo já realizado e que seja completamente apropriado ao ambiente em questão.

#### 3.1.5.5 *Etnografia rápida*

Comparado aos demais tipos, trata-se do mais recente. A etnografia rápida foi idealizada com o objetivo de prover entendimento razoável sobre um ambiente, usuários e atividades ali desempenhadas, considerando equipes sob pressão de tempo e com pouca disponibilidade para atuação no campo. A principal diferença desta abordagem para a *quick & dirty* é que a

etnografia rápida foi formalizada em princípios que devem ser seguidos como a restrição de foco, a utilização de informantes-chave e de múltiplos observadores, e o uso de ferramentas para automatizar a análise dos dados coletados.

Sua principal vantagem, como já é de se esperar, coincide com a da abordagem *quick & dirty*. Já sua principal desvantagem está na dificuldade em seguir os princípios sobre os quais o método se fundamenta.

## 3.2 Entrevistas e Observações

Esta seção descreve os métodos referentes à forma como os dados foram coletados.

### 3.2.1 Entrevistas

De acordo com Dewalt e Dewalt (2002), existem vários tipos de entrevistas, que podem ser classificados sob dois aspectos. Um deles é o grau de controle que o entrevistador tem sobre a entrevista e o outro é o grau de uniformidade das questões perguntadas aos informantes. A Figura 12 ilustra os tipos de entrevistas segundo estes aspectos:

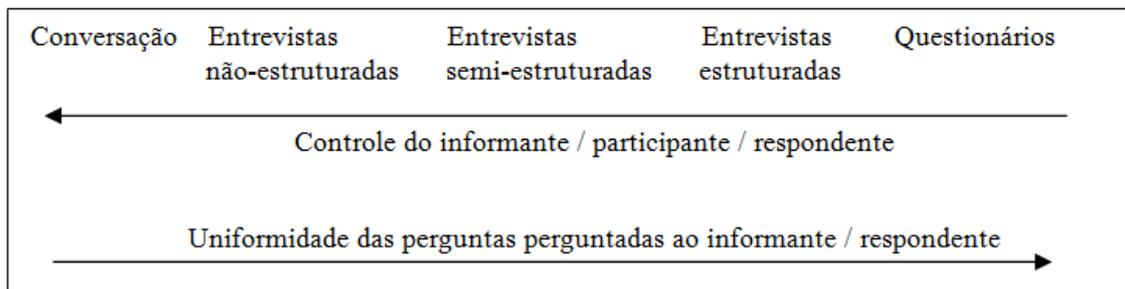


Figura 12 – Tipos de entrevistas (Adaptado de Dewalt e Dewalt, 2002).

Considerando o controle que o pesquisador tem sobre a entrevista: em um extremo está a *conversaão*, onde o pesquisador não possui controle nenhum sobre a entrevista, pois simplesmente não participa da conversa e atua como mero observador da situação. Já no outro extremo estão os *questionários*, cujas perguntas são completamente controladas pelo pesquisador sem haver espaço para a opinião pessoal de um respondente.

Considerando o grau de uniformidade das perguntas: em um extremo reside a conversaão na qual não se pretende seguir um roteiro ou perguntar a mesma pergunta para cada informante. Os questionários estão no outro extremo uma vez que, quando aplicados, apresentam exatamente as mesmas perguntas para cada respondente.

Entre um extremo e outro estão as entrevistas, que podem ser subdivididas em não-estruturadas, semi-estruturadas e estruturadas. Nas entrevistas não-estruturadas, o pesquisador pode até seguir um roteiro para realizar a entrevista, porém os tópicos são apresentados abertamente de modo que o pesquisador exerça o mínimo de controle possível sobre a entrevista. Já as entrevistas semi-estruturadas são guiadas por um roteiro de perguntas e dicas que podem ser cobertas totalmente ou não durante a entrevista. Finalmente, nas entrevistas estruturadas, o entrevistador garante que todas as perguntas do guia serão perguntadas para todos os informantes, isto é, este tipo de entrevista é orientado por perguntas que estão claramente definidas no guia.

Este trabalho utilizou entrevistas não-estruturadas e semi-estruturadas.

### **3.2.2 Observações**

De acordo com Yin (2003), existem dois tipos de observações: a participante (ou participativa) e a direta (ou não participativa).

Nas observações participantes, o pesquisador pode assumir várias funções dentro do estudo podendo, inclusive, participar dos eventos que estão sendo observados. Algumas de suas vantagens são: a) permitir que o pesquisador participe de eventos ou de grupos que são, de outro modo, inacessíveis à investigação científica, b) possibilitar que o pesquisador perceba aspectos sob o ponto de vista de alguém “de dentro” e não sob o ponto de vista externo, entre outras. Dentre as suas desvantagens estão: a) a função do participante pode exigir mais atenção que a função de observador e o pesquisador pode não ter tempo de realizar suas anotações, b) se a organização estudada estiver fisicamente dispersa, o pesquisador pode achar difícil estar no lugar certo na hora certa, tanto para participar de acontecimentos quanto para observá-los, entre outras.

Já nas observações diretas (ou não participativas), o pesquisador não está diretamente envolvido com a situação que está sendo observada (Shull et al, 2007). As observações diretas servem para fornecer informações adicionais sobre o tópico que está sendo estudado e podem ser realizadas, de maneira mais informal, juntamente com as entrevistas.

Este trabalho utilizou apenas observações diretas.

### 3.3 A Teoria Fundamentada em Dados

As próximas subseções mostram o que motivou a escolha da teoria fundamentada em dados como método para análise de dados, e alguns conceitos fundamentais a respeito.

#### 3.3.1 Motivações

Como visto na seção anterior, enquanto método qualitativo de pesquisa, a etnografia não se preocupa em definir hipóteses *a priori*, apenas a ampla pergunta de pesquisa que objetiva responder. Isto significa que ao realizar um estudo etnográfico, o pesquisador não está interessado em provar uma teoria, mas entender uma cultura, e desta forma, identificar uma situação vivenciada por uma determinada comunidade e que seja especialmente interessante para o seu trabalho.

De acordo com Glaser e Strauss (1967), a teoria fundamentada em dados é indutiva e não se preocupa em testar teorias ou hipóteses pré-concebidas. Pelo contrário, ao se utilizar a teoria fundamentada em dados, espera-se entender uma circunstância: como uma determinada comunidade age em seu dia-a-dia? Por que um determinado acontecimento ocorre desta ou daquela maneira?

O objetivo inicial deste trabalho foi realizar um estudo etnográfico para identificar os problemas enfrentados por engenheiros de software envolvidos em projetos de DDS na empresa Alpha. Desta forma, nunca houve uma hipótese a ser provada; pretendia-se apenas estudar os engenheiros de software da empresa Alpha de forma abrangente para, então, identificar problemas e propor soluções. Dentro deste contexto, a teoria fundamentada em dados (do inglês, *Grounded Theory* – GT) (Glaser e Strauss, 1967) se mostrou apropriada porque permitia que teorias emergissem a partir dos dados coletados nas observações e entrevistas, o que coincidiu com o objetivo inicial deste trabalho: propor soluções (teorias) baseadas nos problemas identificados (dados coletados) durante a realização de tarefas de ER em projetos de DDS (circunstância a ser estudada).

#### 3.3.2 Definição

Segundo Strauss e Corbin (1990), uma teoria fundamentada em dados é aquela derivada indutivamente a partir do estudo do fenômeno que a representa. Ou seja, a teoria é descoberta, desenvolvida e provisoriamente verificada através da coleta e da análise sistemáticas de dados. O indivíduo que utiliza a teoria fundamentada em dados não cria a teoria para depois

prová-la. Ao invés disso, a pessoa estuda uma determinada ocasião e os aspectos interessantes emergem desta investigação, compondo uma teoria.

A teoria fundamentada em dados está baseada no conceito de codificação. Durante a coleta de dados, o pesquisador reúne informações sobre uma determinada situação por meio de anotações de campo advindas de entrevistas, observações, inspeção de documentos, etc. Codificar, neste contexto, significa atribuir rótulos/nomes/códigos ao fenômeno observado, os quais devem fazer sentido para o pesquisador, de acordo com o objetivo da sua pesquisa.

A teoria fundamentada em dados corresponde a um processo gradativo de coleta e análise de dados. Este processo está dividido em quatro etapas principais: 1) coleta de dados; 2) codificação de dados; 3) comparação de dados, e; 4) criação da teoria, tal como mostra a Figura 13.



Figura 13 – Passos da teoria fundamentada em dados.

No caso deste trabalho, a coleta de dados ocorreu por meio de um estudo etnográfico (a ser detalhado no capítulo 4), no qual foram realizadas observações diretas (ou não participativas) e entrevistas semi e não-estruturadas. Em seguida, a codificação e a comparação dos dados foram realizadas com o auxílio da aplicação MAXqda2 (MAXqda2, 1995). A codificação dos dados permitiu a estruturação da informação a partir da criação de códigos. Na comparação, identificaram-se relacionamentos entre os códigos criados. Por fim, o fenômeno observado foi explicado a partir dos dados obtidos e a teoria foi originada, dando origem à discussão tratada no capítulo 5.

Vale ressaltar que, embora a figura não mostre adequadamente, a coleta e a análise dos dados ocorrem continuamente, isto é, na medida em que os dados são analisados, novos dados devem ser coletados para serem analisados e assim por diante. Este processo deve parar

quando a análise dos dados não mais sugerir a criação de novos rótulos ou relacionamentos entre os rótulos.

### **3.4 O Contexto organizacional do estudo etnográfico**

Este trabalho descreve um estudo etnográfico realizado em uma empresa de desenvolvimento de software chamada *Alpha*. A empresa Alpha é uma empresa federal que presta serviços de tecnologia da informação. Sua sede está localizada em Brasília e seus pólos estão distribuídos em onze estados brasileiros. Ao todo, o corpo de funcionários da empresa Alpha conta com aproximadamente dez mil empregados, dos quais aproximadamente quinhentos estão no pólo paraense. Geralmente, os clientes da empresa Alpha são empresas e órgãos públicos federais que controlam setores de segurança, de previdência social e de bancos, entre outros.

A empresa Alpha adota um processo bem definido para suas atividades de desenvolvimento de software. Este processo considera diversos papéis, alguns dos quais serão bastante citados no decorrer do texto, por isso os mesmos são apresentados a seguir:

- Analista de sistemas: é o analista que exerce tarefas de requisitos, codificação, testes e banco de dados (apenas “fiscalização” do modelo de dados, isto é, transformar o modelo de dados em tabelas no banco de dados);
- Administrador de dados: é o analista responsável por elaborar o modelo de dados que será “fiscalizado” pelo analista de sistemas. Este analista está localizado no Rio de Janeiro; e
- Coordenador de negócios: é a pessoa responsável por intermediar o contato entre os clientes e a equipe de desenvolvimento em termos de escopo e prazo das demandas.

Na época da condução deste estudo, o pólo paraense da empresa Alpha possuía três equipes que seguiam as práticas do nível 2 do SW-CMM, das quais apenas uma foi observada. Esta equipe era composta por um líder de projeto e analistas de sistemas, contabilizando um total de 12 pessoas. Optou-se por acompanhar o processo de desenvolvimento de um único sistema, referenciado aqui por SYS.

Em virtude da alta rotatividade de funcionários observada na empresa, ou seja, da entrada e saída frequente de funcionários, as pessoas que exercem o cargo de analista de

sistemas estão envolvidas com todos os tipos de tarefas, desde a elicitação de requisitos até a execução de testes no sistema desenvolvido.

A equipe observada pratica desenvolvimento distribuído de software porque seu principal cliente está situado em outro estado (Figura 14). Desta forma, sempre que um novo sistema ou uma nova funcionalidade são solicitados, é necessário o deslocamento, ou da equipe de desenvolvimento (com a ida de um analista para a cidade dos clientes) ou do cliente (com a vinda de um ou mais clientes para a cidade dos analistas). Além disso, os desenvolvedores dependem de outro funcionário da empresa que desempenha o papel de administrador de dados (AD) e que está situado em uma terceira cidade.

Em resumo, o tipo de distribuição geográfica existente na empresa Alpha é caracterizado por a equipe de desenvolvimento estar localizada em Belém enquanto que seu principal cliente, em Brasília. A distância entre clientes, desenvolvedores e administradores de dados causa claros impactos em todas as atividades do processo de Engenharia de Requisitos, conforme será apresentado no próximo capítulo.



Figura 14 – Clientes e analistas da empresa Alpha estão separados por uma distância aérea de 1593 km.

### 3.5 Coleta e análise de dados

Inicialmente, um acordo de colaboração foi estabelecido entre a empresa Alpha e a Universidade Federal do Pará. Este acordo previa a realização de um estudo etnográfico nas dependências da empresa. O critério utilizado para a escolha da equipe observada foi a etapa do processo que estava em andamento naquele momento. Uma das equipes havia acabado de

ser designada para tratar uma demanda de manutenção evolutiva em um sistema, logo havia necessidade de nova elicitação de requisitos. Como o foco deste trabalho era investigar as atividades de requisitos, esta equipe foi a selecionada como alvo da pesquisa. Uma vez que o acordo foi fechado entre a UFPA e a empresa Alpha, o estudo etnográfico pôde ser iniciado. A coleta de dados foi realizada através de entrevistas e observações.

A princípio, duas entrevistas exploratórias, cada uma com duração média de uma hora, foram realizadas com os líderes de projeto da empresa. A partir destas entrevistas, foi possível obter uma visão geral sobre a empresa, o processo adotado e alguns dos problemas enfrentados.

Além das entrevistas exploratórias, observações não-participativas (Jorgensen, 1989) foram realizadas tanto no ambiente local da empresa Alpha quanto no ambiente remoto do cliente. Considerando o contexto remoto, as observações se resumiram a duas viagens à cidade do cliente para o acompanhamento de uma reunião de elicitação de requisitos (fev/2008) e outra de validação de requisitos (abr/2008). A elicitação de requisitos se estendeu por cinco dias, oito horas de reunião diária. Estavam presentes um cliente, um analista da empresa Alpha, o administrador de dados e o coordenador de negócios<sup>7</sup>. Já a reunião de validação de requisitos durou apenas três dias, também oito horas de reunião diária. Nesta, estavam presentes em média de três a quatro clientes, um analista da empresa Alpha e o coordenador de negócios.

No contexto local, os analistas foram observados realizando, especialmente, atividades de documentação e de modelagem de requisitos, além de reuniões para acompanhamento de projeto, em que toda a equipe costumava estar presente. As observações no contexto local foram realizadas aproximadamente em meio expediente, duas vezes por semana, durante cinco meses (fev/2008 a jun/2008).

Além das observações, entrevistas não e semi-estruturadas (Dewalt e Dewalt, 2002) foram realizadas para refinar aspectos notados durante as observações. As entrevistas ocorreram no mesmo período das observações (fev/2008 a jun/2008), sendo a duração de cada entrevista, em média, uma hora e trinta minutos. No entanto, as entrevistas com o cliente e o coordenador de negócios não obedeceram este padrão, pois só ocorreram em abr/2009 em

---

<sup>7</sup> O coordenador de negócios trabalha na empresa Alpha e é a pessoa que media o relacionamento entre o cliente e o pólo de desenvolvimento da empresa Alpha.

razão da disponibilidade tanto dos informantes quanto dos pesquisadores e duraram aproximadamente três horas cada. Em todos os casos, as entrevistas foram gravadas com o consentimento dos entrevistados e, portanto, puderam ser transcritas em sua totalidade.

Os entrevistados foram analistas de sistemas (não existe divisão em papéis nas equipes, todos executam todo tipo de tarefa), líderes de projeto, o coordenador de negócios e o principal cliente da equipe estudada. Tal como as observações, as entrevistas abordaram assuntos relacionados a atividades de requisitos, como elicitação, especificação e validação de requisitos.

Os guias de entrevistas foram elaborados a partir de uma pergunta geral, denominada de *Grand Tour* (Dewalt e Dewalt, 2002), e que procurava direcionar todas as demais perguntas: “Como a distância afeta as suas atividades?”. De um modo geral, as perguntas foram divididas de acordo com as etapas do processo de requisitos (elicitação, documentação, validação e gerência de requisitos). Além disso, o guia também é composto por perguntas de informação geral que normalmente refletiam dúvidas ou questões levantadas durante as observações. Não houve um piloto dos guias de entrevistas dada a natureza qualitativa deste estudo. Os guias utilizados durante as entrevistas podem ser encontrados nos apêndices deste documento.

Um resumo das atividades realizadas durante a coleta de dados está ilustrado na Tabela 1. No total, pode-se observar que 241 horas foram realizadas de coletas de dados.

Qtde	Método de coleta de dados	Local	Papel do informante	Objetivo	Duração
2	Entrevista exploratória	Empresa Alpha	Líderes de projeto	Visão geral da empresa	2 entrevistas de 1h, cada, ou seja, 2h
-	Observações não-participativas	Empresa Alpha	Analistas de sistema	Atividades de especificação e de modelagem de requisitos	Meio expediente, 2x por semana durante cinco meses (fev'08 – jun'08), ou seja, 160h
-	Observações não-participativas	Contexto do cliente	Cliente(s), analista de sistema e coordenador de negócios	Reunião de elicitação de requisitos e de validação de requisitos	Elicitação (cinco dias em fev'08) e validação (três dias em abr'08), ou seja, 64h
10	Entrevistas não e semi-estruturadas	Empresa Alpha	Analistas de sistema, líderes de projeto, coordenador de negócios e cliente	Detalhar atividades de requisitos e refinar aspectos notados nas observações (questionário-guia)	10 entrevistas com duração de 1h30, cada, ou seja, 15h

Tabela 1 – Atividades realizadas durante a coleta de dados na empresa Alpha.

Nem todas as pessoas da equipe foram analisadas. Cinco analistas (Aa, Ab, Ac, Ad e Ae), dois líderes de projeto (La e Lb) e o coordenador de negócios (C) foram entrevistados, pois eram indivíduos que ou possuíam alguma experiência ou estavam diretamente envolvidos com as atividades de requisitos. Dois analistas (Ad e Ae) foram observados no

contexto da empresa, pois eles eram os responsáveis pelas atividades de requisitos na época do estudo (elicitação, modelagem e especificação de requisitos). Além disso, dois analistas (Ab e Ae), o coordenador de negócios e o próprio cliente foram observados no contexto do cliente durante as reuniões de elicitação (Ab) e de validação (Ae) de requisitos. O tempo de empresa dos analistas e dos líderes na época da condução do estudo etnográfico variava aproximadamente de dois a doze anos, sendo os líderes os funcionários mais antigos. Já o coordenador de negócios era funcionário da empresa há vinte e cinco anos.

As observações foram registradas através de notas de campo. Nestas notas de campo procurou-se anotar o que acontecia no momento de cada observação, por exemplo, a seqüência e o detalhamento de atividades executadas pelos informantes, suas falas (geralmente mencionadas entre aspas na nota de campo) e as impressões pessoais da autora. Já as entrevistas, como dito anteriormente, puderam ser gravadas e conseqüentemente transcritas.

Uma vez que os dados começaram a ser coletados, técnicas da teoria fundamentada em dados foram aplicadas com o intuito de analisá-los para que, a partir destes, pudessem ser estabelecidas hipóteses sobre: a) como a distância influenciava negativamente as atividades da Engenharia de Requisitos na empresa Alpha, e b) como *stakeholders* envolvidos em projetos da empresa Alpha buscavam mitigar os efeitos da distância. Dentre as técnicas, utilizaram-se os diferentes tipos de codificação para estruturar os dados coletados: codificação aberta, codificação axial e codificação seletiva. A codificação foi realizada com a ajuda da ferramenta MAXqda2.

O MAXqda2 permite que códigos sejam criados e associados a trechos de textos resultantes de entrevistas transcritas ou de notas de campo. A Figura 15 ilustra como a ferramenta organiza os dados. O primeiro quadrante (*Document System*) mostra todos os textos do projeto, os quais podem ser reunidos em grupos de texto para facilitar a análise. O segundo quadrante (*Text Browser*) mostra o texto que foi selecionado do quadrante anterior. Este texto permanece habilitado para edição, ou seja, o usuário pode marcar segmentos do texto e associá-los a códigos. O terceiro quadrante (*Code System*) mostra as categorias, os códigos e os sub-códigos que foram criados durante a análise. O último quadrante (*Retrieved Segments*) exibe a seleção de segmentos de texto, associados a um determinado código.

The screenshot displays the MAXqda2 software interface during the coding phase of grounded theory. The interface is divided into several panels:

- Document System (Top Left):** Shows a hierarchy of text groups. A red '1' is placed next to the 'Entrevistas' group, which has a count of 467. The total count for all groups is 782.
- Code System (Bottom Left):** Shows a hierarchy of codes. A red '3' is placed next to the 'Solucoes metodologicas' code, which has a count of 11. The total count for all codes is 782.
- Text Browser (Top Right):** Displays a text segment with a red '2' at the beginning. The text describes a meeting in Salvador and the development of a system. Line numbers 13, 14, 15, 16, and 17 are visible.
- Retrieved Segments (Bottom Right):** Shows a list of retrieved segments. A red '4' is placed next to the first segment, which has a count of 100. The text describes a meeting in Belém and the role of the development team.

The interface also includes a menu bar (Project, Edit, Codes, Memos, Attributes, TextSearch, Analysis, Windows) and a toolbar with various icons for document management and analysis.

Figura 15 – Utilização do MAXqda2 durante as etapas de codificação da *Grounded Theory*.

Inicialmente, realizou-se a etapa de codificação aberta, na qual os códigos foram criados com o intuito de simplesmente identificar ou nomear as situações encontradas nos textos (notas de campo e transcrições de entrevistas). Ainda na codificação aberta, os códigos foram agrupados em categorias para reduzir o conjunto de dados sobre o qual se iria trabalhar e, assim, facilitar a construção das teorias. Ao fim da codificação aberta, 782 códigos foram criados e organizados como mostra a Tabela 2 que, apresenta apenas um subconjunto destes códigos, dado o número total de códigos criados. As colunas Códigos e Categorias resultaram da codificação aberta. Todos os códigos podem ser consultados no Apêndice II desta dissertação.

Em seguida, a etapa de codificação axial foi realizada para relacionar as categorias criadas anteriormente umas com as outras a partir de raciocínios ora indutivos ora dedutivos, resultando, por exemplo, em uma relação de causa e consequência do tipo: deduziu-se que a “centralização do conhecimento” *leva* a equipe de desenvolvimento a adotar estratégias para “disseminação do conhecimento”.

Finalmente, durante a codificação seletiva, escolheram-se três principais categorias a partir das quais as teorias seriam escritas: pontos negativos ou problemas, pontos positivos ou boas práticas e sugestões ou oportunidades de melhoria. Em outras palavras, as teorias emergiram das relações encontradas entre os códigos e categorias inicialmente criados e foram, por sua vez, organizadas em três principais categorias.

#	Códigos	Categorias	Principais categorias
1	Falta de dedicação dos clientes	Problemas na elicitação de requisitos à distância	Problemas ou pontos negativos
2	Reunião muito curta		
3	Ausência de videoconferência	Problemas na comunicação	
4	Demora nas respostas		
5	Centralização do conhecimento	Problemas na gerência de conhecimento	
6	Não existe registro de <i>rationale</i>		
7	Requisitos são compreendidos melhor	Validação de requisitos junto ao cliente	Boas práticas ou pontos positivos
8	Reunião dinâmica		
9	Disseminação do conhecimento	Boas práticas de gerência de conhecimento	
10	Registro de <i>rationale</i>		
11	Adaptar o processo	-	Sugestões ou oportunidades de melhoria
12	Enviar mais analistas	Melhorias na comunicação	
13	Utilizar recursos audiovisuais		

Tabela 2 – Organização dos códigos, categorias e principais categorias.

As seções seguintes apresentarão a análise realizada sobre os dados coletados durante o estudo etnográfico. As teorias foram construídas a partir de três grandes conjuntos de dados (principais categorias): a) pontos negativos, sobretudo, problemas mencionados pelos entrevistados/observados, b) pontos positivos, ou seja, boas práticas já utilizadas pela equipe de desenvolvimento e pelos clientes da empresa Alpha, e c) oportunidades de melhoria, isto é, recomendações sugeridas pela autora deste trabalho; todos voltados para a Engenharia de Requisitos em contextos de Desenvolvimento de Distribuído de Software. O último conjunto de dados será tratado no capítulo de discussão uma vez que refletem a opinião pessoal da autora.

## **4 UM ESTUDO ETNOGRÁFICO NA EMPRESA ALPHA**

Este capítulo apresenta os pontos negativos e positivos identificados a partir da análise dos dados coletados na empresa Alpha. Conforme discutido no capítulo anterior, duas formas de coleta de dados foram utilizadas neste estudo: entrevistas não e semi-estruturadas e observações diretas (ou não-participativas). Solicita-se, portanto, que o leitor utilize a legenda [ent] para trechos retirados de entrevistas ou [notes] para notas de campo (observações).

### **4.1 Pontos negativos**

Visando facilitar o entendimento do texto, esta seção está organizada segundo o processo genérico de ER, citado na seção 2.1.1, que considera as seguintes etapas: elicitação, modelagem, negociação, especificação e validação de requisitos. Em alguns casos, os problemas eram transversais ao processo de ER, ou seja, eles se aplicavam em todas as etapas do processo. Desta forma, os pontos negativos descritos nesta seção foram divididos em dois grupos: a) problemas enfrentados durante o processo de ER e b) problemas transversais ao processo de ER.

#### **4.1.1 Problemas enfrentados durante o processo de ER**

Os dados coletados sugerem que a empresa Alpha enfrenta problemas nas diversas etapas da ER. Alguns destes problemas são exacerbados pelas peculiaridades decorrentes da distância geográfica que existe entre clientes e o pólo de desenvolvimento da empresa Alpha.

##### *4.1.1.1 Elicitação de requisitos*

A elicitação de requisitos na empresa Alpha pode acontecer à distância ou presencialmente. No primeiro caso, um analista de requisitos é deslocado para a cidade do cliente, faz o levantamento de requisitos através de reuniões, que ocorrem normalmente durante uma semana (cinco dias úteis), retorna para o pólo de desenvolvimento e repassa o

conhecimento aos colegas. Já no segundo caso, a equipe de clientes é deslocada para o pólo de desenvolvimento, possibilitando que todos os analistas do projeto tenham a chance de ouvir os clientes. Esta seção mostrará quais são os pontos negativos presentes no primeiro caso, ou seja, quando a elicitação de requisitos ocorre à distância. O segundo caso não será abordado, pois a equipe observada não realizou elicitações desta natureza durante o estudo etnográfico.

Nas observações realizadas durante as reuniões de levantamento de requisitos na cidade do cliente, notou-se que em muitas situações o analista enviado não é capaz de elucidar as dúvidas dos clientes. A nota de campo abaixo mostra uma situação em que o analista enviado à cidade do cliente está há muito tempo sem contato com o sistema:

*[notes] O cliente pergunta sobre um erro que acontece em uma determinada funcionalidade do sistema. O analista mal deixa o cliente terminar de falar e já diz que não sabe responder essa pergunta. Noto que o analista não sabe responder muitas perguntas e sempre argumenta que precisa voltar à Belém para conseguir as respostas junto à equipe de desenvolvimento.*

O trecho acima foi retirado de notas de campo que registraram o levantamento de requisitos na cidade do cliente. Já o texto apresentado abaixo relata uma ocasião em que outro analista foi deslocado para realizar uma reunião de validação de requisitos junto ao cliente. É importante ressaltar que o analista das primeiras reuniões (levantamento de requisitos) não foi o mesmo presente nas últimas (validação de requisitos). Além disso, a situação dos analistas é distinta uma da outra, pois o primeiro se ausentou por um tempo do projeto e foi realocado para trabalhar em outros projetos, enquanto que o segundo nunca se afastou do projeto.

*[notes] Nas reuniões de levantamento de requisitos, existiam muitos momentos de impasse, isto é, o primeiro analista, por estar afastado do SYS há muito tempo, tinha uma série de dúvidas, não sabia responder uma série de perguntas, não conseguia contestar os argumentos do cliente. Já o segundo analista, que sempre trabalhou no sistema em questão, mantém uma postura diferente na reunião de validação de requisitos: ele é capaz de interagir mais com os clientes, de entender suas perguntas e respondê-las com conhecimento de causa.*

Ao ser realocado para trabalhar em outro projeto, o primeiro analista perdeu contato com os requisitos do SYS e, por isso, se sentiu perdido nos primeiros dias de reunião com o cliente. Situações como estas prejudicam projetos de DDS, uma vez que o deslocamento de pessoas é caro e, portanto se torna raro, e o tempo é curto, por isso, deve ser aproveitado da forma mais inteligente possível.

Nesta situação, corre-se também o risco de o analista ser deslocado para a reunião com os clientes e se deparar com uma equipe de clientes que não está em consenso quanto aos requisitos do sistema. A nota de campo abaixo foi registrada em uma reunião de levantamento de requisitos. Na ocasião, o analista aproveita a oportunidade para falar sobre quando os analistas não devem estar presentes em uma reunião:

*[notes] Ele [o analista] acha que o analista de requisitos não precisa estar presente quando os clientes não estão certos do que querem. Ele conta um caso em que já esteve em uma reunião em que ‘umas 10 pessoas [clientes] ficaram discutindo entre si’ enquanto que ele ficou só olhando. Isto, segundo ele, não é produtivo para o analista de requisitos: ‘as pessoas não sabem diferenciar quando é uma reunião de especificação e quando não é. Os clientes chamam esse tipo de reunião como reunião de “pré-especificação”’.*

Não só a presença do analista de requisitos é questionada, mas também a presença do AD (administrador de dados). Os trechos a seguir foram retirados da entrevista com o coordenador de negócios: o primeiro trecho fala sobre o acionamento do AD durante o levantamento de requisitos de uma nova demanda e o segundo, durante a homologação dessa demanda.

*[ent] “A primeira reunião que tem de especificação [elicitação], eu chamo o AD. Se a reunião for feita no pólo de desenvolvimento, o AD vai pra lá. Se for feita aqui em Brasília, convoco ele pra cá e eu vou também. Aonde seja, ele se desloca pra participar da reunião. Se for um projeto novo, nas reuniões iniciais, ele participa de tudo. Então se é uma reunião inicial, um projeto novo, vai ter impacto em outros sistemas, o AD tem que participar. Então o AD vem na primeira reunião. Aí na segunda reunião – porque nunca se fecha um projeto numa primeira reunião – a gente avalia se vai ser preciso ter o AD em todos os dias. A gente consulta o desenvolvimento se há necessidade. Se eles disserem lá que ainda tem muita coisa pra especificar, a gente chama o AD nos cinco dias. Aí na terceira reunião (se houver), a gente avalia de novo se precisa. Aí eles [os membros da equipe de desenvolvimento] já podem dizer lá que só precisam do AD nos últimos três dias porque nos primeiros dias vão discutir muita dúvida que ficou com o cliente, então a presença do AD não vai ser útil. Agora sobre a reunião de fechamento [validação de requisitos] que vai servir de entendimento pra elaboração do DRS [documento de especificação de requisitos], geralmente eu chamo o AD por uns dois ou três dias. (...) Pode ser por áudio pra conversar com o desenvolvimento pra dizer quando eles vão entregar o modelo. Ou pode ser presencialmente, por uns dois ou três dias, pra fechar o entendimento [isso quando ainda tem muita dúvida]. Isso num sistema novo. Se for uma demanda de um sistema já existente, vai ter um diminutivo. Começou a construção, o AD desaparece. Nos pontos de controle também”.*

*[ent] “Na homologação, ele está presente, mas não durante a semana inteira. Por que eu convoco o AD na homologação? Pra ele checar se as regras deles tão sendo atendidas. Se está tudo certo*

*o que foi definido lá nos modelos do banco. Também convoco ele porque de repente os clientes pedem pra adicionar alguma funcionalidade, aí o AD avalia se vai haver ou não impacto no banco, já subsidia [a empresa Alpha] dando um diagnóstico, dizendo se a alteração é grande, se ela é pertinente”.*

O coordenador de negócios explica que nem sempre o AD é solicitado na reunião presencial de levantamento de requisitos. No trecho abaixo, o coordenador explica quando o AD é solicitado e quando não é:

*[ent] “Se for uma demanda menor, o desenvolvimento mesmo pode me dizer que vai haver impacto no banco, que eles vão precisar conversar com o AD. Aí eles fazem uma áudio. De repente, nem precisa de ter a reunião presencial. Mas, se for uma alteração muito grande, eles lá [da equipe de desenvolvimento] me dizem que vão precisar de uma reunião de especificação com o AD. Aí, mesmo sendo uma demanda de manutenção de algo que já existe, a gente traz o AD. Às vezes, eles dizem pra trazer só dois dias. Nunca é toda a semana”.*

Tudo que foi apresentado até então sobre a presença do AD corresponde à opinião do coordenador de negócio, que diverge da opinião dos analistas. Durante as reuniões de levantamento de requisitos, o analista afirmou que o AD não deveria estar presente em todas as reuniões desde o início:

*[notes] O analista diz que não concorda que os ADs devam estar presentes em todas as reuniões. Isto deve depender do cliente. Se o cliente for do tipo que já chega com os requisitos fechados, é interessante ter o AD desde o início, mas quando o cliente é do tipo que “começa falando numa coisa, daqui a pouco já tá falando em outra”, não tem necessidade do AD.*

Finalmente, a opinião do AD sobre sua presença nas reuniões de levantamento de requisitos é a seguinte (trechos também retirados das reuniões de elicitação de requisitos):

*[notes] O AD acha que os ADs deveriam estar presentes e afirma ‘pensou em reunião de especificação, pensou em AD’. O AD diz que antigamente, as reuniões de especificação aconteciam e ele só era chamado na última reunião, quando o assunto já estava bem amadurecido, às vezes já existia até um modelo de classes pronto pra ele. Ele não concorda com isso.*

Outro problema identificado quando a elicitação ocorre à distância são as estimativas. Acabada a reunião, os clientes perguntam em quanto tempo os requisitos serão documentados e implementados. Este tipo de resposta não pode ser dado neste momento, pois o analista se encontra distante de sua equipe e impossibilitado de estimar o tamanho dos requisitos e, conseqüentemente, o prazo necessário. O texto abaixo foi retirado de uma entrevista com um analista:

*[ent] Pesquisadora: “Durante a última reunião, o cliente reclamou de prazos?”*

*Analista: “Não, não. No final da reunião, o coordenador de negócios chegou pra negociar quando é que a gente vai documentar isso tudo que foi tratado em BSB e enviar a documentação completa. Ele deu o prazo de uma semana. (...) Eu viria pra cá [Belém], faria o repasse, todo mundo documentaria e mandaria pra ele. Aí o cliente perguntou ‘ah tudo bem, mas eu quero saber quando tudo vai terminar?’. Eu respondi: ‘como saber? Agora mesmo a gente não pode te dar um prazo. Acabamos de elicitar aqui as coisas. Não temos como te dizer que dia tal tá entregue. Vai passar pela análise da equipe lá, depois quando estiver na parte do cronograma, a gente vai te dar o prazo’. Ele disse: ‘ah, mas eu queria isso pra ontem...’ – é sempre assim. Cliente sempre quer as coisas o mais rápido possível”.*

Embora seja natural e aceitável para quem é da área de tecnologia, o fato de o analista não ser capaz de dar um prazo ao fim da reunião de elicitação de requisitos, isto se torna pouco compreendido pelo cliente. Situações de cobrança de prazos, além de citadas em entrevistas (como o trecho acima), também foram registradas em notas de campo, como mostra o trecho abaixo. Na ocasião, o cliente está falando com o analista sobre uma determinada funcionalidade do sistema:

*[notes] O cliente pergunta qual a complexidade para se fazer o requisito, mas o analista já adianta que é difícil para ele estimar a complexidade de qualquer requisito naquele momento. Imagino que o cliente esteja perguntando isso para saber quanto tempo a mudança levaria para ser desenvolvida, porém o analista precisa da ajuda do pessoal em Belém para estimar.*

Quando a elicitação de requisitos na empresa Alpha ocorre remotamente, ou seja, com o deslocamento de um analista para a cidade do cliente, os resultados sugerem que falta dedicação por parte dos clientes:

*[ent] Pesquisadora: “Você percebe alguma diferença entre você ir e eles [os clientes] virem?”*

*Analista: “Eu acho melhor quando eles vêm porque aqui [no pólo de desenvolvimento], eles estão fora do ambiente deles e eles se concentram. Lá não. Lá eles fazem o horário que eles quiserem, toca telefone... Prefiro quando eles vêm pra cá que aqui a gente controla, a gente não deixa as coisas atrapalharem. Já eles lá não, eles não ficam em tempo integral”.*

Esta falta de dedicação, além de citada em entrevistas, foi presenciada nas observações realizadas durante o levantamento de requisitos, que ocorreu na cidade dos clientes. Na ocasião, observou-se que os clientes se retiravam inúmeras vezes da reunião, o que deixava o analista bastante desconfortável. Dentre os motivos para os clientes se ausentarem estavam atender ao celular e ir ao médico, sugerindo que os analistas tinham razão ao dizerem que quando a reunião ocorria no ambiente dos clientes, os mesmos não se dedicavam plenamente.

A ida de apenas um analista para a reunião de elicitação de requisitos ocasiona diversos problemas no processo de desenvolvimento de software. Um deles é a dificuldade do analista em ter foco nas reuniões de levantamento de requisitos. O trecho abaixo foi retirado de uma entrevista com um analista que já se deslocou para elicitar requisitos junto aos clientes. Na ocasião, ele fala sobre a importância da ata da reunião durante a elaboração do documento de visão<sup>8</sup>.

*[ent] “[A ata] Também ajuda a adicionar aspectos que o analista não identificou como sendo elementos do DV durante a reunião em BSB, por falta de tempo, de foco, afinal o analista se preocupa com várias coisas ao mesmo tempo”.*

Como o analista tem pouco tempo junto aos clientes, ele deve se esforçar para conseguir o máximo de informação possível. As reuniões normalmente são cansativas, pois os envolvidos discutem um grande volume de informação. Há casos em que diversos clientes discutem vários requisitos ao mesmo tempo e existe apenas um analista para ouvir, entender e registrar todas as decisões. O registro, em particular, é bastante problemático, como mostra o trecho abaixo, retirado de uma entrevista com um analista:

*[ent] “(...) [acontece] de você estar discutindo com o cliente um assunto e ter que entender aquilo e ter que escrever ao mesmo tempo. Às vezes, é um conceito um bocado complicado que você demora pra entender o que o cara tá querendo explicar. Depois que você entende ‘Ah, entendi! Então perai que vou ter que escrever agora’. Aí atrasa mais um pouquinho a reunião porque eu vou ter que escrever (escrever mesmo na agenda, num bloco de notas) pra eu poder armazenar aquela informação em algum lugar, ao passo que se tivesse outra pessoa, eu acho, que seria bem mais ágil”.*

Outro problema é a dificuldade no repasse do conhecimento. O analista que se encontra com os clientes, ao voltar para o pólo de desenvolvimento tem que repassar o conhecimento aos demais colegas e passa a atuar como um “substituto do cliente”. O trecho abaixo foi retirado de uma entrevista com um analista:

*[ent] “Outra dificuldade que a gente tem é essa questão de só ir uma pessoa especificar [elicitar os requisitos]. Esse repasse pra equipe depois é um pouquinho complicado. A gente passa um tempo tendo dificuldades. A equipe tem dificuldade de entender. Depois que nivela o conhecimento, fica bem melhor, mas no início, nas primeiras semanas, é difícil”.*

---

<sup>8</sup> Segundo os dados coletados, o documento de visão (DV) é um artefato inicial que armazena as necessidades e as funcionalidades dos clientes. Sua elaboração antecede a elaboração do documento de especificação de requisitos (DRS).

Durante a reunião de repasse, é provável que muitos analistas levantem questões que não ocorreram ao analista *liaison*<sup>9</sup> durante o levantamento de requisitos. É o risco de se deslocar apenas um analista: não se consegue estabelecer um entendimento comum acerca dos requisitos levantados, ou seja, o analista *liaison* não consegue pensar em todas as possibilidades, pois está sozinho junto aos clientes, o que possibilita voltar com requisitos incompletos. Os trechos abaixo foram retirados de entrevistas realizadas, respectivamente, com um analista que já se deslocou para a cidade do cliente e com um líder de projeto:

*[ent] “Pra mim, eu achei que ficou claro, mas na hora pode ser que uma outra pessoa [outro analista] venha e faça uma outra pergunta que de repente também era uma dúvida. Fica uma concentração [de informação] e uma sobrecarga muito grande em cima de uma pessoa”.*

*[ent] “A pessoa que foi tem que vir e fazer repasse pra toda a equipe. E aí é justamente o que eu não gosto. Porque o conhecimento fica em uma pessoa só, a interpretação fica em uma pessoa só. É muito fácil de ter deturpações de entendimento do que quando você tem várias pessoas ouvindo o mesmo cliente falando”.*

As notas de campo das reuniões de levantamento de requisitos mostraram requisitos incorretos que se transformaram em funcionalidades incorretas. Na ocasião, o cliente e o analista estão lendo o documento de um requisito que já foi inclusive implementado, mas que não atende às necessidades do cliente:

*[notes] Ao ler este arquivo [que contém a especificação do requisito], tanto o cliente quanto o analista se dão conta de que o requisito está de fato especificado errado. O cliente então deixa claro que o requisito está especificado errado e diz para o analista: “Analista A, eu só queria um botãozinho...”. O cliente diz isso porque o que está implementado é muito maior do que aquilo pedido por ele.*

Ainda, observou-se que clientes da área de informática podem prejudicar o bom andamento das reuniões de elicitação e do projeto como um todo. Este problema foi inicialmente citado por um líder de projeto durante sua entrevista e foi confirmado, posteriormente, durante as observações realizadas no contexto do cliente. Na ocasião da entrevista, o líder explicava como ocorria a elicitação de requisitos, ou seja, quais estratégias eram escolhidas quando ele precisava selecionar os analistas que se deslocariam para conversar com o cliente:

---

<sup>9</sup> O *liaison* é a pessoa que viaja entre os lugares onde os *stakeholders* estão localizados. Seu papel é facilitar o fluxo de comunicação entre os *stakeholders*, interligar suas culturas, mediar conflitos e resolver problemas de comunicação (Carmel e Agarwal, 2001).

*[ent] “Aí tem que tomar cuidado. Eu sempre coloco uma pessoa experiente com uma inexperiente porque, como alguns deles [dos clientes] também são da área de tecnologia, tem que ter cuidado porque eles [clientes da área de TI] já querem definir o físico [aspectos relacionados à arquitetura do sistema]. Então, tem que ter pessoas que sejam experientes”.*

O trecho acima sugere que se um analista inexperiente se encontrar sozinho com o cliente, ele pode ser influenciado. Embora o trecho indique uma estratégia utilizada pelo líder de projeto para mitigar este problema, isto não foi presenciado na prática, durante o estudo etnográfico: apenas um analista foi descolado para as duas reuniões observadas (levantamento e validação de requisitos).

Quando a equipe de clientes é composta por pessoas com formação na área de computação, as reuniões podem ser prejudicadas por atrasos desnecessários. A nota de campo abaixo mostra um dos clientes se interessando pelos aspectos técnicos do sistema:

*[notes] Um dos clientes pergunta pelo modelo de dados do SYS. Ele é da área de Computação e costuma interferir em conversas mais técnicas. Agora ele está conversando com o analista sobre como os dados são estruturados no SYS, sobre as tabelas do modelo de dados.*

Na seqüência, um dos clientes explica que uma determinada funcionalidade não está correta no sistema e que precisará ser modificada. O analista, por sua vez, explica o impacto causado por essa modificação:

*[notes] O analista começa a explicar detalhes bastante técnicos e mostra para o cliente qual é o impacto de isso ser modificado: reescrever rotinas X, Y e Z e testá-las. O analista fala que estas rotinas já estavam implementadas desde muito tempo atrás e que elas aparecem em inúmeras telas do sistema. Por causa disso, eles perdem um pouco de tempo pensando em soluções. A discussão foi motivada pelo cliente que é da área de TI.*

De um modo geral, alimentar discussões técnicas junto aos clientes causa uma perda de tempo indesejada. Além disso, isto pode desgastar a relação entre clientes e analistas, pois os clientes naturalmente, por não terem a visão geral do projeto, assumirão que as modificações são simples e que gastariam pouco tempo para serem implementadas:

*[notes] Agora o cliente de TI está estimando os riscos e diz que lhe parece simples modificar a “rotina da contagem”. Seu tom é irônico. (...) Por ele ser da área de computação, vive intervindo em assuntos mais técnicos, pedindo explicações que não podem nem devem ser dadas a clientes.*

Todos estes aspectos confundem a ordem da reunião e até o próprio analista. Entende-se que isto atrapalha a elicitação de requisitos e o processo de ER como um todo porque, dentro

do contexto da empresa Alpha, o tempo em que clientes e analistas se encontram presencialmente é muito curto e, portanto, precisa ser aproveitado de maneira otimizada.

De um modo geral, pode-se concluir que a distância entre os *stakeholders* também resulta em requisitos mal entendidos, mal especificados e, conseqüentemente, mal desenvolvidos. Isto acontece porque a distância impacta decisivamente na comunicação entre os *stakeholders* e prejudica as atividades do processo de ER das quais podem surgir requisitos de baixa qualidade. Em outras palavras: o cliente recebe funcionalidades que não satisfazem suas necessidades. O trecho abaixo foi retirado de uma entrevista com um analista. Neste momento, o analista explicava como foi seu primeiro contato com o SYS:

*[ent] “Então a gente já começou implementando no SYS. Eu, por exemplo, comecei com uma consulta lá. O chefe dizia ‘ah, faz assim, assim e assado e tem que trazer esse resultado’. Perfeito, funcionou, né? Aí um tempo depois eles [os clientes] vieram pra uma especificação de requisitos e aí eles tocaram nessas consultas que a gente fez e comentaram o que eles queriam. Aí que a gente foi entender, assimilar a questão dos requisitos que eles queriam e o que a gente tinha implementado. Realmente, o que a gente tinha implementado não tava satisfazendo. Satisfazia o nosso ponto de vista que era ‘ah com tais parâmetros, retorna tal coisa’, mas não satisfazia o que eles queriam”.*

Nota-se que o chefe na época foi primeira pessoa a entrar em contato com o cliente para levantar os requisitos do sistema. O chefe, naturalmente, entendeu os requisitos à sua maneira e repassou este conhecimento, também à sua maneira, aos seus colegas de equipe, que implementaram as funcionalidades da forma como as haviam entendido. Entretanto, no momento em que os clientes foram deslocados, entraram em contato com todos os desenvolvedores e puderam lhes explicar quais eram os requisitos de fato, o analista entrevistado se deu conta de que as funcionalidades por ele implementadas não satisfaziam as necessidades dos clientes.

#### *4.1.1.2 Modelagem de requisitos*

Como visto no capítulo 2, a modelagem de requisitos prevê a elaboração de diagramas para representar o sistema como um todo. Alguns dos diagramas previstos nesta etapa são os de caso de uso para representar as funcionalidades do sistema e o modelo de dados para representar as entidades, seus atributos e seus relacionamentos.

Na empresa Alpha, os analistas responsáveis pela elaboração do modelo de dados são os administradores de dados (ADs) e estão concentrados em uma cidade que não é a mesma onde

os desenvolvedores da empresa Alpha estão localizados, por isso estão envolvidos com uma série de problemas, descritos abaixo.

Os ADs estão em pouca quantidade e precisam atender muitas demandas que são provenientes de diversas partes do país. A distância dificulta o contato entre os ADs e os pólos de desenvolvimento, logo sobrecarregam os ADs de atividades, como é possível perceber nos seguintes trechos:

*[ent] “As demandas são muitas e os ADs, poucos. Eles vivem tendo que se deslocar pra onde estas reuniões vão ocorrer [BSB, SP, BEL]. Eles viajam muito, têm que se deslocar muito”.*  
[coordenador de negócios]

*[ent] “Como ele tem demandas de várias partes do Brasil acontecendo ao mesmo tempo, é difícil dessa pessoa [o AD] poder comparecer a todas, né? E além de comparecer, ele ainda tem que voltar pra regional dele e fazer o trabalho que foi especificado. Então, isso tem complicado muitas vezes o trabalho [dos pólos de desenvolvimento que dependem do AD]”.* [analista]

Outro problema está nos regulamentos da empresa Alpha, que não permitem que os desenvolvedores tenham permissão de escrita no modelo de dados elaborado pelos ADs, ou seja, apenas os ADs possuem autonomia para alterar o modelo de dados. As notas de campo mostram um analista afirmando que durante o desenvolvimento do SYS, o AD foi solicitado, entretanto, como ele estava distante e muito ocupado com outro projeto, só elaborou um primeiro modelo (que não atendeu às necessidades da equipe). Assim, a equipe de desenvolvimento precisou realizar alterações diretamente no código da aplicação, pois percebeu que do modelo proposto pelo AD, apenas uma tabela estava presente no código. Em outras palavras, o banco de dados utilizado na aplicação não conferia com o modelo de dados elaborado pelo AD. Diante disto, a equipe se viu obrigada a realizar uma engenharia reversa, que serviu para reconciliar as informações do modelo de dados com a base da aplicação, tal como mostra o trecho abaixo, retirado de uma entrevista realizada com o analista responsável pela engenharia reversa (analista A):

*[ent] “Como foi que eu descobri o diagrama [de classes]? Eu fiz uma [engenharia] reversa do banco. Eu não tinha o diagrama aqui pra ver. Só lá no CASE que eu fico olhando, mas é complicado de ver. Então, eu peguei uma ferramenta, fiz uma engenharia reversa, criei uma “boneca” [diagrama] pra os meninos verem, aí através dela, a gente viu como é que tava acontecendo e começou a identificar os problemas que existiam [no modelo]: por que tinha tabela duplicada? E não era só uma. E tinha outra [tabela] que levava todos os campos, menos a chave. (...) Aí a gente foi fazendo os ajustes em cima disso”.*

Parte da engenharia reversa foi acompanhada durante uma observação no local de trabalho da equipe de desenvolvimento. No momento da observação, o analista A estava criando o modelo de dados a partir do banco de dados de aplicação:

*[notes] O analista diz que o modelo deve ficar o mais parecido possível com o banco de dados de aplicação. Ele fica tanto com o banco aberto (PL/SQL Developer) – assim ele pode visualizar as tabelas e os atributos que já estão sendo utilizados pela aplicação – quanto o modelo no DBDesigner. Ele utiliza estes dois documentos porque não tem permissão de escrita nos documentos do banco de dados [só quem tem essa permissão é o AD]. O analista está tendo que fazer uma engenharia reversa: pegou o banco da aplicação e está fazendo o modelo de dados.*

Meses depois, em uma reunião de elicitação de requisitos realizada na cidade do cliente em que o AD estava presente, observou-se que, em um dado momento, o AD pede para o analista presente (que não é o analista responsável pela engenharia reversa) explicar o modelo oriundo da engenharia reversa:

*[notes] O AD aproveita pra tirar dúvidas com relação ao modelo de dados. Ele parece não ter contato com o sistema há muito tempo. Na medida em que o analista explica o modelo, ele pensa em nível de dados, fala de relacionamentos, multiplicidade de relacionamento: ‘ah, isso é um pra um?’.*

Regularmente, o AD não deveria pedir para os analistas lhe explicarem o modelo de dados, pois na teoria a tarefa de elaborar e manter este modelo de dados é de sua competência. Entretanto, quem elaborou o modelo foi o analista A juntamente com a equipe de analistas via engenharia reversa. O AD não participou dessa atividade porque na época estava muito ocupado com outros projetos, o que é acentuado pelo fato de ele estar distante do pólo de desenvolvimento. Assim, a equipe de desenvolvimento, diante da necessidade de fazer modificações no modelo e sem, no entanto, ter acesso ao modelo, foi obrigada a realizar uma engenharia reversa para recuperar o modelo de dados a partir do banco de dados utilizado na aplicação.

A distância existente entre os ADs e o pólo de desenvolvimento, somada à sobrecarga existente neste cargo, pode fazer com que todo o processo de gerência de requisitos seja impactado, como mostra o trecho abaixo, retirado de uma entrevista com um analista:

*[ent] Pesquisadora: “Quando existe uma mudança de requisitos, os clientes costumam compreender e aceitar a extensão do prazo?”*

*Analista: “Às vezes é coisa que não depende da gente aqui [a equipe de desenvolvimento]. Se mexe, por exemplo, no modelo de dados, aí pronto. Passa pra uma outra área que é no RJ. A*

*gente envia a demanda pra lá, eles também são cheios de demanda e aí tem que esperar. Não tem jeito”.*

Quando a equipe de desenvolvimento precisa que o AD altere o modelo de dados na ferramenta CASE, todo o processo é demorado:

*[ent] Analista: “(...) Eu mandei pra ele, pra ele ver o que tava lá na ferramenta CASE. E mesmo assim, depois, eu fui olhar e ele não tinha feito as mudanças”.*

*Pesquisadora: “E até hoje ele não fez?”*

*Analista: “Depois... ele fez. O líder do projeto ligou pra ele e ele fez. Mas depois a gente identificou que ficaram faltando algumas coisas, que ele não tinha feito. E aí a gente mandou pra ele, mas ele já fez. É complicada essa distância. É difícil, né?”.*

A preocupação com a demora na execução das tarefas do AD, por parte dos analistas, foi observada durante a reunião remota de elicitação de requisitos, em que o AD foi convocado. Na ocasião, o analista e o AD conversavam sobre a prioridade com que um dado conjunto de requisitos deveria ser tratado:

*[notes] O analista diz ao AD que ele [o AD] precisa considerar a prioridade dos três primeiros requisitos e explica por que ele precisa considerar essa prioridade: ‘porque o prazo da equipe de Belém depende da sua modelagem’. Embora o analista reconheça esse problema [o problema da demora] e se incomode, ele é categórico ao afirmar que ‘o projeto do cliente não pode ficar esperando a agenda do AD’.*

Alguns informantes argumentaram que os ADs não possuem conhecimento suficiente sobre o negócio e atribuíram a culpa disso ao fato de os ADs estarem distantes da equipe de desenvolvimento. O trecho a seguir foi retirado de uma entrevista com um analista e ilustra isto:

*[ent] “Olha, eu acho assim que na verdade, a questão é que ele tá muito distante da equipe de desenvolvimento. Porque, por mais que eu vá pra especificação, aí eu ouço o cara [cliente] lá falar, aí eu fico imaginando uma solução de banco de dados porque essa é a minha função. Mas se eu não acompanho sempre isso, eu vou me esquecer, né? E eu acho que é isso que acontece com ele [o AD]. Ele não tem o domínio [do negócio] e pra gente poder trabalhar dessa forma como ele tem que trabalhar [à distância], ele teria que dominar a função de cada uma das tabelas, saber como elas se relacionam e tal e ajudar o desenvolvimento da minha equipe”.*

A falta de domínio do problema, além de ter sido relatada durante as entrevistas, também foi observada durante uma reunião de elicitação de requisitos. A presença do AD foi solicitada nesta reunião. A falta de contato frequente com o negócio do cliente deixou o AD “perdido” em alguns momentos, como é possível perceber nos trechos abaixo:

*[notes] Tenho a impressão de que o AD está meio distante do projeto. Ele pergunta muitas coisas (elementares) sobre o sistema para o cliente: ‘que planilha é essa?’.*

*[notes] O AD admite que está afastado do projeto: ‘sabe que faz um tempão que eu não vejo isso?’, se referindo às regras de negócio, requisitos e modelos de dados do sistema.*

O fato de os ADs lidarem com inúmeros projetos, diferentes em sua essência, dificulta para que os mesmos se tornem profundos conhecedores do negócio de cada projeto. O trecho abaixo, de uma entrevista com um analista, ilustra isso:

*[ent] “Geralmente ele [o AD] até vai pra especificação pra saber o que é que vai acontecer de novo pra ele saber, mais ou menos, como ele vai ajustar o banco naquele momento. (...) É aquela dificuldade... Ele deveria entender tudo do negócio porque teoricamente é ele que faz o modelo. Ele tem essa função. Mas, não é a verdade. (...) Aí acaba dando mais trabalho aqui pro desenvolvimento do que era necessário porque a gente acaba tendo que ensinar tudo, explicar tudo pra ele, que é o trabalho dele e não da gente”.*

De um modo geral, os problemas identificados durante a modelagem de dados são principalmente causados pela existência de poucos ADs, o que os deixa sobrecarregados de atividades e incapazes de acompanhar todos os projetos aos quais eles são designados. Entretanto, embora não seja a principal causa, a distância acentua os problemas identificados nesta fase do processo, pois dificulta a comunicação entre os ADs e os analistas. Se estas pessoas estivessem no mesmo ambiente de trabalho, o contato presencial seria mais fácil, rápido, simples, menos dispendioso e, portanto, mais freqüente.

#### *4.1.1.3 Negociação de requisitos*

Um dos problemas que a equipe de desenvolvimento enfrenta na empresa Alpha está relacionado com o atendimento das demandas. A análise dos dados indica que o cliente modifica com freqüência o escopo da demanda (normalmente, aumentando este escopo) em momentos inoportunos:

*[ent] Pesquisadora: “Você foi lá pra uma modificação de requisitos?”*

*Analista: “Exatamente. Inclusive dessa vez quando eu fui e que você [pesquisadora] foi também. Geralmente é isso mesmo. Eles repassam e a gente muda”.*

*Pesquisadora: “Mas aquela última que nós fomos era um fechamento [validação]. Ou deveria ser um fechamento, certo?”.*

*Analista: “É, só que o cliente começa a elicitar as coisas. Então, tem várias coisas pra alterar, coisas novas que ele inventa...”*

O trecho acima mostra que durante uma reunião de fechamento de requisitos, que deveria ser o momento para validar requisitos já elicitados, o cliente está preocupado em incluir novos, aumentando o escopo da demanda.

Uma situação especial foi vivenciada durante as observações realizadas na cidade do cliente. Ocorria uma reunião de validação de requisitos quando o coordenador de negócios da empresa Alpha solicitou uma reunião entre os clientes e o pólo de desenvolvimento, via áudio-conferência. O problema em questão tratava-se de um requisito que havia sido especificado, mas que não seria desenvolvido:

*[notes] Dois dos clientes não se conformam porque a especificação foi feita, documentada, mas não será implementada. O líder e o coordenador de negócios entendem que o requisito foi adicionado a esta demanda porque se tratava de uma emergência. O impasse é que o cliente chegou a discutir com o analista presente na reunião de levantamento de requisitos que 'n' coisas poderiam não ser feitas, mas que o requisito em questão tinha que ser corrigido de todas as formas. A reunião está tensa. O coordenador deixa claro para os demais que se o requisito está especificado, ele será desenvolvido. O cliente assume o telefone: 'a demanda não pode ir pra produção se este requisito não estiver direito!'*

Em um dado momento, o líder de projeto assume que o requisito havia sido erroneamente especificado antes do tempo:

*[notes] Na áudio, o líder se justifica: 'como a demanda já tava aberta, na hora que a gente tava elaborando o documento, esse texto acabou indo junto'. Percebe-se que o requisito foi escrito, mas que não está no escopo da demanda. Todos percebem que foi erro da empresa Alpha, que a demanda veio escrita por engano. Os clientes ficam em silêncio, mas balançam a cabeça em sinal de desaprovação.*

Diante da situação, o cliente sugere que ocorra uma junção de demandas, entretanto o coordenador de negócios não aprova a idéia:

*[notes] O coordenador de negócios se posiciona: 'isso é aquele negócio... não homologa nem uma demanda, nem a outra... é um caos!'. O coordenador está visivelmente irritado porque em virtude de um erro humano, cometido pela empresa Alpha, uma demanda terá que ser inserida em outra. De acordo com o líder do projeto, foi erro humano. O item foi adicionado em um documento de fechamento de requisitos, então é como se aquele requisito estivesse previsto para ser implementado. Os clientes, por sua vez, se mostram irredutíveis e não concordam que o requisito esteja em um documento de fechamento e não seja incluído na implementação.*

Resolve-se, ao final da áudio-conferência, que os requisitos das demandas serão unidos em uma só e que o cronograma será alterado. Esta situação indica que os requisitos não são

negociados formalmente, isto é, após a etapa de elicitação, parece não haver um momento de negociação para que os *stakeholders* entrem em acordo acerca do que é necessidade para o cliente, quais são suas prioridades e se isto é factível pela equipe de desenvolvimento. Entende-se que, no caso acima, o escopo da demanda não ficou claro o suficiente para a equipe de desenvolvimento e, por isso, foi especificado no momento incorreto.

Em um processo de ER, o escopo de uma demanda deve ser fechado durante a etapa de negociação de requisitos e, uma vez fechado, é sobre este escopo que a equipe de desenvolvimento deve trabalhar. Portanto, clientes e analistas não devem adquirir o hábito de modificar o escopo aleatoriamente durante as etapas do processo de ER. Não foi possível identificar como exatamente a etapa de negociação de requisitos ocorre na empresa Alpha, porém identificou-se que a distância atrapalha a forma como o escopo das demandas é definido, o que, portanto, influenciaria diretamente na etapa de negociação de requisitos, visto que é nesta etapa que se define o escopo da demanda. Na verdade, o escopo das demandas da empresa Alpha é modificado com bastante frequência e em momentos inoportunos, como em reuniões de validação de requisitos porque os *stakeholders* não se vêm com frequência em virtude da distância. Em projetos distribuídos, os clientes tendem a solicitar modificações em requisitos de outras demandas a qualquer momento, em uma tentativa desesperada de aproveitar todo o tempo disponível para informar suas necessidades ao analista presente. Isto prejudica o andamento da reunião como um todo e cria situações indesejadas de resistência entre clientes e analistas, pois os clientes forçam acordos, os analistas dizem que só podem decidir esses aspectos ao voltarem para o pólo e os clientes se incomodam com toda a situação.

Por conta disto, a equipe de desenvolvimento vive em constante (re) negociação de prazo junto aos clientes, um dos motivos pelos quais a relação entre eles se desgasta. Situações como estas originam questionamentos do tipo: será que a etapa de negociação de requisitos está sendo tratada com a devida importância que um contexto distribuído requer? Isto será discutido no próximo capítulo, na seção 5.1.

#### *4.1.1.4 Especificação de requisitos*

A especificação dos requisitos na empresa Alpha é realizada à distância. Já se sabe que a elicitação de requisitos pode acontecer no contexto do cliente (com o deslocamento de um analista) ou no próprio pólo de desenvolvimento (com o deslocamento dos clientes). No caso de o analista ser deslocado, ele se torna responsável por repassar os requisitos elicitados junto

ao cliente. Uma vez que ocorrem o levantamento e o repasse de requisitos, a equipe de desenvolvimento começa a especificá-los, isto é, a elaborar os documentos de especificação de casos de uso. Como este é um processo demorado, ele ocorre posteriormente ao deslocamento do cliente ou do analista, isto é, quando ambos estão geograficamente distantes.

Os dados coletados mostraram que uma das reclamações dos clientes sugere a existência de documentos pouco claros. Nas observações realizadas durante o fechamento de requisitos, os clientes precisaram sugerir modificações no documento diversas vezes, ou por falta de clareza ou por incompletude do requisito. Seguem trechos das notas de campo resultantes do fechamento de requisitos:

*[notes] Os clientes estão lendo o documento e agora passam para o próximo requisito. Um dos clientes diz que se ler o texto do jeito que está não dá para entender direito. Os clientes então sugerem como o texto deveria ficar, adicionam textos em vermelho realçando-os em amarelo. (...) O cliente se queixa da forma como a regra de negócio foi escrita, então o analista explica o que o requisito quer dizer. O cliente diz: 'ah, é isso que ele [analista que escreveu o requisito] quis dizer? Então precisa melhorar, hein?'*

O fato de os documentos serem fonte de reclamação dos clientes por apresentarem requisitos incompletos ou pouco claros exige que o cliente gaste muito tempo da reunião explicando coisas que já haviam sido explicadas na reunião de levantamento de requisitos. Entretanto, perder tempo com retrabalho neste momento é complicado, pois a reunião de fechamento de requisitos inicialmente deveria ser curta e servir unicamente para lerem o documento e consertarem o mínimo de erros possível, como a seção seguinte (validação de requisitos) mostrará. Os trechos abaixo foram retirados das notas de campo escritas durante as reuniões de fechamento de requisitos:

*[notes] O cliente se queixa muito dos documentos. O documento de regras de negócio parece apresentar muitos erros. As regras de negócio estão mal escritas ou incompletas. No final das contas, o cliente está tendo que gastar muito tempo explicando regras de negócio para o analista que está presente [que não é o mesmo que veio para a reunião de levantamento de requisitos]. Por sorte, o analista entende tudo com facilidade.*

A análise dos dados sugere que o retrabalho observado nos trechos acima é causado pela distância existente entre os envolvidos no projeto. Esta distância não permite que os analistas visitem seus clientes frequentemente e conheçam seu ambiente de trabalho ou a forma como realizam suas atividades. Já se sabe que, normalmente, apenas um analista é deslocado para a reunião de levantamento de requisitos e que quando o mesmo retorna a seu pólo de

desenvolvimento, ele é responsável por repassar os requisitos elicitados para seus colegas. Deter o conhecimento em uma única pessoa pode ser fatal, pois o analista que se desloca pode entender as necessidades do cliente de uma forma que não necessariamente seja a correta:

*[notes] O cliente explica que muitas vezes os fluxos descritos nos documentos não estão consistentes com o fluxo do trabalho.*

O trecho acima, retirado de notas de campo coletadas na cidade do cliente, sugere que existe um problema de compreensão entre analistas e clientes durante a etapa de elicitação de requisitos que ocasionam a existência de requisitos cujos fluxos não estão de acordo com o que acontece de fato.

Dificuldades no entendimento de um requisito podem causar outro problema bastante grave, que ocorre posteriormente no processo de desenvolvimento: a entrega de funcionalidades incorretas ou defeituosas aos clientes. Em diversos momentos, durante a coleta de dados, constatou-se que os clientes enfrentaram este tipo de dificuldade após o sistema ser homologado e estar em produção. O trecho abaixo ilustra isso:

*[notes] O cliente aproveita o momento para falar que quando eles tentam exportar pro [Microsoft] Excel (na versão do sistema que está em produção), acontece uma série de erros. Ele diz que precisa desbloquear um monte de avisos até conseguir finalmente que o Excel mostre sua planilha. Neste momento, o analista cede sua máquina para que um dos clientes reproduza o erro. O cliente se autentica no sistema, tenta reproduzir o erro, mas agora, para surpresa de todos, nem a planilha está sendo gerada.*

A atividade de especificação de requisitos ocorre apenas no pólo de desenvolvimento, ou seja, o cliente tem pouca interação com a equipe de desenvolvimento durante esta atividade. Entretanto, à medida que os documentos ficam prontos, a equipe de desenvolvimento os envia para revisão dos clientes, momento em que mais problemas causados pela distância voltam a aparecer.

O processo de revisão de documentos sofre os efeitos naturais da distância, como mostra o trecho abaixo, retirado de uma entrevista com o cliente:

*[ent] “Eu fico olhando ali no cronograma, às vezes a gente faz a especificação em uma semana, geralmente as nossas especificações são de uma semana, a gente tem 3 semanas pra fechar a documentação, pra empresa Alpha mandar a documentação. Dependendo, a gente faz a análise da documentação em uma ou duas semanas, aí a gente leva mais uma semana até botar todo o protocolo, então olha só, foram sete semanas. Do ponto que a gente abriu lá a demanda, até a*

*hora que eles vão abrir o projeto pra efetivamente começar a fazer alguma coisa. E aí depois começa todo esse processo”.*

A revisão dos artefatos demora porque a distância causa um retardo na comunicação entre equipes geograficamente distribuídas. O trecho abaixo foi retirado de uma entrevista com um dos analistas da empresa Alpha e ilustra este problema. Na ocasião, o analista foi perguntado sobre as dificuldades que ele sentia ao lidar com clientes geograficamente distantes do pólo de desenvolvimento:

*[ent] “Eu vejo a demora das respostas. Eles mandam [o artefato de requisitos] pra gente aí a gente tem um tempo pra ver e mandar de novo. Eles têm o tempo deles pra ver e retornar. Aí se mudar [o requisito], repete todo o ciclo. Falta instantaneidade na coisa. Acho que a distância favorece isso, das coisas demorarem mais”.*

A falta de instantaneidade citada na entrevista acima pôde ser constatada durante a observação realizada na cidade do cliente. A reunião em questão tratava do levantamento de requisitos e notou-se que os clientes demoraram a enviar o documento e por isso o analista não teve tempo hábil para lê-lo:

*[notes] O documento a partir do qual a reunião se inicia é do cliente. Este documento foi enviado apenas na sexta-feira [a reunião começou na segunda-feira seguinte], por isso o analista deslocado do pólo para a reunião não pôde ler o mesmo com cuidado.*

Como o processo de revisão depende principalmente da comunicação entre os *stakeholders* (envio dos documentos por e-mail e realização de áudios para sanar dúvidas), se a comunicação é prejudicada pela distância, o processo todo de revisão é prejudicado também. Durante a revisão dos artefatos, os *stakeholders* tornam-se extremamente dependentes um do outro, o que confere um caráter assíncrono à atividade: os clientes precisam esperar até que todos os documentos sejam elaborados. Uma vez que os clientes têm acesso aos documentos, é a vez dos analistas esperarem até que a revisão finalize e eles tenham acesso às modificações sugeridas pelos clientes.

A falta de sincronia durante a especificação de requisitos, especialmente quando os artefatos estão sob a revisão dos clientes, desencadeia outro problema que é a elaboração de documentos inconsistentes. A primeira versão do documento normalmente parte da empresa Alpha, mas permite-se que os clientes alterem o documento caso os requisitos estejam incompletos ou incorretos. Isto é problemático porque o documento começa a “ir e vir” entre as partes, com as mesmas adicionando novas coisas e supondo que o outro lado está interpretando os requisitos da forma correta. Embora essa medida pareça vantajosa em

projetos distribuídos, afinal de contas evita que as partes se desloquem para se reunir, ela causa problemas sérios em longo prazo já que cria o risco de analistas e clientes trabalharem com versões diferentes do documento, tornando-o inconsistente. O trecho a seguir foi retirado de uma entrevista realizada com o cliente e ilustra esta situação:

*[ent] Cliente: “Às vezes nesse trânsito de documento, vai e volta, fica um pensando que tá trabalhando numa versão e o outro tá pensando que tá trabalhando em outra versão. Já aconteceu isso por diversas vezes no SYS. Porque eles não receberam um documento ou a gente não recebeu uma outra versão”.*

*Pesquisadora: “Vocês tiveram muitos problemas [por causa disso]?”*

*Cliente: “Tivemos. Tanto é que a demanda demorou quase dois anos. Também por desencontro de documento”.*

A demora na revisão dos documentos causa a impressão no pólo de desenvolvimento de que os clientes não se dedicam como deveriam à tarefa. Entretanto, a “falsa” falta de dedicação dos clientes talvez seja explicada pela concentração das tarefas de revisão em um único cliente. Por estarem distantes, os analistas não possuem a percepção adequada do contexto de trabalho dos clientes e por isso assumem determinadas coisas, como ilustra o trecho abaixo, retirado de uma entrevista com um analista:

*[ent] “O problema principal da dispersão é a dedicação do cliente para a atividade. Porque a gente [os analistas] tá dedicado totalmente pra isso. A gente tá alocado no projeto. O nosso dia-a-dia é esse projeto. A gente vai trabalhar nele o tempo todo. Mas o cliente, ele é escolhido lá na empresa do cliente como uma pessoa que vai passar pra empresa Alpha o que vai ter que ser feito. Então ele é o nosso cliente, está representando o nosso cliente e ele tem que entender o que o sistema tem que fazer. Só que esse cara faz ‘n’ outras coisas. Às vezes é difícil fazer com que ele se dedique mesmo à especificação desse produto. Essa é uma dificuldade. (...) As dificuldades de dedicação acabam atrapalhando um pouquinho. Demora um pouquinho mais pra sair a documentação toda de requisitos”.*

#### 4.1.1.5 Validação de requisitos

O fechamento de requisitos da empresa Alpha corresponde à validação de requisitos de um processo de ER genérico. O fechamento pode ocorrer à distância, quando o pólo de desenvolvimento envia a documentação por e-mail para que o cliente a leia e depois retorne ao pólo de desenvolvimento por telefone dizendo se concorda ou não com a documentação, ou presencialmente, quando um analista é deslocado para ler o documento junto ao cliente e então obter *feedback*. Os dados coletados mostraram problemas em ambas as situações, como discutido nos parágrafos a seguir.

Uma das pessoas mais contrárias à realização do fechamento de requisitos à distância é o coordenador de negócios. Os trechos abaixo foram retirados de observações realizadas na cidade do cliente e mostram a insatisfação do coordenador de negócios sobre o fechamento à distância:

*[notes] O coordenador afirma [de forma agressiva] que não quer fechamento de requisitos à distância. Por isso, deve haver uma nova reunião, dessa vez mais curta (de dois dias no máximo) para fechar o documento de requisitos de forma presencial. (...) provavelmente haverá uma nova reunião – de dois dias apenas – para realizarem o fechamento da documentação. O coordenador de negócios explica: “pra evitar essa coisa de fechar por telefone...”. (...) Durante a reunião, o coordenador de negócios afirma: “esse negócio da gente especificar à distância e depois mandar documento... a gente já viu que dá muito problema”.*

Um dos problemas em se realizar o fechamento de requisitos à distância reflete em outra etapa do processo de desenvolvimento (homologação), tal como mostram os trechos abaixo, retirados de uma entrevista com um analista:

*[ent] “A gente tem enfrentado bastante problema na etapa de homologação... Eu diria que uma das causas tá originada lá na questão dos requisitos. Como a gente não tinha toda essa facilidade pra ter uma segunda reunião de fechamento de requisitos com o cliente (ou seja, presencial), algumas coisas não ficavam bem claras para o cliente. Ou até pra gente mesmo, né? Pode acontecer... Então, havia desentendimentos em requisitos, muitos, nas homologações”.*

Como visto em outra seção, a elaboração de documentos pouco claros deve-se à ida de apenas um analista para levantar requisitos com os clientes. Quando este analista volta ao pólo de desenvolvimento, ele precisa repassar a seus colegas todo o conhecimento adquirido durante aquela semana. Neste momento, muita informação pode ser perdida. A equipe, então, elabora os documentos de requisitos e envia os mesmos para os clientes, via e-mail. Este processo se mostrou ineficaz. O trecho abaixo, retirado da mesma entrevista acima, explica por que:

*[ent] “Isso foi um processo que se mostrou ineficiente porque você conversar com a pessoa num primeiro momento e depois fazer o fechamento dessa documentação só através de e-mail se mostrou muito problemático porque algumas coisas não ficavam bem claras. Então, o cliente assumia que ‘bom, mas eu comentei tal coisa na reunião então eu imaginei que estivesse subentendido aqui na documentação’ e não é bem assim, né? Ou está escrito, ou não será feito. Então a gente já teve alguns problemas nesse sentido”.*

A comunicação ruim e a pouca dedicação dos clientes durante o fechamento de requisitos também foram citadas como problemas quando este fechamento ocorre à distância. Os trechos

abaixo foram retirados de entrevistas com vários analistas. Eles sugerem que os clientes, que estão responsáveis por fornecer os requisitos do sistema a ser desenvolvido, não estão alocados unicamente para essa atividade, isto é, eles também possuem suas próprias obrigações e isto talvez faça com que eles não se dediquem tanto quanto deveriam.

*[ent] “Em geral, a reunião remota não funciona muito bem, tanto pela comunicação que não é tão boa (áudio-conferência), quanto pela falta de dedicação dos clientes. A falta de dedicação dos clientes é explicada pela quantidade de coisas que eles têm lá pra fazer. Por causa disso, situações como a seguinte tendem a ocorrer: se você manda um documento e fala: ‘olha, lê e me liga pra dizer o que você achou’ – ele acaba não lendo do jeito que ele deveria ler e a documentação fica toda falha”.*

Uma consequência da pouca dedicação dos clientes é o atraso nas tarefas de requisitos, tal como mostra o trecho abaixo, também retirado de uma entrevista com um analista:

*[ent] “As dificuldades de dedicação acabam atrapalhando um pouquinho. Demora um pouquinho mais pra sair a documentação toda de requisitos”.*

Quando a empresa Alpha identificou que o fechamento à distância era problemático, passou a realizar reuniões presenciais, em que um analista era deslocado do pólo para a cidade do cliente, para ler o documento junto ao cliente e, assim, obter *feedback* sobre os requisitos elaborados. Entretanto, as primeiras reuniões logo serviram para mostrar que o tempo dedicado para esta tarefa, dois dias, não era suficiente. O trecho abaixo corresponde a notas de campo escritas durante a validação de requisitos, na cidade do cliente, e ilustram o problema:

*[notes] O coordenador de negócios diz que o fechamento aconteceria em dois dias, mas confessa que o tempo foi estimado de forma errada. É impossível fechar a especificação em dois dias. (...) Novamente, o coordenador afirma que para fechamento de documentação, dois dias não dá. No mínimo, cinco dias, pois “as nossas demandas são grandes”, ele diz.*

Além do pouco tempo dedicado à reunião, a pouca dedicação dos clientes também prejudica o fechamento de requisitos presencial (assim como prejudica várias outras etapas do processo). O trecho abaixo foi retirado de uma observação realizada na cidade do cliente, durante um fechamento de requisitos. Na ocasião, dois clientes deveriam participar da reunião para lerem junto ao analista o documento de especificação. Um deles já havia avisado previamente que não poderia estar presente. Quando o outro cliente solicita que ocorra reunião somente à tarde, o analista se mostra preocupado com o pouco tempo que restará e informa o líder do projeto, por telefone, que também se preocupa:

*[notes] O cliente sugere que as reuniões aconteçam somente à tarde, mas o analista está receoso de não dar tempo. O analista, então, consulta o líder para comunicar os problemas de agenda dos clientes. Pelo visto, o líder também não está satisfeito com a situação, entretanto nada pode ser feito.*

## **4.1.2 Problemas transversais ao processo de ER**

### *4.1.2.1 Problemas na gerência de requisitos*

Toda vez que o cliente solicita alterações nos requisitos, a equipe de desenvolvimento deve registrar o impacto desta solicitação num relatório, denominado Relatório de Análise de Impacto. Nota-se que existe uma preocupação grande por parte da coordenação de negócio em respaldar a equipe de desenvolvimento, em termos do que pode ser atendido pelo pólo de desenvolvimento ou não, como mostra o trecho abaixo:

*[ent] “Muitas vezes tudo pro cliente é muito simples: ‘ah, mas eu tava olhando aqui e isso é muito simples. Tu só precisas alterar aquela tabelinha e pronto’. Só que aquela tabela pode ser usada em várias outras funcionalidades e quando você vê, acaba gerando um impacto grandioso. Então a gente avalia o que pode atender ou não”. [coordenador de negócios]*

Não só a coordenação se preocupa em analisar o impacto causado por uma alteração de requisitos. O líder de projeto se preocupa em mostrar o impacto da modificação para o cliente. A nota de campo abaixo ilustra uma ocasião em que os clientes haviam solicitado a alteração de um determinado requisito e o líder do projeto (via áudio) mostrou-se preocupado em explicar a importância de se analisar o impacto da modificação para o cliente:

*[notes] O líder explica que o requisito é simples, mas que uma reunião de fechamento seria necessária, pois informações de um módulo são reaproveitadas em outros módulos. Por isso, a análise de impacto é muito importante [saber quais módulos são impactados, quais telas são alteradas, etc]. O líder diz que a equipe já conhece os requisitos e já os têm compreendidos. Então, a reunião de fechamento vai servir para ele [cliente] se tornar ciente dos pontos impactados pra poder fechar a implementação deste requisito.*

Os clientes, entretanto, não concordavam que uma nova reunião de especificação fosse agendada. Notou-se uma grande preocupação por parte do cliente principalmente em relação ao tempo que seria despendido no decorrer do processo:

*[notes] O cliente está bastante irritado. O cliente não entende por que precisaria de uma reunião para uma nova especificação. Ele não entende a dificuldade disso e deixa isto claro de forma muito irritada.*

O que se percebe é que a forma como a análise de impacto é disponibilizada para o cliente não é suficiente para fazê-lo entender a dimensão e a complexidade da alteração daquele requisito. O cliente parece estar desassistido, com muitas dúvidas e pouca informação. A entrevista realizada com o cliente nos mostrou que ele sente falta de um relatório mais detalhado, que mostre, por exemplo, no modelo de dados onde aquela alteração de requisitos ocorrerá, assim como quaisquer outras áreas do modelo que venham a ser afetadas:

*[ent] Cliente: “Mas isso [se referindo aos detalhes que poderiam existir no relatório de análise de impacto] eles [da equipe de desenvolvimento] não mostram. A não ser que seja uma coisa assim que chame muita atenção. Por exemplo, teve um determinado caso que não existia estrutura no banco de dados”.*

*Pesquisadora: “Tinha que criar novas tabelas?”*

*Cliente: “Exatamente! Tinha que criar aquela entidade. Aquela entidade não existia nem do ponto de vista lógico, nem físico, então tinha que criar. Então, naquele momento ali a gente precisou falar a respeito, mas isso não aparece descrito no modelo do relatório de análise de impacto”.*

A partir do momento que a equipe de desenvolvimento não detalha as alterações no relatório de análise de impacto, isto é, não mostra onde no modelo aquela alteração influenciará, quais entidades deverão ser criadas no banco, ou quais telas do sistema deverão ser modificadas, abre-se um precedente para que os clientes se sintam excluídos do projeto e tirem conclusões precipitadas. O trecho a seguir ilustra essa situação:

*[ent] “A gente tem uma ferramenta lá que é uma calculadora. A calculadora é igual em qualquer lugar e a gente diz ‘pega lá a calculadora e coloca aqui!’ e eles respondem ‘ah não, perai que a gente precisa desenhar a calculadora de novo’ – eu não consigo entender isso! E eles também não conseguem me explicar! ‘Primeiro que eu já te paguei isso’ [cliente, como se estivesse falando com o líder da equipe]”.*

Em contextos de DDS, a confiança é naturalmente abalada em decorrência da distância existente entre os envolvidos que não tiram proveito das “conversas de corredor”, isto é, conversas informais, possíveis através do contato *face-to-face*. Sendo assim, omissões como as observadas no trecho acima enfraquecem ainda mais o nível de confiança, já tão prejudicado pela distância em projetos de DDS. Projetos desta natureza precisam contar com *stakeholders* que confiam uns nos outros, consigam trabalhar de forma eficiente e possuam o objetivo comum de executar o projeto com sucesso.

#### 4.1.2.2 *Problemas na gerência de conhecimento*

Organizações envolvidas com atividades de TI lidam com enormes quantidades de informação, por isso a gerência de conhecimento se torna tão importante para empresas deste tipo. Durante a coleta de dados, entretanto, observou-se que a empresa Alpha possuía dificuldades para compartilhar conhecimento tanto em nível intraorganizacional, quanto em nível interorganizacional. Em nível intraorganizacional porque o pólo de desenvolvimento não utiliza práticas de gerência de conhecimento (GC) internamente. E em nível interorganizacional porque a distância dificulta a comunicação entre analistas e clientes e, por isso, o compartilhamento de conhecimento ocorre com mais dificuldade também.

Diante disto, a discussão apresentada nesta seção mostrará problemas comuns da disciplina de gerência de conhecimento e problemas mais específicos do desenvolvimento distribuído de software, ou seja, aqueles que são causados pela distância, e estes problemas normalmente estarão relacionados, o que significa que um causa ou é resultante do outro.

#### ***Centralização do conhecimento***

Como os clientes estão distantes da equipe de desenvolvimento e o deslocamento freqüente de analistas é um procedimento custoso para a empresa Alpha, apenas um analista é enviado para elicitar requisitos junto ao cliente. Em virtude disto, este analista retém todo o conhecimento adquirido e se torna encarregado de repassá-lo ao restante da equipe. Assim, um dos problemas causados pela centralização do conhecimento é a interrupção constante da pessoa que centraliza o conhecimento. O trecho abaixo, retirado de uma entrevista com o cliente, ilustra como o problema das interrupções é iniciado:

*[ent] “O problema é que o analista A vem [pra reunião de levantamento de requisitos]. Ela vai especificar o pacote de todo mundo. Aí ela volta [pro pólo de desenvolvimento]. (...) Você dá um pacote pro analista B e um pacote pro analista C. Acontece de o analista B estar com uma compreensão e o analista C com outra. Chega num determinado momento que aqueles pacotes vão se juntar, aí eles não casam. Aí eles [B e C] ligam. Aí entra a áudio e eu digo ‘não, faltou aqui uma peçinha de ligação’. Quem é que tem o conhecimento daquela peçinha de ligação? O analista A. É aí que a gente perde”. [cliente A]*

Neste caso, o analista que participa da reunião de levantamento é quem vai deter todo o conhecimento relacionado aos requisitos do cliente. Ele terá a difícil tarefa de voltar ao pólo de desenvolvimento e repassar aquele conhecimento aos seus colegas. Por diversos motivos (reunião curta demais, ida de apenas um analista, impossibilidade de gravar a reunião, ausência de mecanismos para gerenciar conhecimento), o registro de toda aquela informação

será insuficiente e muitos detalhes ficarão apenas na cabeça do analista que se reuniu com os clientes. Por conta disto, este analista será interrompido inúmeras vezes por seus colegas durante a etapa de especificação dos requisitos, ou seja, quando o resto da equipe estiver elaborando os documentos de requisitos.

O trecho abaixo foi retirado de notas de campo que registraram o analista A enquanto desenvolvia suas tarefas. Na ocasião, o analista A está responsável pelo modelo de dados e precisa da ajuda do analista B (que esteve presente na reunião de levantamento de requisitos):

*[notes] Ele [o analista A] vai imprimir o modelo de dados e vai mostrar pro analista B, que teve sua atividade interrompida para supervisionar o trabalho do analista A. (...) Agora, o analista B vai começar a analisar o modelo. O analista B começa a apontar atributos que faltam. Nota-se que a maior preocupação é com a ligação entre as tabelas e com a criação das chaves. O analista B tira dúvidas do analista A quanto à ligação de tabelas.*

### ***Ausência de mecanismos para gerenciar conhecimento***

Como visto, anteriormente, a distância obriga que apenas um analista seja deslocado para as reuniões de elicitação de requisitos e, involuntariamente, centralize em si todas as informações coletadas junto ao cliente. A análise dos dados mostra que não existem mecanismos para gerenciar o conhecimento na empresa Alpha, isto é, para externalizar o conhecimento adquirido por cada analista deslocado para o restante da equipe. O trecho abaixo foi retirado da entrevista com um analista e ilustra isso:

*[ent] “Eu entendo que um erro muito grande da equipe [no SYS] foi permitir que o conhecimento ficasse centrado. Então, muita coisa não está na documentação, está apenas na cabeça da pessoa: e se essa pessoa for embora? Se ela não estiver disponível? Ou simplesmente: e o dia que essa pessoa está de férias? Ninguém é insubstituível, mas quando o antigo líder saiu, o conhecimento também tava todo na cabeça dele, a gente teve muita dificuldade”.*

Não só os analistas dão indícios de que o conhecimento não é corretamente gerenciado dentro do pólo de desenvolvimento. O cliente também percebe este problema, que é acentuado pela distância existente entre clientes e analistas, o que o transforma de uma questão intraorganizacional para interorganizacional. As notas de campo abaixo foram escritas durante uma elicitação de requisitos na cidade do cliente. Na ocasião, o cliente fala sobre sua dificuldade com a falta de padronização das interfaces gráficas do sistema:

*[notes] Ele [o cliente] me conta que nunca houve problema de navegação no SYS. Os desenvolvedores sempre traziam novas idéias de navegação para os clientes, as quais eram normalmente bem aceitas. Entretanto, atualmente existe um problema de navegação muito*

*grande. O cliente associa esse problema ao fato de cada desenvolvedor ter desenvolvido um módulo sem seguir um padrão. Parece que existia um documento falando sobre esses padrões [o que os botões “voltar” e “refazer” significavam, por exemplo: o voltar significava retornar à página anterior enquanto que o “refazer” significava retornar à página contendo os parâmetros para a busca], mas eles não foram seguidos. Isso começou a acontecer porque em Julho de 2007, a empresa Alpha estava em greve e o líder contava na maioria com novatos. Em virtude disso, cada novato desenvolveu o módulo pelo qual ficou responsável de uma maneira diferente, embora existisse um documento prevendo todos os padrões. O cliente diz que os usuários finais são os maiores prejudicados.*

O trecho abaixo, de uma entrevista com o cliente, confirma a falta de padronização de interfaces e ilustra como a equipe de desenvolvimento funcionou durante a greve:

*[ent] “Às vezes a gente pede num protótipo, três botões numa mesma área. E aí, cada hora a empresa Alpha faz uma salada. Tem horas que eles criam uns botões lindos. E como a gente adora criatividade, a gente logo consome os botões. Mas por que o botão Salvar não é uma caixinha de ferramenta que toda vez e em todo lugar que ele entrar seja a mesma coisa? Eu não consigo entender por que não tem isso! E eles não conseguem me explicar por que não tem!”*

Durante a greve de meados de 2007, os analistas experientes saíram de férias e os novatos ficaram responsáveis pelo desenvolvimento do sistema. Em virtude da distância, apenas um analista é deslocado para conversar com o cliente e o conhecimento permanece centralizado neste analista. Além disso, a empresa Alpha não possui mecanismos de GC, por isso, infere-se que quando os “conhecedores dos padrões de interface” aderiram à greve, os novatos involuntariamente desenharam as telas cada um à sua maneira, entretanto não conseguimos confirmar esta inferência.

Institucionalizar um repositório para armazenar conhecimento, como uma memória organizacional, poderia ter sido uma boa estratégia para registrar a forma como as interfaces eram construídas, por exemplo, o *rationale*<sup>10</sup> associado a cada botão da tela. Organizações envolvidas com atividades de TI lidam com enormes quantidades de informação. No contexto da empresa Alpha, estas enormes quantidades de informação (conhecimento sobre os requisitos) ficam centralizadas em algumas poucas pessoas em razão do deslocamento de poucos analistas para a cidade do cliente, por isso a gerência de conhecimento se torna ainda mais importante para seu processo de ER.

### ***Entrada de novatos***

---

<sup>10</sup> Lógica, raciocínio.

Quando contratados, os analistas recebem treinamentos sobre o processo de desenvolvimento de software seguido pela empresa Alpha e sobre disciplinas específicas da área tecnológica. Entretanto, eles não são apresentados ao cliente, nem entram em contato direto e imediato com seu negócio, logo não adquirem o conhecimento necessário para começar a desenvolver funcionalidades para o sistema. O cliente se mostra preocupado e não concorda com esta situação. O trecho abaixo, retirado de uma entrevista com o cliente, ilustra a situação:

*[ent] “(...) também não adianta aquela história de ‘o analista chegou, ah vamos prepará-lo já na ferramenta!’. Toda vez que entra um analista novo eles me dizem assim ‘esse tá há uma semana em treinamento’. Eu não sei o quê que ele tá treinando porque também ninguém me diz, mas ele tá treinando pra vir trabalhar no meu negócio! Eu não sei quem ele é, eu não falei com ele, eu sei que ele tá treinando alguma coisa lá na empresa Alpha. Ele deve estar treinando o processo de desenvolvimento, ele deve estar treinando em como fazer a gente brigar entre a gente, porque eu acho que a empresa Alpha deve ter metodologia pra isso (risos) porque eles são ‘expert’ nisso, a tentar colocar palavras na nossa boca (risos). E o negócio do cliente que eles vão assumir? Eu não sei se tem espaço lá (no treinamento) pra isso. De repente até tem e eu não tô sabendo”.*

A distância impede que cada novo analista da empresa Alpha entre em contato com o cliente, pois se torna custoso arcar com o deslocamento tanto dos analistas quanto dos clientes. De fato, quando novos analistas são contratados pela empresa Alpha não existe uma metodologia oficialmente institucionalizada para receber estes novatos e para colocá-los em contato com o negócio do cliente e com o cliente propriamente dito. O que acontece é a aplicação de algumas heurísticas que supostamente ajudariam o novato a entender o negócio:

*[ent] Pesquisadora: “Quando os novatos chegam, existe alguma metodologia com eles?”*

*Analista: “Não. Metodologia propriamente dita não. (...) A gente procura eleger uma pessoa que vai ser o mentor dele (do novato) temporariamente – vai de vez em quando lá tirar uma dúvida, vai ajudar ele em alguma coisa. A gente também explica como é o sistema, como é o negócio, a pessoa [o novato] dá uma lida na documentação pra entender de modo geral como funciona. A gente vai passando os casos de uso que a pessoa tem que implementar e a gente define uma pessoa pra ser o orientador dela, pra ajudar a tirar dúvidas e tudo mais. Tem um lado positivo que permite que a pessoa passe a conhecer o negócio, mas tem um lado negativo também que é o lado de uma dependência que as pessoas criam e que não é boa pro trabalho. As pessoas passam a ver assim ‘ah eu não posso fazer nada se aquele meu colega ali que foi escolhido pra me orientar não tiver junto’. Isso é muito ruim. As pessoas acabam criando essa dependência que depois pra você tirar essa cultura dessas pessoas é um pouco mais difícil”.*

A falta de padronização nas interfaces gráficas sugere que, embora algumas heurísticas sejam aplicadas durante a entrada dos novatos, elas não parecem ser suficientes para tornar o conhecimento do negócio comum entre os novos analistas. A distância acentua este problema, pois impede que os novatos entrem em contato com o cliente e com o seu negócio, uma vez que apenas um analista (de preferência, experiente) é enviado para as reuniões junto ao cliente.

#### 4.1.2.3 Comunicação

A distância prejudica a comunicação entre os *stakeholders* durante o processo de desenvolvimento de software. Clientes e analistas não tiram proveito da comunicação informal e por isso o contato entre eles depende das raras reuniões presenciais e das áudio-conferências. Notou-se, entretanto, que as áudio-conferências são cansativas. O trecho abaixo foi retirado da entrevista realizada com o coordenador de negócios:

*[ent] “Hoje a gente tem muitos recursos, telefone, as áudios. Algumas vezes a áudio é suficiente, mas muitas vezes eu fico com meu ouvido cansado. Ela é muito cansativa. Eu passo o dia inteiro no telefone, às vezes 2, 3, 4h direto no telefone”.*

A áudio-conferência, com o próprio nome sugere, carece de recursos visuais. Durante um das observações realizadas no pólo de desenvolvimento da empresa Alpha, observou-se que a equipe de desenvolvimento, os clientes e o coordenador de negócios se reuniram via áudio para discutir as soluções propostas para um determinado requisito. Em um dado momento, sentiu-se falta de um recurso visual para localizar o cliente no fluxo do caso de uso sob discussão. Em um caso como este, as equipes poderiam realizar uma videoconferência para que o cliente visualizasse de forma mais clara e entendesse corretamente o diagrama de atividades em questão. Embora os clientes já trabalhem com a idéia da videoconferência internamente, o contato com o pólo de desenvolvimento ainda não chegou a este nível. O trecho abaixo foi retirado da entrevista com o cliente:

*[ent] Pesquisadora: “Vocês dispõem hoje de alguma tecnologia, tipo videoconferência?”*

*Cliente: “Aqui na nossa empresa nós estamos trabalhando com videoconferência desde recentemente, começou no ano passado. Eu não sei como é na empresa Alpha e também não sei como seria pra abrir isso pra um sistema, se isso já foi feito”.*

*Pesquisadora: “Mas já houve iniciativa da empresa Alpha para testar a videoconferência?”*

*Cliente: “Não. A empresa Alpha nunca propôs isso”.*

O envio de documentos também é problemático em projetos de DDS. As reclamações vêm de ambos os lados. O cliente se queixa de que os documentos enviados pelo pólo de desenvolvimento são grandes demais:

*[ent] “Então tem horas que a gente não consegue circular nem dentro do nosso correio interno de tão pesado que ele [o documento] é”.*

*[notes] Neste momento, o cliente se vira para mim e fala que a distância é muito complicada principalmente quando sua caixa de e-mail é pequena e enche rápido. Pergunto se a documentação era muito grande. O analista responde que sim – imenso – porque continha muitas imagens de protótipo.*

Já o coordenador de negócios se queixa de que o cliente manda e-mails que não chegam ao time de desenvolvimento:

*[ent] “Aconteceu uma situação que o cliente mandou o e-mail da casa dele, mas o líder de projeto disse que não recebeu. E o que ele pedia no e-mail acabou não entrando no escopo [da demanda]. Mas, na hora do fechamento do documento, ele [o cliente] pediu aquilo que estava no e-mail. Na hora, ele puxou o e-mail [enviado]. E e aí? Como é que eu fico? Ele [cliente] diz que mandou, mas será que mandou? O líder diz que não recebeu, mas será que ele recebeu e fez que não recebeu?”*

Os dados coletados também mostram que uma vez que os requisitos são levantados, ocorre um distanciamento ainda maior entre clientes e analistas. A distância colabora para que isso aconteça, pois dificulta a comunicação entre os stakeholders.

*[ent] Pesquisadora: “Em sua opinião, o que você vê de mais dificuldade em virtude de o cliente estar distante?”*

*Analista: “Comunicação mesmo. Eu acho que eles deveriam participar mais durante o desenvolvimento do projeto. Eu acho assim um pouco distante ainda. A gente especificou agora, a gente fez algumas áudio-conferências, mas a partir do momento que chegou na implementação, acabou o contato. A gente só vai entrar em contato de novo quando acabarem os testes e entrarmos na homologação. A gente passa esse período inteiro aqui na implementação sem falar com eles”.*

A falta de contato com o cliente durante o desenvolvimento é problemático porque permite que na entrega do sistema, ou do módulo, o cliente argumente que aquilo que está desenvolvido não foi o solicitado. Isto gera desconforto entre os envolvidos, diminui drasticamente o sentimento de confiança entre eles e acarreta em replanejamento do projeto e em alteração de cronograma, como mostra o trecho abaixo, retirado de uma entrevista com um analista:

[ent] [Na entrega, os clientes] *queriam coisas a mais. Diziam ‘ah, não está no documento, mas eu achei que ficou subentendido que vocês iam implementar isso’.* Esse tipo de coisa que a gente [equipe de desenvolvimento] *queria evitar. Se eles tivessem acompanhado desde o início, ‘ah, embute aquele negócio lá’, tudo bem, a gente ia colocar. Agora chegar no final e falar isso [que o que foi implementado não era o que eles queriam], aí complica, compromete. A gente tem que voltar e implementar aquilo de novo, estender mais o prazo.*

Não existir contato entre clientes e analistas durante o desenvolvimento do software pode ser resultado de outro problema, também identificado durante a coleta de dados: o cliente não acompanha o processo de desenvolvimento, pois não possui percepção (ou *awareness*) do contexto do pólo de desenvolvimento. Isto será discutido na seção a seguir.

#### 4.1.2.4 *Falta de awareness do processo de desenvolvimento de software*

A distância causa uma falta de *awareness* (percepção) do ambiente de trabalho das partes envolvidas, ou seja, os analistas não conhecem profundamente como os clientes seguem o fluxo de seu negócio, assim como os clientes desconhecem como os analistas realizam suas tarefas. De forma geral, o processo de desenvolvimento de software seguido pela equipe é desconhecido pelo cliente e até pelo coordenador de negócio, o que inclui as tarefas realizadas, a ordem em que elas são executadas (cronograma), os produtos gerados por estas tarefas, isto é, os artefatos elaborados, entre diversos outros aspectos. Esta falta de percepção contribui para que o cliente se sinta insatisfeito quando os analistas pedem um prazo muito longo:

[ent] *“Eu fico olhando ali no cronograma, às vezes a gente faz a especificação em uma semana, geralmente as nossas especificações são de uma semana, a gente tem 3 semanas pra fechar a documentação, pra empresa Alpha mandar a documentação. Dependendo, a gente faz a análise da documentação em uma ou duas semanas, aí a gente leva mais uma semana até botar todo o protocolo, então olha só, foram sete semanas. Do ponto que a gente abriu lá a demanda, até a hora que eles vão abrir o projeto pra efetivamente começar a fazer alguma coisa. E aí depois começa todo esse processo. Aí eles ficam lá construindo. Eles dizem um prazo e praticamente sobre a demanda a gente não vai se falar muito, vai se falar em apenas algum daqueles casos que eu narrei pra você – se eles acharem que tão entendendo mal ou se tem algum conflito na hora que eles começam a juntar os pacotes – mas praquilo ali a gente praticamente não vai se falar. Aí vai abrir um outro momento em que eles vão preparar toda a documentação que é quase uma semana disponibilizando. Aí vai entrar o cronograma de homologação que a gente já sabe que nunca é tempo suficiente. Eu não sei se isso é de propósito, às vezes eu tenho as minhas dúvidas. Eu não sei se a gente finge que vai dar conta e eles fingem que é plausível. Eu não sei, eu não entendo de onde vem o raciocínio. Acho que a gente perde tempo demais! Quer dizer, em 90 dias,*

*vamos supor que a gente passe 7 semanas comendo papel, num período que é pra ter 12 semanas. Será que não tem uma forma de enxugar isso?”*

As incertezas do cliente em relação ao cronograma indicam que o cliente precisa estar bem informado sobre o que acontece com a equipe de desenvolvimento após a elicitação e posterior validação de requisitos. Nota-se que o cliente sente a necessidade de entender por que a equipe de desenvolvimento demora tanto para disponibilizar a documentação e por que existe um prazo tão grande entre a elicitação de requisitos e o início do desenvolvimento do sistema. Como o processo de desenvolvimento da empresa Alpha não é conhecido pelos clientes, eles não entendem o motivo para esta demora, apesar de alguns dos clientes serem da área de TI.

Outro problema causado pela falta de percepção que clientes e analistas sentem em relação aos ambientes de trabalho uns dos outros, tornou-se claro quando o cliente se mostrou bastante descontente com o *template* utilizado para o registro dos requisitos. Seguem trechos de notas de campo coletadas na cidade do cliente:

*[notes] Noto que o documento de especificação de requisitos é muito criticado tanto por um cliente quanto pelo outro. Um dos clientes gosta de coisas mais visuais, dando a entender que o documento escrito não é de muita utilidade.*

Ainda sobre o documento de requisitos, os clientes se mostraram incomodados com o conceito de fluxos (fluxo principal e fluxos alternativos):

*[ent] “Aquela forma que você vai falando de fluxos alternativos, seqüências do fluxo de como ele funciona, etc. Aquilo ali não é didático! (...) eu acho que a gente já até incorporou um know-how de saber como aquilo funciona, mas é muito ruim”. [cliente]*

O cliente também afirmou que nunca houve uma iniciativa por parte da empresa Alpha em obter *feedback* junto ao cliente sobre o documento utilizado. O texto abaixo foi retirado da entrevista realizada com o cliente:

*[ent] Pesquisadora: “Já houve algum esforço por parte da empresa Alpha em buscar as dificuldades dos clientes com o documento?”*

*Cliente: “Não. A empresa Alpha diz a mesma coisa o tempo todo: a documentação da empresa Alpha não vai mudar. O modelo é aquele e não muda”.*

Já o trecho a seguir foi retirado de uma das observações realizadas no contexto remoto. Na ocasião, o cliente se queixou que a empresa Alpha nunca explicou o que eram o fluxo principal e os fluxos alternativos referenciados no documento de especificação de requisitos.

O “pezinho” e o “azinho” citados no texto a seguir representam, no documento, os passos do fluxo principal e do fluxo alternativo, respectivamente.

*[notes] O cliente fala que nunca ninguém da empresa Alpha explicou para o pessoal (cliente) o que era um “pezinho” e um “azinho” desses contidos no documento de especificação de requisitos.*

Outra reclamação do cliente, ainda sobre o documento, diz respeito à forma como as regras de negócio são escritas e organizadas:

*[ent] Pesquisadora: “Se você pudesse, o que mudaria nele?”*

*Cliente: “Primeiro, eu buscaria uma forma mais clara de escrever regras de negócio. A regra de negócio é txt puro. Uma regra atrás da outra. E não tem nem ordem de onde essas regras se encaixam. E é regra de banco de dados, regra da aplicação, tudo junto. Eu acho que todo usuário e gestor de sistemas devem conhecer as regras do seu sistema, mas eu acho que eles precisavam ter organizado. Primeiro, por exemplo, definir quais são as regras de banco? Então, uma parte do documento ia escrito “Regras de Banco”. Aí a outra parte “Regras do Aplicativo”. E aí elas (regras) terem ordem porque na hora que você revisar qualquer módulo, você tá lembrando as regras que estão ligadas umas com as outras. Então, tem muita coisa hoje na documentação do SYS que é desatualizada por conta disso. Digamos assim, não existe o casamento da interface com a regra”.*

*Pesquisadora: “Como se mudassem as regras, mas não atualizassem as regras?”*

*Cliente: “Exatamente! Uma coisa que os meninos [analistas] fizeram e que já melhorou bastante, é que ali eles enumeram as regras que estão vinculadas àquele caso de uso. Então, melhorou um pouquinho, mas você ainda corre o risco de esquecer algumas porque lá na origem elas estão embaralhadas”.*

Nota-se que o próprio cliente se preocupa em manter uma rastreabilidade entre as regras de negócio e as interfaces. A seção 4.2.1.1 mostrará que uma das boas práticas da empresa Alpha foi adotar protótipos durante as reuniões de levantamento de requisitos e relacionar estes protótipos com as regras de negócio. O texto abaixo foi retirado de uma entrevista com o cliente:

*[ent] “Então, tem muita coisa hoje na documentação do SYS que é desatualizada por conta disso. Digamos assim, não existe o casamento da interface com a regra. Como se mudassem as regras, mas não atualizassem as regras? Exatamente!”*

Outro problema citado durante as entrevistas (tanto com o pólo de desenvolvimento quanto com os clientes), e constatado durante as observações realizadas na cidade do cliente, foi a dificuldade que os clientes têm em acompanhar o processo de desenvolvimento seguido

pelo pólo de desenvolvimento. Esta dificuldade é um dos impactos causados pela distância existente entre clientes e equipe de desenvolvimento. Como os clientes não possuem *awareness*, isto é, percepção do ambiente de trabalho da empresa Alpha, conseqüentemente, ele não têm conhecimento das tarefas realizadas pelo pólo de desenvolvimento. O trecho abaixo foi retirado de uma entrevista com um analista e sugere, a partir de uma reflexão do próprio analista, que o cliente não acompanha as tarefas da equipe. Na ocasião, o analista discorre sobre a etapa de especificação de requisitos que havia sido “menos demorada”:

*[ent] “O cliente até se espantou com isso porque ele esperava que a gente fosse demorar mais pra escrever todos os casos de uso. Talvez ele não tivesse a noção de que a gente já tava andando com isso, né?”*

De fato, o problema sugerido no trecho acima se confirmou, como mostra o trecho abaixo, retirado de uma entrevista com o cliente:

*[ent] “Eles [no pólo de desenvolvimento] dizem um prazo e praticamente sobre a demanda a gente não vai se falar muito, vai se falar em apenas algum daqueles casos que eu narrei pra você – se eles acharem que tão entendendo mal ou se tem algum conflito na hora que eles começam a juntar os pacotes – mas ‘praquilo’ ali a gente praticamente não vai se falar”.*

A situação do cliente (falta de informações sobre o andamento do projeto) é, aparentemente, agravada pela disponibilização de cronogramas “macro”, pouco detalhados, aos clientes justamente para que o pólo de desenvolvimento possa lidar com possíveis atrasos de forma transparente para o cliente:

*[ent] “Na assinatura da validação já é dado um cronograma macro pra eles [os clientes]. Bem superficial ‘ah, a implementação vai terminar lá em tal período, a homologação vai ser em tal período, a implantação...’, bem macro. O cronograma é macro para que o líder possa trabalhar ali dentro (lidar com possíveis atrasos, etc).”*

Não só o cliente enfrenta problemas na hora de acompanhar o processo de desenvolvimento de software em um projeto. Os dados coletados mostram que o coordenador de negócio também não consegue acompanhar o processo de forma “micro”, ou seja, etapa por etapa, como mostram os trechos abaixo, retirados de uma entrevista com o coordenador de negócio:

*[ent] “Eu tenho os cronogramas. Na hora que a gente fecha a validação de requisitos, eu tenho os cronogramas todos. Eles [pólo de desenvolvimento] têm lá com eles o cronograma de todas as tarefas, tarefa por tarefa, mas eu vou te ser sincera, não tenho como acompanhar 30 projetos nesse nível de detalhamento. Eu tenho aqui minha agendinha”* – [neste momento, ele mostra um calendário impresso. Em cada dia, ele coloca em que tarefa qual equipe está. O cronograma é em

alto nível, por exemplo: o desenvolvimento acaba no dia tal, então os testes começam no dia tal] – “coloco todos os meus compromissos e fico controlando. Eu prefiro fazer as minhas coisas assim. (...) Como tudo funciona à distância, eu tenho que ter os cronogramas todinhos. Eu não tenho assim um sistema de ‘alerta de cronograma’. O controle fica muito na minha mão. Às vezes, uma ou outra nota importante passa despercebida, então eu tenho que ter uma agenda, um controle muito grande disso. Não tenho uma ferramenta eletrônica que me facilite isso. Não tenho. (...) A gente não tem gestão do que está sendo feito do lado de lá [no pólo de desenvolvimento]. Eu não tenho condições de saber. Se ele [o gerente do projeto] me disser que vão ser 10 homens-dias, eu vou ter que acreditar, mas eu não sei se são 10 homens-dias. Eu não tenho uma ferramenta que diga que a medida está certa, então o cliente desconfia e questiona por que são 10 homens-dias. Eu não sei por que são 10 homens-dias!”

Finalmente, tanto a empresa Alpha quanto seu cliente dispõe de tecnologias para compartilhar os artefatos produzidos durante o projeto. Existem dois sistemas (um no pólo de desenvolvimento e outro no sítio do cliente) que deveriam agir como espelho um do outro, disponibilizando os artefatos produzidos pelos analistas para os clientes consultarem sempre que necessário. Entretanto, o que se concluiu a partir das entrevistas e observações é que o sistema do cliente nem sempre contém a última versão dos artefatos.

[ent] “Não sei se da parte deles [clientes], todos acessam e como isso é feito, mas vira e mexe, eles querem e dá problema que eles não tão conseguindo acessar. Eu não sei se eles acessam facilmente, se o acesso é amigável, se eles estão efetivamente fazendo isso, se isso funciona, não sei”. [coordenador de negócios]

O trecho acima mostra que, além do repositório não funcionar bem, o coordenador não está ciente do contexto de trabalho dos clientes. Embora o coordenador não esteja ciente sobre a realidade dos clientes, o trecho seguinte, retirado de notas de campos escritas sobre o contexto dos clientes, nos permite assegurar que o sistema de fato não funciona. Na ocasião, um dos clientes está falando sobre o sistema que deveria armazenar os artefatos no lado do cliente e que, no entanto, não funciona adequadamente:

[notes] Agora ele [o cliente] fala sobre o sistema que fica do lado do cliente. Neste site, a empresa Alpha deveria ser responsável por disponibilizar os documentos aos seus clientes. Atualmente, os documentos existem, mas estão desatualizados. O cliente fala inclusive que os documentos que estão lá datam de 4 anos atrás.

## 4.2 Pontos positivos

Os dados coletados durante este estudo etnográfico identificaram não só aspectos problemáticos, mas também pontos positivos que trazem benefícios à equipe de

desenvolvimento e aos clientes da empresa Alpha. Estes pontos foram subdivididos em três categorias: soluções metodológicas, soluções organizacionais e soluções tecnológicas. No primeiro caso, serão discutidas todas as metodologias utilizadas durante o processo de desenvolvimento adotado pela empresa Alpha. No segundo caso, serão discutidas as medidas tomadas em nível organizacional pela empresa. E, finalmente, no terceiro caso os recursos tecnológicos utilizados pela organização serão apresentados.

#### **4.2.1 Soluções metodológicas**

Os dados coletados mostraram que tanto os analistas de sistema quanto os clientes da empresa Alpha seguem determinadas práticas, ainda que as mesmas não estejam institucionalizadas formalmente no processo da empresa. Notou-se que o principal intuito dos envolvidos era o de amenizar os impactos causados pela distância. Decidiu-se chamar de “soluções metodológicas” para todas aquelas que envolvessem atividades da engenharia de software, por exemplo, adotar boas práticas de especificação de requisitos, de gerência de conhecimento e de testes.

##### *4.2.1.1 Melhorar a qualidade da documentação*

Uma das preocupações identificadas neste estudo foi com a qualidade da documentação. A seção de pontos negativos mostrou que a distância causa problemas durante a elicitação de requisitos, a qual pode gerar requisitos incorretos que futuramente se transformarão em funcionalidades incorretas. A análise dos dados identificou que clientes e analistas buscam melhorar o processo de documentação de requisitos. A nota de campo abaixo foi escrita durante uma elicitação de requisitos, na cidade do cliente. Na ocasião, o cliente e o analista de negócios sugerem melhorias no processo de escrita do documento:

*[notes] O cliente mostra que é muito importante documentar de forma objetiva, mas detalhada: “Pode ser a redundância da redundância”, ele diz. Ele também aprendeu que quando encontra um termo técnico, deve manda trocar para um termo “do Português”. O coordenador de negócios acha que o uso do “ou seja” é muito importante porque explica mais de uma vez uma mesma coisa.*

A empresa Alpha também conta com o auxílio direto de uma equipe da área de requisitos. O papel desta equipe é revisar os requisitos antes de eles serem enviados para os clientes. O trecho abaixo foi retirado de uma entrevista com um analista e explica como esta cooperação funciona:

[ent] Analista: “A gente tem o apoio da área de requisitos daqui também. (As pessoas desta equipe aconselham) ‘olha, vamos preencher o caso de uso dessa forma aqui porque eu acho que fica mais claro pro cliente entender e pros desenvolvedores’. A gente tem essa consciência e faz daquela forma”.

Pesquisadora: “Que área de requisitos é essa?”

Analista: “É uma área responsável pela macroatividade de requisitos. O líder da equipe dá o apoio. Qualquer projeto de software que tenha aqui que passe por requisitos, antes de passar pro cliente, ele vai lá com a equipe dele e revisa. Qualquer um que for pegar um caso de uso, é pra entender não é? Pra chegar no final e ver o que é que se tem que fazer. Ele vai lá e diz ‘olha, não entendi essa parte aqui, o que é pra fazer?’, ‘procura melhorar antes de enviar pro cliente’. E isso tem ajudado bastante aqui. Ajudou muito mesmo. Apesar do processo dizer como é pra preencher, mas cada um preenche da sua forma. Não tem como. Eu coloco aqui verbo no infinitivo, ele vai lá e coloca de outra forma. [A equipe de requisitos sugere] ‘Vamos padronizar isso aqui?’. Ele chega e padroniza mesmo, ‘olha, fulano preencheu dessa forma. E aí? Faz todo mundo como ele fez ou faz aqui como está?’. É mais ou menos assim”.

Finalmente, tanto a equipe de desenvolvimento quanto os clientes estão conscientes de que deve existir uma preocupação especial com a elaboração da documentação para que o processo de revisão de requisitos, tão demorado em virtude da distância, não seja ainda mais demorado. Os dados coletados mostraram que cada lado cria suas próprias estratégias para conferir qualidade aos documentos elaborados.

No caso dos analistas, eles passaram a criar protótipos junto aos clientes para melhorar o entendimento do negócio. O trecho abaixo foi retirado de uma entrevista com o coordenador de negócios e ilustra esta prática:

[ent] “Então a gente adotou uma medida que com ela o documento é bem especificadinho, a gente usa protótipo, bem detalhado”.

Os protótipos foram propostos pela própria equipe com o intuito de enriquecer a documentação.

Um aspecto interessante observado é a questão da “cultura da documentação”. Os clientes não possuíam essa cultura, ou seja, o costume de se compreender e aceitar a importância da documentação não estava disseminado adequadamente entre os clientes, principalmente porque os clientes não possuem percepção da forma de trabalho do pólo de desenvolvimento, aspecto já discutido na seção 4.1.1.4 de problemas. Além disso, os clientes eram acostumados a solicitar modificações de requisitos sem seguir um processo organizado, tal como mostra o trecho abaixo, retirado de uma entrevista com um analista:

*[ent] “Acho que a coisa mais produtiva desse projeto foi criar neles a ‘cultura de documentação’ porque eles viram como um descuido deles em olhar a documentação causa muitos problemas. Se eles tivessem lido toda a documentação que foi feita, eles veriam que tava alguma coisa fora de contexto. (...) Antes as coisas eram pedidas por e-mail, reunião e não ficava muito explícito lá na demanda. E também não se documentava nada. Agora não. Agora por mais que venha uma situação nova, é tratado tudo na documentação, então não fica nada de fora”.*

Nota-se que, apesar das dificuldades encontradas no decorrer do processo, os analistas se empenham em seguir a documentação elaborada, ou seja, implementar apenas aquilo que está registrado na documentação. Além disso, procuram mostrar ao cliente a importância de manter todos os requisitos bem documentados, como mostra o trecho abaixo, retirado de uma entrevista com um líder de projeto:

*[ent] “A gente ainda está numa fase de amadurecimento dessa prática, de realmente fazer com que o cliente pegue a documentação, leia passo a passo e use o sistema. Geralmente o cliente não gosta de pegar o caso de uso, ler o passo e executar, ler o outro passo e executar. Eles são meio resistentes com relação a isso. Eles já têm uma idéia pré-concebida na mente deles de como eles querem o sistema, então eles já querem ir direto testando. Então, a gente tem tido um pouquinho de dificuldade de fazer realmente o cliente validar passo a passo ali do que está no caso de uso. O caso de uso muitas vezes acaba sendo usado só como um documento pra resolver discussões. O cliente diz: ‘bem aqui eu gostaria que esse botão fizesse isso’. E nós podemos dizer: ‘no caso de uso não está especificado isso, por isso nós não fizemos’. Então, é algo que a gente está tentando melhorar. Passar pro cliente a importância dele ler a documentação e acompanhar essa documentação na homologação e fazer a validação daquilo ali realmente. E não apenas ficar na idéia do que ele tem em mente, que alguma coisa pode passar despercebida. Então isso é uma prática que a gente tá tentando melhorar ainda”.*

O empenho dos analistas parece estar surtindo efeito. Os dados coletados mostraram que os clientes estavam conscientes da importância da documentação. No trecho abaixo, nota-se que muitas vezes o próprio cliente tem a iniciativa de adicionar detalhes a um determinado requisito, o que nos permite inferir que os clientes estão de fato preocupados com a qualidade da especificação de requisitos elaborada, com o que será liberado pros desenvolvedores codificarem.

*[ent] Cliente: “Então por isso que às vezes na regra de negócio eu quero botar um exemplo lá e eles [analistas] não deixam. E eu digo ‘ou vocês deixam o meu exemplo aqui, ou eu não vou atestar o anexo [homologar]’. Se eu não coloco um exemplo, não vai ficar claro que a gente entendeu a mesma coisa”.*

*Pesquisadora: “Então você é bastante preocupado em deixar tudo bem documentado?”*

*Cliente: “Sim, porque nós já perdemos muito tempo com isso. (...) A gente já chegou à conclusão. Já faz uns três anos, antigamente a gente passava muito rápido pela documentação. E hoje a gente sabe que isso não é bom principalmente depois do que aconteceu na saída de um analista do pólo [este analista era um ponto chave da equipe, pois conhecia bastante do sistema] e todas essas situações que ficaram pendentes de documentação”.*

Além das entrevistas, as notas de campo também indicam que o cliente está empenhado em aprimorar os documentos de requisitos. Durante as reuniões de elicitação de requisitos, as regras de negócio eram registradas nos protótipos à medida que eram levantadas junto ao cliente. Como a reunião era apoiada pela utilização de um *datashow*, tornava-se possível que o cliente acompanhasse tudo que o analista registrava, permitindo, portanto, que o analista obtivesse feedback do cliente em tempo real:

*[notes] O analista começa a descrever as regras de negócio na própria interface que está criando. O cliente auxilia no decorrer do processo, ditando as regras enquanto o analista as registra. O cliente aprova a forma como as regras são registradas: “tá, tá ótimo” – o analista está entendendo as regras. As regras de negócio são enriquecidas com exemplos dados pelo próprio cliente.*

O uso do *datashow* se mostrou vantajoso durante as reuniões de elicitação e validação de requisitos, pois o analista deslocado e os clientes tinham acesso ao mesmo documento, de maneira uniforme, e podiam discutir sobre o mesmo sem correr o risco de estarem falando sobre coisas distintas. Isso tudo de forma simples e rápida, tornando a reunião dinâmica. Por exemplo, nas reuniões de elicitação de requisitos, enquanto o analista construía o protótipo e associava as regras de negócio às telas (ou aos componentes de interfaces, como botões, tabelas, etc), os clientes podiam acompanhar em tempo real e, no caso de algum equívoco, podiam identificar prontamente e solicitar facilmente a correção imediata da regra para o analista. Da mesma forma, nas reuniões de validação de requisitos, como o documento era lido junto aos clientes, com o auxílio do *datashow*, os clientes eram capazes de identificar rapidamente erros e sugerir correções.

A nota de campo abaixo foi escrita durante uma reunião de elicitação de requisitos, em que o analista foi deslocado de seu pólo de desenvolvimento para a cidade do cliente. Na ocasião, os clientes estão lendo o documento de regras de negócio junto com o analista:

*[notes] A leitura conjunta do documento de regras de negócio parece ser eficiente porque na medida em que eles [clientes e analista] lêem o documento, já tiram dúvidas sobre como a interface deve ficar. O cliente corrige coisas erradas no documento e o analista toma nota das modificações.*

Nas reuniões de validação de requisitos, a leitura conjunta do documento também foi importante, pois muitos trechos do documento de requisitos não estavam bem compreendidos pelos clientes. Entretanto, como todos os envolvidos na reunião liam conjuntamente o documento com a ajuda do *datashow*, eram capazes de identificar o problema imediatamente, sugerir modificações e realizá-las, como mostram as notas de campo abaixo escritas durante uma reunião de validação de requisitos:

*[notes] A análise dos casos de uso consiste da leitura do documento. O cliente lê o documento e o analista adiciona explicações. Ele [cliente] tem sua própria cópia impressa, mas o documento também está aberto e exposto via datashow.*

*[notes] Vários itens não foram entendidos pelos clientes. Por isso, o analista está agora relendo o texto e anotando todas as alterações e inserções de requisitos. Além de anotar as alterações e as inserções, ele também está alterando o próprio documento para deixá-lo mais claro.*

Outra prática adotada pela empresa Alpha para aprimorar a qualidade da documentação é a diminuição do escopo das demandas. O trecho abaixo foi retirado da entrevista com o coordenador de negócios:

*[ent] “Nós tínhamos demandas do SYS que demoravam 6 meses, 8 meses, 1 ano pra serem atendidas pela empresa Alpha. As atividades não se mantêm intactas 6 meses, 8 meses, 1 ano. Então por isso hoje nós concordamos plenamente com o pólo de desenvolvimento de que o escopo tem que ser reduzido para no máximo a gente trabalhar com demandas em torno de 90 dias. Isso pra mim hoje é mais claro do que qualquer outra coisa. Essa redução de escopo hoje eu entendo plenamente porque que ela é tão importante”.*

Quando módulos muito grandes e complexos passam a ser divididos em demandas pequenas, os analistas têm mais chances de entender melhor o negócio junto ao cliente e elaborar requisitos mais coerentes, ou seja, melhorar a qualidade da documentação. Isto é possível porque, com demandas reduzidas, as reuniões remotas deixam de ser curtas e possuir uma enorme quantidade de assuntos a serem tratados e passam a ser mais bem aproveitadas pelos *stakeholders*. O resultado esperado é a construção de funcionalidades cada vez mais condizentes com as necessidades de seus clientes.

#### 4.2.1.2 Registrar reuniões

A distância pode dificultar que informações pertinentes a um projeto estejam disponíveis a todas as pessoas envolvidas, sejam elas clientes ou analistas. Um dos pontos positivos identificados por este estudo procura amenizar esta dificuldade: notou-se que a equipe de desenvolvimento está preocupada em registrar o máximo possível das reuniões, sejam elas

realizadas via áudio-conferência, ou remotamente, e disponibilizar estas informações a todos os participantes do projeto.

O trecho abaixo mostra o ponto de vista de um líder de projeto sobre o registro de uma áudio-conferência:

*[ent] “Sempre temos o cuidado de todo mundo saber qualquer coisa que seja tratada. Aí a gente faz ata de todas as áudios e manda pra todo mundo revisar e aprovar. E se for só uma ligação [diferente de áudio, pois só duas pessoas conversam], sai em nota. A nota é como uma ata, bem formal, ela tem todos os pontos discutidos e também vai pra todo mundo [as pessoas que participaram da ligação]”.*

Já o trecho a seguir, retirado de uma entrevista, mostra como o coordenador de negócios evita problemas quando os clientes tentam entrar em contato direto com os analistas:

*[ent] “Eu digo pra eles [analistas]: ‘me botem na linha de frente porque esse cliente vai discutir comigo e ele não vai poder dizer que disse nada pra vocês porque está tudo no meu caderno’ [nesse momento, ele abre o caderno e me mostra suas anotações sobre todas as áudios realizadas]. Eu faço questão de registrar tudo (...) Mesmo que a áudio seja técnica, eu participo. Por mais que eu não participe do início ao fim, mas no fechamento da áudio eu estou presente para registrar tudo que ficou decidido”.*

Os trechos que seguem, diferentemente dos anteriores que se referem às áudio-conferências, estão relacionados com as reuniões remotas e mostram como o registro destas reuniões ocorre. Abaixo segue uma parte de uma entrevista realizada com um analista:

*[ent] Analista: “Geralmente eu faço um sumário das minhas alterações. Todos os pontos que foram mexidos, eu faço um sumário pra depois remeter rapidamente. E antes mesmo de escrever, eu anoto as observações deles, coisas que não vão entrar na documentação, mas que eles tão falando lá. Eu tento registrar ao máximo o que eles tão falando pra não perder”.*

*Pesquisadora: “Mesmo que o cliente esteja falando de um ponto que não tenha a ver com essa demanda...?”*

*Analista: “Ah, eu anoto também. Pra não perder nada. Mesmo que não seja tão importante”.*

*Pesquisadora: “O que tu fazes com essas anotações?”*

*Analista: “Eu guardo. Algum dia eu vou usar. Tá tudo guardado aqui”.*

Coincidentemente, houve a oportunidade de se observar o analista acima durante uma reunião de validação de requisitos e, tal como afirmado durante a reunião, o analista realmente se preocupa em registrar ao máximo a reunião:

*[notes] O analista procura documentar tudo de forma bastante detalhada. Em todas as modificações realizadas no documento de regras de negócio, ele adota um padrão de cores para marcar o que foi retirado, o que foi adicionado e o que foi sugerido pelo cliente.*

O trecho abaixo mostra o ponto de vista de um líder de projeto sobre a boa prática de se registrar o máximo possível de uma reunião:

*[ent] “Uma coisa que eu acho importantíssima é a gente já sair [da reunião de levantamento de requisitos] com uma ata ou o rascunho da ata, principalmente em termos de cronogramas macro. Se eles [clientes] ficaram de tirar alguma dúvida [entre eles], já fica registrado lá na ata, ou alguma coisa que depende ou da empresa Alpha enviar pra eles ou deles enviar pra empresa Alpha. São os compromissos. Tem que ter muito cuidado com relação a compromissos. Do mais simples ao de risco mais alto”.*

Todas as boas práticas apresentadas anteriormente são realizadas com o intuito de “destacar” o máximo de informação possível dada pelos clientes nas reuniões presenciais e disponibilizá-la a todos os envolvidos no processo de desenvolvimento de software. Esta prática permite que os *stakeholders* possuam um entendimento comum sobre os requisitos e suaviza os problemas identificados na qualidade dos documentos construídos durante a especificação de requisitos, como vimos na seção de pontos negativos.

#### 4.2.1.3 Economizar tempo durante as reuniões

Os dados mostraram que a equipe de desenvolvimento segue determinadas estratégias para economizar o tempo gasto nas reuniões, uma vez que tais encontros são onerosos e, portanto, raros. Uma delas está relacionada com a organização das reuniões presenciais. Todos os trechos abaixo compõem notas de campo registradas na cidade do cliente, durante um levantamento de requisitos.

O trecho a seguir fala sobre a importância de o cliente ir para a reunião com os tópicos fechados, o que evita que se perca tempo desnecessariamente divagando sobre pontos que deveriam ser discutidos entre os clientes, sem a presença do analista:

*[notes] Hoje é o último dia de reuniões na cidade do cliente. Eu e o analista conversamos sobre os pontos positivos da reunião. O analista explica que foi muito importante que o cliente já viesse com os tópicos da reunião fechados. Assim, não se repetem coisas que já se repetiram em outras reuniões como, por exemplo, os clientes chegarem com os requisitos imaturos e começarem a discutir na frente do analista de requisitos uma série de dúvidas que ainda têm sobre o próprio negócio. Isto, felizmente, não aconteceu durante esta semana.*

Durante a reunião de elicitação de requisitos, o analista se preocupou principalmente em poupar tempo. Uma das estratégias utilizadas é manter o foco da conversa, isto é, novamente, não permitir que o cliente divague sobre assuntos que não dizem respeito à reunião propriamente dita:

*[notes] O analista consegue interceder de forma adequada: sempre que o cliente começa a mudar o foco da conversa [por exemplo, estavam conversando sobre uma determinada funcionalidade do SYS e o cliente começou a falar de outra], o analista é capaz de parar e dizer “calma Cliente X, vamos voltar ao que estávamos falando? Que aí eu não me perco...”. O analista me diz que não tem má vontade, ou seja, não é que ele não queira falar sobre a outra funcionalidade, ele pode falar tranquilamente desde que seja em outro momento.*

Além de manter o foco da reunião, observou-se que o analista buscou poupar tempo na construção do protótipo. Uma das tarefas do analista durante a reunião de elicitação de requisitos era construir um protótipo para coletar as modificações solicitadas pelos clientes tanto em requisitos quanto em regras de negócio. Para economizar tempo, o analista decidiu primeiramente conversar com o cliente para só então desenhar o protótipo. A nota de campo a seguir explica o raciocínio que o levou a tomar esta decisão:

*[notes] O analista me diz que não fez os protótipos no primeiro dia de reunião porque queria que o cliente falasse. Ele acha que se dedicasse o tempo para conversar com cliente e fazer o protótipo ao mesmo tempo, a conversa não fluiria bem. De fato, no primeiro e no segundo dias, o cliente falou bastante, enquanto que o analista só tomava notas. No terceiro dia, o analista chegou cedo à sede da empresa Alpha [onde a reunião estava acontecendo], construiu as telas com base em suas anotações e, quando o cliente chegou, a conversa já fluiu bastante: o cliente já falou bem menos no sentido de dar novas informações. Na verdade, o cliente deu bastante palpite nas telas [onde colocar um botão ou quando um determinado campo era obrigatório, etc], enquanto que o analista gastou bastante tempo nas telas. O analista finaliza dizendo que foi “uma boa tática”, que poderia não ter dado certo, mas que felizmente funcionou [lembrando que a recomendação do líder, antes de o analista viajar, era que ele construísse protótipos no FrontPage junto ao cliente desde o primeiro dia de reunião].*

#### 4.2.1.4 Realizar a elicitação de requisitos no pólo de desenvolvimento

Outra boa prática coletada durante o estudo etnográfico realizado na empresa Alpha consistiu no deslocamento dos clientes de suas cidades para a reunião de elicitação de requisitos no pólo de desenvolvimento. Apesar de a situação não ter sido presenciada, pôde-se coletar informações sobre esta estratégia através de um analista, que testemunhou a vinda dos clientes para a reunião de elicitação de requisitos no pólo de desenvolvimento:

[ent] “Foi bacana, legal mesmo! Porque quando eles vêm pra cá, todos os desenvolvedores participam da reunião [de elicitação]. Fica bem mais fácil de entender. Se eles estão falando de algum requisito de uma parte que o desenvolvedor X conhece mais, ele [o desenvolvedor X] já levanta e fala ‘Ei, peraí... E aqui? Vocês querem de outra forma?’. Quer dizer, indo só uma pessoa, fica mais difícil. Até mesmo de lembrar e repassar as coisas. Não dá pra lembrar de tudo. Presencial é muito melhor”.

Nota-se que é importante reunir o maior número de pessoas para, literalmente, ouvir o cliente e fazer anotações, o que reforça o problema já discutido na seção de pontos negativos causado pela ida de apenas um analista (ver seção 4.1.1.1). Os clientes também reconhecem que preferem ir até o pólo quando os requisitos a serem coletados são complexos demais:

[ent] Pesquisadora: “O que você prefere: que o analista venha aqui ou você vá até lá?”

Cliente: “Depende, se for uma funcionalidade muito complexa, nós preferimos ir lá. (...) A vantagem de nós [clientes] irmos é o seguinte: todo o grupo está junto, tanto a empresa Alpha quanto os clientes, no mesmo ambiente. Nós estamos apartados da nossa atividade, então nós estamos com foco só naquele desenvolvimento. Isso é um ganho fabuloso. Nós discutimos ali os modelos, podemos desenhar conjuntamente, explicamos às pessoas que vão cuidar efetivamente daquele pacote de trabalho. Então ali, eles já sabem que o “analista A” vai cuidar daquele determinado pacote. Então ele já vem com as idéias dele e a gente já fala ‘mas olha, isso não é assim, é assado, muda aqui, vai ali’. (...) Quando todo mundo está junto [clientes e desenvolvedores], tem essa dinâmica, essa compreensão coletiva”.

Finalmente, as notas de campo reforçam o que havia sido dito nas entrevistas: o contato entre clientes e analistas traz, de fato, ganhos consideráveis ao projeto. Seguem notas de campo oriundas das reuniões de elicitação de requisitos na cidade do cliente, com um analista deslocado do pólo de desenvolvimento. Na ocasião, o cliente está solicitando modificações em requisitos de um módulo no qual o analista já trabalhou anteriormente:

[notes] Neste momento, o cliente está manuseando um documento sobre as demandas. Ele fala sobre um requisito que no passado, por coincidência, ficou sob a responsabilidade do analista que está presente. A conversa flui naturalmente e o analista não mostra dúvidas. Isto sugere que o ideal é que o cliente se encontre com a equipe de programadores. Quando determinados requisitos com os quais o analista não está familiarizado são mencionados, ele opina muito pouco em relação aos requisitos diretamente envolvidos com ele.

O trecho mostra que como o analista presente havia participado da especificação do requisito, a conversa transcorreu tranquilamente, ao contrário de discutir sobre um requisito desconhecido pelo analista, o que foi mais complicado. Isto indica que se os analistas entram em contato com os clientes ainda no levantamento de requisitos, as chances de os analistas

adquirirem um conhecimento comum, mais correto e mais completo sobre esses requisitos aumentam consideravelmente. Além disso, quando uma determinada necessidade do cliente exigir uma solução um pouco mais elaborada, vários analistas estarão presentes para opinar e situações como a seguinte serão evitadas:

*[notes] [o cliente e o analista estão discutindo sobre um relatório que está grande demais]: o cliente começa a dar soluções para o tal relatório, como retirar colunas, por exemplo. Nota-se que se o restante da equipe estivesse presente, eles poderiam ajudar dando mais soluções.*

#### 4.2.1.5 Realizar a validação dos requisitos junto aos clientes

Esta é uma das melhores boas práticas identificadas pela coleta de dados. Um grande problema enfrentado pelo pólo de desenvolvimento era realizar a validação de requisitos à distância. Diante disto, a equipe de desenvolvimento, o coordenador de negócios e os clientes combinaram que, para algumas situações, o fechamento deveria começar a ser feito de forma presencial, isto é, com o deslocamento de um analista do pólo de desenvolvimento para reunir com os clientes e o coordenador de negócios na cidade do cliente ou vice-versa (os clientes se deslocariam para a cidade onde os analistas trabalham). O importante é que existissem dois momentos de contato presencial, um na elicitação e outro na validação de requisitos, tal como mostra o trecho abaixo, retirado de uma entrevista com um analista:

*[ent] Analista: “Uma ação que foi tomada é que a partir de agora, toda especificação tem que ter, no mínimo, dois momentos: a especificação inicial [elicitação de requisitos] e uma segunda reunião pra fechamento [validação de requisitos] dessa documentação presencialmente e não só enviando a documentação por e-mail. Então, se forem possíveis mais reuniões ao longo do tempo, são permitidas. Se houver verba, disponibilidade... É possível, mas no mínimo agora a gente tem que ter uma no início e uma no final.”*

Um dos fatores apontados como benéficos em se realizar a validação de requisitos de modo presencial está relacionado com a dedicação dos clientes. O trecho abaixo foi retirado de uma entrevista com um analista e retrata a situação em que os clientes foram deslocados de seu ambiente de trabalho para o pólo de desenvolvimento e foram obrigados a se dedicar ao aprimoramento da documentação:

*[ent] “A gente faz uma reunião só pra forçar que ele [o cliente] venha pra cá, pra forçar que ele leia aquele documento, gaste aquela semana lendo e melhorando o documento. E a gente já tem um histórico aí de que isso não acontece quando o cara não vem”.*

Um ponto positivo, particularmente observado nas situações em que os clientes foram deslocados para o pólo de desenvolvimento, foi tornar a reunião de validação de requisitos

mais dinâmica. De um modo geral, os dados coletados mostraram que a validação presencial é bastante produtiva porque os requisitos são mais bem compreendidos, isto é, uma vez que clientes e analistas se encontram, eles têm a oportunidade de discutir, sanar dúvidas e tornar o entendimento do requisito comum a todos os presentes na reunião. Além disso, todos os analistas estão presentes e são capazes de subsidiar os clientes com recursos mais detalhados, por exemplo, uma tela do sistema. O trecho abaixo foi retirado de uma entrevista com um analista e ilustra a situação:

*[ent] “Por isso, a equipe de desenvolvimento de Belém fez questão que os clientes viessem pra cá, pra sair de seu ambiente de trabalho e poder dar o feedback para a equipe que havia elaborado os documentos de especificação de requisitos [embora apenas um tenha participado da elicitação de requisitos, toda a equipe participou da documentação]. Também foi interessante o cliente vir pra Belém porque à medida que ele pedia alterações, a equipe já ia fazendo – isso deu dinamicidade ao processo. Quando o cliente encontrasse um problema, por exemplo, em um caso de uso, a pessoa responsável pelo artefato estaria presente e já poderia consertar os erros, novamente mostrando a dinamicidade das reuniões”.*

Para casos em que ocorra o inverso, isto é, quando o analista é deslocado para realizar a validação no contexto do cliente, espera-se que ao retornar para o pólo de desenvolvimento, o analista consiga repassar o requisito ao restante da equipe com a menor quantidade possível de detalhes mal compreendidos. O trecho abaixo foi retirado de uma entrevista com o coordenador de negócios e mostra as vantagens da validação presencial:

*[ent] “A reunião presencial é boa porque serve pra fechar o entendimento do documento. Você dá o máximo pro cliente, pra ele ver como as coisas vão ficar [telas]. E ele tem a oportunidade de dizer que quer tal componente com tal tamanho, com aquela cor... É uma semana [três, quatro dias] que você fica junto ao cliente discutindo se faltou alguma coisa. Não é só o cara ler sozinho. Se ele ler sozinho é uma coisa, se ele ler junto ao analista é outra coisa. O analista pode tirar dúvidas sobre a especificação, detalhar mais o documento. Temos um cliente que quer que escreva tudo isso no documento. Isso vai de cliente pra cliente. Então o fechamento acontece presencialmente quando todos percebem que precisa haver aquele contato, quando ainda existem muitas dúvidas”.*

A importância da validação presencial não só foi citada durante as entrevistas como também durante as observações.

#### 4.2.1.6 Executar práticas de gerência de conhecimento

Como visto na seção de problemas (seção 4.1.2.1), a empresa Alpha enfrenta uma série de dificuldades relacionadas à Gerência de Conhecimento, que são acentuados em virtude da

distância entre os *stakeholders*. Da mesma forma que os dados coletados identificaram problemas, também apontaram algumas boas práticas realizadas pelos *stakeholders*.

Um dos pontos identificados com a coleta de dados consistiu na preocupação de alguns líderes em disseminar o conhecimento das pessoas mais experientes na empresa para as menos experientes, tal como mostra o trecho abaixo, retirado de uma entrevista com um líder de projeto:

*[ent] “Quando a reunião [de elicitação de requisitos] é fora daqui [do pólo de desenvolvimento], a gente tem muita restrição orçamentária, então não dá pra mandar uma equipe grande, então no máximo quando vão, são duas pessoas. Aí eu pego um experiente pra ir e um menos experiente pra acompanhar e que já vai aprendendo”.*

Apesar do líder de projeto em questão mencionar esta estratégia, não se conseguiu presenciar a mesma sendo posta em prática durante o estudo etnográfico. Apenas um analista foi deslocado nas duas reuniões remotas que foram observadas (elicitação e validação de requisitos).

Os dados coletados mostraram que a equipe de desenvolvimento procura registrar lições aprendidas, ainda que o processo seja inicial, ainda imaturo e pouco difundido entre a equipe, tal como mostram os trechos abaixo, retirados de entrevistas com analistas:

*[ent] “A gente não documenta lições aprendidas, mas a gente conversa entre a gente mesmo e vê o que deu certo no projeto e o que deu errado. O protótipo foi proposto assim. O primeiro que fez isso acho que foi o analista [X]. Ele fez essa reunião com protótipo. Aí no final do projeto, nós concluímos: ‘olha, vamos fazer com protótipo que deu certo’. A gente faz bastante isso. As coisas que dão certo, a gente procura conversar e adotar”.*

Há também, segundo os dados coletados, um esforço em se registrar o *rationale* envolvido na solução elaborada para um determinado problema do cliente, como mostra o trecho seguinte, retirado de uma entrevista com um analista:

*[ent] “[O administrador de dados] teve muita dúvida. A gente discutiu por e-mail várias vezes. Aí eu especifiquei tudo lá [no documento] que a gente tinha visto e que a gente ia mudar, e aí escrevi um e-mail enorme... Pra cada item, cada coluna que foi adicionada ao modelo de dados, tudo que eu ia fazer... E mesmo assim ele teve várias dúvidas. A gente explicou depois por e-mail, por telefone”.*

Na situação acima, o administrador de dados apresentou muitas dúvidas por motivos já discutidos na seção de pontos negativos (seção 4.1.1.2). Por este motivo, o líder da equipe de desenvolvimento, orientou que o analista responsável pelas alterações no modelo de dados,

enriquecesse-o com a descrição dos campos criados, isto é, com a definição dos campos, o *rationale* que motivou sua criação, etc. Esta atividade foi acompanhada durante uma das observações conduzidas no pólo de desenvolvimento da empresa Alpha:

*[notes] O analista [o mesmo da entrevista acima] retorna à sua mesa. Agora ele está elaborando um documento no Microsoft Word com alguns atributos. No decorrer do modelo, o analista adiciona “labels” para determinados campos. Essas labels são dadas a atributos novos para explicar ao administrador de dados o que exatamente o atributo quer dizer. (...) Percebo que em algumas labels, o analista coloca uma espécie de regra de negócio para detalhar ainda mais os campos para o administrador de dados, por exemplo, o que o campo quer dizer e por que ele foi criado.*

Ainda que tímido, pode-se afirmar que existe um esforço por parte da equipe de desenvolvimento em compartilhar o *rationale* utilizado para solucionar um determinado problema. Isto é muito importante, sobretudo, dentro do contexto da empresa Alpha, em que o analista e o AD estão geograficamente distantes e sofrem com os efeitos da distância sobre a comunicação.

Finalmente, aquilo que se pode considerar como a melhor prática para gerenciar o conhecimento: a existência de *knowledge brokers*<sup>11</sup>, como os trechos abaixo indicam:

*[ent] “Como eu fiz a elicitação de requisitos, então eu meio que concentro as informações. O analista X foi também pra fazer o encerramento dos requisitos, então também concentra um pouco nele por causa disso”.*

*[ent] “O jeito é chegar, conversar, explicar e deixar a documentação disponível pra todo mundo consultar a hora que quiser e ficar [você mesmo] disponível pra tirar dúvida da equipe”.*

O papel de *broker* também foi identificado no lado do cliente:

*[ent] Pesquisadora: “O cliente que entra em contato com o analista de requisitos é o usuário do sistema?”*

*Analista: “Ele é o gestor do usuário. Ele participa do dia-a-dia do usuário. Ele, muito provavelmente, é um usuário. Ele é um cara que, digamos, centraliza as dúvidas dos usuários dentro da empresa deles”.*

O trecho acima mostra que o gestor do usuário é a pessoa que centraliza conhecimento em si para repassar aos analistas de requisitos da empresa Alpha. Seu papel é equivalente ao do analista que se desloca do pólo para levantar os requisitos junto ao cliente e retorna para

---

<sup>11</sup> Knowledge brokers são indivíduos que agem como intermediários entre duas partes que não se comunicam (Brössler, 1999).

repassar o conhecimento aos seus colegas. O trecho abaixo foi retirado de uma entrevista com o cliente, que neste caso é o gestor do usuário, e ilustra como esta centralização de conhecimento ocorre:

*[ent] Cliente: “O segundo módulo do projeto era exatamente o trabalho da minha atribuição direta”.*

*Pesquisadora: Isso significa que nesse momento você ficou mais isolado conversando com a empresa Alpha, como fornecedor único [de requisitos] ou não?”*

*Cliente: “Não. Nunca houve essa posição de fornecedor único. O que acontece é que sempre houve no SYS o seguinte modelo: nós buscamos o subsídio nos usuários e a minha função é concatenar esses objetivos para objetivos institucionais. Eu tenho que pegar essas necessidades, distribuídas em todos os usuários, locais, regionais, nacionais de todas as esferas, e transformar isso numa prioridade nacional. No total hoje são mais de 400 representantes”.*

De uma forma geral, a existência de *brokers* tanto do lado dos analistas da empresa Alpha quanto do lado dos clientes possibilita que, embora a vasta quantidade de conhecimento envolvida na concepção do SYS esteja centralizada em algumas poucas pessoas, estas se empenham em disseminá-la aos interessados. Sobretudo quando se trata da empresa Alpha já que o analista que viaja para se reunir com o cliente e que, portanto, concentra toda a informação, se mantém disponível para que os demais colegas tirem suas dúvidas sobre um determinado requisito. E o mais importante, que é a pessoa detentora do conhecimento reconhecer seu papel no processo de desenvolvimento e a importância de se manter disponível para prestar esclarecimentos aos demais analistas.

#### 4.2.1.7 Utilizar metodologias / ferramentas

A coleta de dados identificou que diversas metodologias são utilizadas pela empresa Alpha para suavizar os efeitos causados pela distância. Uma delas é a utilização de diagramas de atividades para agilizar a condução de reuniões com o cliente. A necessidade surgiu quando os analistas começaram a perceber que o diagrama facilitava o entendimento dos clientes, como mostra a nota de campo a seguir, registrada durante a reunião de validação de requisitos:

*[notes] O analista me conta que eles [os analistas] não costumam mostrar os diagramas para os clientes. Agora que eles estão começando a adquirir esse costume porque se percebeu que os clientes compreendem melhor via diagrama.*

A análise dos dados revelou que o cliente concorda que os diagramas melhoram o entendimento do fluxo de negócio:

*[ent] Pesquisadora: “Eles sempre utilizaram diagramas com vocês?”*

*Cliente: “Sempre”.*

*Pesquisadora: “É aquele de atividades?”*

*Cliente: “É! Esse mesmo! Ele facilita muito a nossa compreensão porque o grupo tem essa articulação de trabalhar com o visual”.*

O diagrama de atividades é útil porque ilustra o fluxo das atividades do cliente, assegura que ambos (tanto o cliente quanto o analista) estão falando sobre a mesma coisa, garantindo o entendimento comum do negócio. A nota de campo abaixo foi escrita durante uma reunião de validação de requisitos. Na ocasião, o analista está elaborando o diagrama e um dos clientes se manifesta sobre a utilidade da figura:

*[notes] O analista continua elaborando o diagrama de atividades. Um dos clientes acha o diagrama muito conveniente e sugere que ele esteja disponível para o resto dos clientes, junto com os demais documentos.*

Outra metodologia utilizada pelos analistas para auxiliar no processo de desenvolvimento consistiu na criação de protótipos junto ao cliente, durante a elicitação de requisitos. Os protótipos são úteis para ensinar a navegação do sistema a funcionários novatos, como mostra a nota de campo abaixo. Na ocasião, o analista estava realizando tarefas de especificação de requisitos no pólo de desenvolvimento:

*[notes] O analista retorna para o documento de especificação de caso de uso. Ele loga no SYS e navega até uma determinada tela. Ele também consulta os protótipos elaborados pelo outro analista durante a reunião de elicitação de requisitos, e lê as duas regras de negócio relacionadas com a tela em questão. (...) As regras de negócio são adicionadas ao documento de especificação de requisitos.*

Os protótipos também são úteis para facilitar a etapa de elicitação de requisitos, uma vez que os clientes têm a oportunidade de opinar sobre como a interface do sistema será, bem como o processo de repasse de requisitos. O trecho abaixo mostra a opinião de um analista sobre a utilidade dos protótipos, principalmente no repasse de requisitos e posterior implementação:

*[ent] Analista: “Lá na reunião [de levantamento de requisitos], a gente chega, abre o FrontPage e faz um protótipo da tela. (...) Eu fiz assim e dois antes de mim fizeram assim também. Faz um prototipozinho na hora e coloca as regras em baixo, lá mesmo na própria tela. E assim a gente vai montando tela por tela. Chega aqui é mais fácil pra repassar. Por quê? Já está a carinha da tela lá e as regras ficam em baixo então fica mais fácil de lembrar e repassar. Com isso, o resto da*

*equipe só faz anotar o que tem que fazer. Depois é só dividir (quem fica com qual caso de uso). Já estão no protótipo todas as regras. Fica mais tranquilo pra documentar”.*

*Pesquisadora: “Tu achas melhor o repasse com os protótipos ou sem os protótipos?”*

*Analista: “Com os protótipos eu acho bem melhor. Facilitou muito. Antigamente era sem protótipo. Era só um ‘documentão’ com um monte de regra lá dentro que a gente não [conseguia repassar facilmente]... eram coisas que a gente não tinha idéia de como ficaria aquilo implementado. Então com o protótipo melhorou bastante mesmo”.*

O trecho seguinte já mostra a opinião do cliente. Para eles, o protótipo é interessante porque dá ao cliente a oportunidade de opinar sobre como ficarão as telas do sistema:

*[ent] Pesquisadora: “Eles mostram alguma tela pra vocês?”*

*Cliente: “Nas nossas últimas especificações, já foi assim”.*

*Pesquisadora: “O que você acha de eles já mostrarem as telas?”*

*Cliente: “Excelente! É ótimo! (...) Desde que as nossas documentações passaram a conter telas, se você for ver os casos de uso do SYS de 1 ano e meio pra cá, tá cheio de tela no final. E as telas é que tão falando o que a gente [cliente] tem que fazer. Na realidade, o texto é pra descrever os campos da tela. Aí fica mais fácil. (...) Antigamente, há uns 3 anos, a gente alterava muito requisito porque a gente não sabia como aquilo que a gente pedia em txt ia se comportar na tela. Eu acho que foi muito por isso que a gente migrou pro modelo visual [protótipos]”.*

Já o trecho abaixo mostra qual a opinião do coordenador de negócios acerca da adoção de protótipos durante o levantamento de requisitos:

*[ent] “Eu considero que protótipo deveria ser obrigatório. Quando a gente apresenta pro cliente, a gente nota que muito dificilmente a gente perde o rumo. Principalmente se é um projeto novo ou uma alteração radical, grande. Você não apresenta num primeiro momento, aí quando apresenta a tela já pronta [implementada], tem muita mudança. O protótipo a gente não faz pra todos os casos, mas em alguns sistemas é imprescindível. Há casos em que a gente somente fechou a especificação quando fechou o protótipo. Quando os clientes já tinham uma visão e sabiam o que iam receber. Eu acho uma excelente prática”.*

O estudo etnográfico identificou muitas evidências de que o protótipo é uma boa prática. Os trechos a seguir foram retirados de notas de campo registradas durante as reuniões de elicitação de requisitos, na cidade do cliente, e retratam alguns momentos em que o analista está desenhando o protótipo:

*[notes] O analista fez rabiscos no intervalo de tempo entre ontem e hoje. São rabiscos do protótipo no FrontPage. Ele mostra uma das telas do protótipo para o cliente, que dá palpites na interface. Isso mostra que o esquema dos protótipos proposto pelo pólo de desenvolvimento está*

*funcionando. (...) Muitas dúvidas são tiradas enquanto ele desenha as telas. O cliente e o coordenador de negócios acompanham. O analista descreve as regras de negócio na própria interface que está criando. O cliente auxilia no processo, ditando as regras enquanto o analista digita. As regras de negócio são enriquecidas com exemplos. As regras de negócio são colocadas ao lado do respectivo campo ou componente de interface – e são descritas mais abaixo. Por exemplo, ao lado de um botão, o analista escreve RNGI e mais abaixo, no mesmo formulário criado no FrontPage, ele descreve qual comportamento a tela deve apresentar quando o botão for clicado.*

*[notes] Durante a reunião, o coordenador de negócios pergunta se da última vez tinha sido especificado dessa forma. O cliente responde que “com esse nível de detalhamento, não”. O coordenador diz então que as próximas reuniões devem ser feitas assim. Ele mostra que o esquema dos protótipos é ideal porque “o que vocês (clientes) pediram, vai estar lá!”. Ele também diz que pequenos detalhes podem ser documentados no protótipo, como, por exemplo, o alinhamento de tabelas. Ele diz: “As próximas especificações vão ser desse jeito, a gente pode perder um pouco de tempo, mas depois vai ganhar”, se referindo ao tempo gasto durante a semana de reuniões remotas e o tempo que será poupado no pólo já que a especificação vai chegar bem fechada para a equipe de desenvolvimento. Ele também afirma que a estratégia “evita desgaste pra vocês (clientes) e pra nós (empresa Alpha)...”, e finaliza dizendo que está feliz. Ela também parece não gostar de quando as pessoas vêm especificar e voltam pra Belém com um monte de papel, dando a entender mais uma vez que os protótipos são o ideal.*

Observou-se também que a construção de protótipos pelos analistas da empresa Alpha é apoiada pela criação de protótipos no papel, técnica conhecida como *paper prototyping*. Os dados coletados identificaram momentos, durante a reunião de elicitação de requisitos, em que esta técnica foi posta em prática:

*[notes] O cliente pega um papel e começa a rabiscar coisas. O analista também toma notas. O cliente dá palpites quanto à interface do sistema. O analista então abre o sistema no navegador para clarear as idéias do cliente. O cliente faz rascunhos da interface no papel, o analista compreende o que o cliente quer e segue tomando notas.*

Na situação abaixo, o cliente expôs um problema em um determinado relatório, que estava extenso demais e que precisava ser diminuído. A técnica da prototipagem em papel ajudou a encontrarem rapidamente e facilmente a melhor solução:

*[notes] Finalmente cliente e analista resolveram o problema do relatório. Não criaram nenhuma consulta nova, apenas reformularam as antigas. Uma boa estratégia utilizada nesse momento foi novamente o paper prototyping. O analista desenha o layout do relatório em uma folha de papel e o cliente vai acompanhando. (...) O cliente então diz para o analista: “Lindo, Analista A! Fechou!”. O cliente está satisfeito com o resultado da conversa e com o protótipo desenhado no papel pelo analista.*

Em contextos de DDS em que o tempo é escasso e, portanto, deve ser utilizado da forma mais inteligente possível, a técnica da prototipagem em papel surge como uma metodologia eficiente porque é barata, simples e rápida.

#### 4.2.2 Soluções organizacionais

O estudo etnográfico realizado na empresa Alpha identificou que o coordenador de negócios toma determinadas medidas que não são tecnológicas nem metodológicas, as quais foram denominadas “administrativas”. Estas medidas estão relacionadas principalmente com o esforço que a coordenação de negócios tem em participar ativamente dos projetos sob sua responsabilidade, na tentativa de minimizar os problemas causados pela distância. Esta seção descreve estas soluções e será orientada tanto por notas de campo quanto por fragmentos extraídos de entrevistas, tendo sido estas últimas realizadas com analistas e com o próprio coordenador de negócios.

Um dos esforços do coordenador de negócios é mediar qualquer contato entre clientes e analistas. Este tipo de contato direto pode se tornar extremamente delicado se os clientes se utilizarem do artifício de negociar diretamente com os desenvolvedores para forçar acordos informais, por exemplo, de modificação de escopo. O nível de confiança que já é naturalmente baixo em contextos de DDS pode ficar ainda mais abalado, o que causaria um desconforto desnecessário entre os *stakeholders*. O trecho a seguir foi retirado de uma entrevista com um analista e explica a importância de um coordenador ser atuante:

*[ent] Pesquisadora: “Como é o relacionamento da sua equipe com o cliente?”*

*Analista: “Depende do cliente. Tem cliente que tem perfis dos mais variados possíveis. Tem alguns [clientes] que realmente o relacionamento é muito bom, é muito fácil de lidar, é bem tranquilo, principalmente quando os coordenadores atuam com bastante participação. Quando eles [os coordenadores] estão a par de todos os fatos do processo, o relacionamento com o cliente tende a ser melhor”.*

O trecho a seguir foi retirado de uma entrevista com o coordenador de negócios e mostra seu esforço em tomar conhecimento sempre que houver contato (ou tentativa de contato) entre desenvolvedor e cliente, bem como sua preocupação com a ocorrência de que acordos sem a sua ciência:

*[ent] “Eles [os analistas] fazem o documento de especificação de requisitos, encaminham sempre pra coordenação e eu faço muita questão que esses documentos passem pela coordenação. Por quê? Pro cliente não dizer que não recebeu, e aí a coisa está lá pra ele aprovar e ele não lê, ele não aprova. Ou então tem essa questão do vínculo do cliente com o líder. Então o líder está lá*

*com a equipe, aí o cliente quer entrar em contato direto com eles lá [no pólo de desenvolvimento], eu proíbo. Eu digo ‘se ligarem pra você, manda falar comigo!’, eu corto. (...) Eu observo que, por exemplo, alguns dos projetos em que nós tivemos problema não tinham a presença da coordenação. Fica líder de projeto junto com cliente. E aí, a pressão é grande, quando você vê, os acordos já foram montados. Não são todos os projetos, mas existe isso, há essa dificuldade. (...) Quando tem fechamento, eu gosto de participar. Eu trago pra cá pra Brasília. É o caso de se controlar a situação, ficar por dentro do que está acontecendo, tomar ciência dos acordos”.*

O trecho acima nos permite inferir que o coordenador de negócios prefere estar presente nas reuniões para controlar melhor a situação e para não acontecerem casos em que clientes peçam coisas que não estão na demanda e os analistas simplesmente aceitem, via acordos informais. Entretanto, o coordenador é uma pessoa extremamente atribulada e, por isso, não consegue estar presente o tempo todo em todos os projetos, ainda que se esforce para acompanhar diariamente as reuniões (no caso de elicitação, validação e homologação de requisitos), como mostram os trechos abaixo:

*[ent] “Eu tenho 30 projetos, então não posso ficar a semana toda lá, presente. Então eu vou na abertura [da homologação] e deixo o líder lá encarregado. Aí todo dia eu vou lá num horário pra ver como é que está, ouvir do cliente o que aconteceu, ver os casos reportados no Mantis, quantos casos foram reportados”.*

Durante a homologação, os clientes são orientados a testar as funcionalidades disponibilizadas pelo pólo e, no caso de encontrarem erros, reportarem-nos para a equipe de desenvolvimento resolvê-los. Porém, há casos em que os clientes tentam pedir modificações nas funcionalidades entregues, mesmo com o documento de requisitos atestado muito previamente por eles mesmos. O trecho abaixo, retirado de uma entrevista com o coordenador de negócios, mostra que o contato direto entre analistas e clientes está proibido, a menos que o coordenador de negócios esteja presente:

*[ent] “É uma determinação da superintendência: analista e líder de projeto não podem conversar com cliente. Não podem entrar em contato pra fazer qualquer fechamento de prazo, de cronogramas, de alterações, acordos. Não podem. Não podem participar sem a presença da coordenação”.*

Quando um acordo informal ocorre entre os clientes e a equipe de desenvolvimento, o coordenador de negócios precisa ser firme no sentido de não permitir que os requisitos entrem para a homologação. Esta cautela serve para impedir que requisitos incluídos tardiamente prejudiquem a homologação ao ponto de invalidá-la. A homologação, portanto, deve ocorrer

como o previsto e, caso o cliente queira modificar alguma funcionalidade, isto deverá ser tratado como uma nova demanda, isto é, desde o início do processo de desenvolvimento.

*[ent] “Se [o cliente] quiser, a gente homologa aquilo do jeito que está e depois o cliente abre uma demanda com as modificações necessárias e a equipe implementa. E aí, tem que ser muito firme porque eles sempre vão dizer que podia mudar [o requisito]. E se a gente ceder e fizer as alterações e não homologar? Ninguém vai dizer que foi só uma coisinha que mudou. Vão dizer que a culpa é da empresa Alpha. Então, por causa disso, eu tenho que ser muito criterioso. É uma homologação de funcionalidades que já foram especificadas. Novas ficam pra depois”.*

Por este motivo, a homologação ocorre na cidade do cliente. Ali, o coordenador de negócios é capaz de monitorar o processo de perto já que ele está localizado nesta mesma cidade e, assim, evitar que mudanças informais de requisitos aconteçam:

*[ent] “Já a homologação veio pra cá, pois é uma maneira dos clientes virem pra cá e da gente monitorar o ambiente todo pra não ter queda (do sistema), de acompanhar bem e de estar perto de tudo”.*

Esta boa prática torna-se válida dentro do contexto da empresa Alpha porque, como vimos no capítulo de pontos negativos, a distância causa a constante alteração do escopo das demandas em momentos inadequados, pois clientes e analistas não se vêem com frequência, então sempre que uma reunião presencial ocorre, os clientes sentem a necessidade de informar suas necessidades aos analistas.

Outra preocupação do coordenador de negócios é assegurar que o documento oriundo de uma reunião (via áudio) seja bem compreendido pelos clientes. Sempre que a equipe de analistas se reúne com os clientes, o coordenador faz questão de ler o documento gerado e adaptá-lo para uma linguagem menos complexa, tal como mostra o trecho abaixo, retirado de uma entrevista com o coordenador de negócios:

*[ent] “A menos que a conversa seja essencialmente pra um esclarecimento técnico, eu participo. O líder de projeto e os analistas dão uma conotação muito técnica. Quando eu vou escrever, eu dou uma conotação mais gerencial. A equipe técnica diz que não dá porque tem que mexer no ClearCase, fazer SS [solicitação de mudança], por exemplo. Quando chega na minha mão, eu digo que nós temos ‘atividades de planejamento’ pra facilitar o entendimento no lado dos clientes, pra não haver confusão. Eu digo pra eles [analistas]: ‘me botem na linha de frente porque esse cliente vai discutir comigo e ele não vai poder dizer que disse nada pra vocês porque está tudo no meu caderno’ [ele abre o caderno e me mostra suas anotações sobre todas as áudios realizadas]. Eu faço questão de registrar tudo”.*

Esta boa prática promove a disponibilização de informação a todos os envolvidos em um projeto de DDS e permite que os *stakeholders* tenham o conhecimento comum sobre o que foi definido em uma reunião, portanto ameniza alguns pontos negativos citados na seção anterior.

O coordenador de negócios, por fim, reconhece que está mais atuante e que, por isso, a incidência de problemas tem diminuído bastante. O trecho abaixo mostra um pouco da participação ativa do coordenador de negócios dentro do projeto e mitiga os problemas identificados na seção anterior sobre a mudança freqüente no escopo das demandas:

*[ent] “A presença da coordenação tem diminuído muitos problemas. A gente [da coordenação] está mais atuante. Mandou alguma coisa pro líder que não veio pra coordenação, então não está valendo. Assinou o anexo com qual requisito? Então é esse que está valendo. Quem diz, sou eu. (...) Nas reuniões, eu participo, estou acompanhando sempre tudo que está acontecendo. Fechamentos, cronogramas, todos têm a minha participação e saem da coordenação. Áudios todas com a coordenação”.*

### 4.2.3 Soluções tecnológicas

Dentre as soluções tecnológicas identificadas na análise de dados, as que mais se destacam são:

- Uso de áudio-conferência;
- Uso de correio eletrônico;
- Uso de sistema interno para armazenamento de arquivos; e
- Uso de videoconferência.

Trataremos estas soluções uma a uma nos parágrafos a seguir.

#### 4.2.3.1 Uso de áudio-conferência

As áudio-conferências são utilizadas para manter a comunicação entre os *stakeholders*, por exemplo, quando, após a elicitación de requisitos na cidade do cliente, a equipe de desenvolvimento apresentar dúvidas pontuais sobre um determinado assunto. O trecho abaixo foi retirado de uma entrevista em que um analista defende o uso da áudio-conferência:

*[ent] “Para mim, a áudio-conferência funciona, pois normalmente eles [equipe de desenvolvimento] utilizam este meio para tirar dúvidas pontuais. Em geral, são coisas simples. Às vezes acontece de, durante a áudio, a outra pessoa [cliente] não conseguir responder alguma dúvida. Nestes casos, a pessoa do outro lado [cliente] marca uma áudio de retorno para esclarecer a dúvida do pessoal de Belém. Existem casos, no entanto, que uma nova reunião presencial é necessária”.*

O registro das áudio-conferências é realizado via elaboração de ata de reunião, tal como explica um líder de projeto no trecho de entrevista abaixo:

*[ent] “Então a gente explora muito a áudio. Usa também muito e-mail pra ficar registrado, pra não ficar aquela coisa informal. Então aquelas coisas de maior risco que a gente vê que depois podem gerar algum mal-entendido, a gente escreve e gera ata de reunião. A gente registra o máximo que consegue registrar. Não usamos outros meios a não ser os e-mails e as áudios.”*

Observou-se também que as áudio-conferências podem ser realizadas durante a validação de requisitos. A nota de campo abaixo foi escrita durante uma reunião de validação de requisitos realizada na cidade do cliente. Na ocasião, o analista A estava validando requisitos elicitados pelo analista B. Durante a reunião, uma dúvida surgiu e o analista A utilizou a áudio-conferência para tirar a dúvida com o analista B que desta vez permaneceu no pólo de desenvolvimento:

*[notes] O cliente se queixa de coisas que foram acordadas de uma forma durante a reunião de elicitação com o analista B, mas que foram documentadas de outra forma. O analista A liga para o analista B para tirar a dúvida. O analista B então explica o requisito para o analista A que, por sua vez, o explica para os clientes e re-escreve a regra.*

#### 4.2.3.2 *Uso de correio eletrônico*

A coleta de dados realizada neste trabalho sugere que o correio eletrônico é utilizado principalmente para registrar decisões e para enviar artefatos. Isto em contextos distribuídos ameniza a distância, pois facilita a externalização, documentação de conhecimento e, conseqüentemente, sua disponibilização a todos os *stakeholders* do projeto.

Já vimos no tópico anterior que o correio eletrônico é utilizado para registrar decisões tomadas em reuniões realizadas, sobretudo via áudio-conferências. O correio eletrônico também vem sendo usado há muito tempo para o pólo de desenvolvimento enviar documentos (especificação de requisitos, por exemplo) ao cliente, tal como o trecho abaixo, retirado de uma entrevista com o cliente, mostra:

*[ent] Pesquisadora: “Como você tem acesso a esse documento [especificação de requisitos]. É por e-mail?”*

*Cliente: “Nós temos um e-mail e temos um banco de repositório. Esse repositório está começando agora, então de vez em quando a gente tem problemas, mas [o documento] trafega por e-mail”.*

A seção de problemas, no entanto, mostrou que o envio de e-mails tornou-se problemático em virtude das ocasiões em que arquivos grandes demais foram enviados e não foram

recebidos por seus destinatários. Por este motivo, os clientes sugeriram a criação de um banco de repositório, citado no trecho acima, e que será explicado com mais detalhes no tópico seguinte.

#### 4.2.3.3 *Uso de sistema interno para armazenamento de arquivos*

No momento da coleta de dados, os clientes estavam começando a utilizar um portal de colaboração para o armazenamento de arquivos, como mostra o trecho abaixo, retirado da entrevista com o cliente:

*[ent] Pesquisadora: “Como você tem acesso ao documento de especificação de requisitos? É por e-mail? Eles [analistas] deixam em algum repositório?”*

*Cliente: “Nós temos um e-mail e temos um banco de repositório. Esse repositório está começando agora, então de vez em quando a gente tem problemas, mas [o documento] trafega por e-mail. (...) Agora tem o portal X. E no X, o SYS já tem o ambiente dele. É no Moodle e foi desenvolvido por nós e pela empresa Alpha”.*

*Pesquisadora: “Mas foi iniciativa de quem?”*

*Cliente: “Nossa. Do nosso setor de tecnologia”.*

*Pesquisadora: “O X serve pra dar acesso aos artefatos que a empresa Alpha elabora?”*

*Cliente: “Exatamente, e pra gente ter a certeza de que a gente sempre está falando do mesmo documento”.*

Nota-se que o portal X é uma ferramenta bastante promissora, principalmente para ajudar clientes e analistas com a questão da inconsistência entre documentos (já discutida na seção de pontos negativos), visto que as versões mais atualizadas poderiam ser “subidas” para o portal X pelos próprios clientes, os quais não seriam mais obrigados a esperar a empresa Alpha disponibilizá-los.

Os dados também identificaram que o cliente gostaria de usar o portal X para, além de armazenar, agilizar o processo de revisão dos documentos elaborados pela equipe de desenvolvimento:

*[ent] “Se eu puder lá no Portal X, à medida que os meninos (desenvolvedores) tão construindo, eu estar vendo o documento junto, nós não vamos mais perder tempo com análise de documento! (...) Se isso( pudesse, eu acho que seria um ganho assim... Pra mim seria em todos os sentidos. Porque a gente não gastaria tanto tempo na homologação, não teria uma perda de esforço, deslocamento... Quando a coisa chegasse funcionando, já estaria bem mais tranqüila. E tendo uma equipe como o SYS tem, que eu acho que esse é um grande diferencial, eu acho que seria um ganho extraordinário. Eu não consigo nem dimensionar exatamente o tamanho desse ganho”.*

Entende-se que se o cliente fosse capaz de acompanhar a escrita dos documentos de requisitos, poderia identificar primariamente erros de entendimento e consertá-los antes que os requisitos fossem disponibilizados para a validação.

#### 4.2.3.4 *Uso de videoconferência*

A videoconferência foi citada pelo coordenador de negócios como um artifício para amenizar os efeitos da distância, embora não seja utilizada com frequência. O trecho abaixo foi retirado de uma entrevista realizada com o coordenador de negócios e ilustra sua opinião sobre o uso da videoconferência:

*[ent] “Hoje em dia tem a videoconferência. Não gosto de usar esse tipo de ferramenta [vídeo] pra especificação e pra fechamento porque é extremamente cansativo. Mas quando tem corte orçamentário e você não pode deslocar as pessoas, você tem que apelar pra vídeo. Então teve o caso do sistema que precisou de construção de protótipo e esse protótipo seria apresentado por videoconferência. Envolvia gente de Fortaleza e como teve corte [orçamentário], ninguém pôde se deslocar pra lá. Então nós fizemos uma videoconferência. O analista apresentava na tela com o micro e ia mostrando pra outra localidade. Tinha umas 10 pessoas aqui e todo mundo falava de todos os lugares. Você fica morto de cansado, mas é uma ferramenta que ajuda muito.”*

Apesar de o trecho sugerir que a experiência do coordenador de negócios com a videoconferência foi mais negativa que positiva, o recurso visual é reconhecidamente uma ótima ferramenta para suavizar os efeitos da distância. O uso da videoconferência será abordado em maiores detalhes na seção seguinte, onde são apresentadas sugestões de melhorias.

## 5 DISCUSSÃO

Como visto no capítulo 2, existem muitas propostas que visam solucionar ou mesmo minimizar os problemas causados pela distância entre os *stakeholders* envolvidos nas atividades de Engenharia de Requisitos. Estas sugestões podem ser encontradas sob várias naturezas: metodologias, ferramentas e processos. O objetivo deste capítulo é principalmente discutir estes trabalhos identificados durante a revisão bibliográfica sob a luz dos resultados obtidos a partir da análise dos dados coletados no estudo etnográfico e descritos no capítulo anterior. A partir desta discussão, espera-se que o estudo qualitativo proposto nesta dissertação indique novas direções de pesquisa em Engenharia de Requisitos em contextos de Desenvolvimento Distribuído de Software. Esta seção está dividida em três partes: a) pontos em que o estudo etnográfico corrobora os modelos analisados; b) pontos em que o estudo etnográfico corrobora as ferramentas analisadas e c) pontos que ficaram em aberto nos trabalhos analisados, isto é, aspectos que não foram corroborados, mas que tampouco podem ser considerados como refutados pelo estudo etnográfico, os quais, portanto, necessitam ser investigados de forma mais minuciosa.

### 5.1 Considerações iniciais

No contexto deste trabalho, sugere-se que as metodologias sejam compreendidas como práticas de trabalho formais que podem ser incorporadas a um processo de desenvolvimento de software, uma vez que os métodos de Engenharia de Software fornecem a técnica de como fazer para construir software (Pressman, 2004). Assim, metodologias e processos serão discutidos juntos. Primeiramente, questões relacionadas às limitações serão debatidas, bem como sugestões de melhoria para os processos de ER citados no capítulo 2. Em seguida, serão recomendadas metodologias passíveis de serem incorporadas a um processo de ER para contextos de DDS.

A primeira reflexão válida sobre os modelos de processo propostos e os resultados do estudo etnográfico apresentado nesta dissertação reside no fato de o processo adotado pela empresa Alpha se mostrar inadequado diante das especificidades impostas por um contexto de desenvolvimento de software distribuído. Por exemplo, na empresa Alpha, as demandas são criadas e divididas em iterações, onde cada iteração está associada ao desenvolvimento de um conjunto de funcionalidades. Como os clientes estão distantes do pólo de desenvolvimento, torna-se muito oneroso realizar reuniões presenciais para a validação de todos os requisitos. Além disso, a validação dos requisitos através da troca de documentos via e-mail também é problemática (4.1.1.5). Portanto, observou-se que quando os requisitos não são validados presencialmente, eles freqüentemente são traduzidos para funcionalidades incorretas, que não satisfarão as necessidades do cliente. Diante disto, quando a demanda é homologada, e isso engloba todas as iterações nas quais a demanda foi fragmentada, muitos problemas certamente são encontrados (seção 4.2.1.5). Sabe-se que quanto mais tardia a identificação de um *bug*, mais caro se torna seu conserto (Pressman, 2004), portanto, é possível que se a validação de requisitos ocorresse de maneira presencial a cada iteração do processo, muitos problemas poderiam ser identificados nas primeiras etapas do processo e, conseqüentemente, sua correção seria antecipada e menos dispendiosa.

Outra evidência de que o processo da empresa Alpha não está ajustado ao contexto distribuído no qual ela está inserida é sua pouca flexibilidade quanto à modificação de requisitos. O processo atual dita que se ocorrerem mudanças dos requisitos durante o processo de desenvolvimento dos mesmos, as atividades de codificação devem ser suspensas para que sejam retomadas apenas depois da execução da gerência de requisitos. Isto provoca o desperdício de uma grande parte do cronograma, sobretudo se considerarmos o contexto distribuído em que as reuniões presenciais para elicitação de requisitos são raras e existe uma demora para que os documentos sejam revisados pelos clientes (seção 4.2.2.1).

Ainda sobre a não aderência do processo ao contexto de DDS, os resultados descritos no capítulo anterior indicam que uma das causas para a demora no processo é o tempo que o cliente leva para revisar todos os requisitos confeccionados em uma determinada iteração (seção 4.2.1.4). Como oportunidade de melhoria, o próprio cliente sugeriu<sup>12</sup> que o processo da empresa Alpha se tornasse mais colaborativo, que permitisse, por exemplo, que a escrita dos

---

<sup>12</sup> A sugestão foi dada pelo cliente durante a entrevista, porém ainda não havia sido levada à Empresa Alpha até o momento da conclusão deste trabalho.

requisitos fosse realizada de maneira colaborativa e que ele, o cliente, pudesse acompanhar em tempo real a elaboração dos documentos. Assim, o cliente conseguiria monitorar a construção do documento e executar planos de ação o mais cedo possível caso percebesse que algum conceito de negócio não estivesse sendo compreendido corretamente (seção 4.4.3.3).

Dentro desse contexto, os dados coletados confirmam a necessidade de um processo particular para a Engenharia de Requisitos que se adapte às limitações impostas pela distância geográfica na empresa Alpha, uma característica de projetos de DDS. Dito isso, as seções a seguir apresentam uma análise crítica em que se comparam os trabalhos identificados no capítulo 2 (processos de Engenharia de Requisitos, metodologias e ferramentas) e os resultados encontrados no capítulo 4.

## **5.2 Pontos em que o estudo etnográfico corrobora os modelos analisados**

### **5.2.1 Planejamento de reuniões de elicitação**

Uma característica interessante do modelo proposto por Medeiros (2007) é a preocupação que o autor teve em mostrar a importância de uma reunião presencial ser bem planejada e bem conduzida. Os resultados deste estudo mostram que um dos problemas inicialmente enfrentados pela empresa Alpha é a falta de planejamento da reunião. A análise dos dados indica que, no passado, embora as reuniões fossem formais, os envolvidos não tinham o costume de seguir um roteiro pré-definido e, muitas vezes, as pessoas convocadas não eram necessariamente as diretamente interessadas na reunião. Na empresa Alpha, isto acontecia principalmente quando uma reunião de elicitação de requisitos era realizada sem que os clientes estivessem em consenso sobre os requisitos a serem tratados, o que subutilizava a presença dos analistas de requisitos e/ou administradores de dados. Como visto na seção 4.4.1.3, este estudo também mostrou que quando as reuniões ocorrem na cidade do cliente, enviam-se analistas conhecedores do domínio do negócio a ser tratado, bem como enérgicos o suficiente para manter o foco durante a condução da reunião. Da mesma forma, os clientes são orientados a estarem preparados para o evento, ou seja, em consenso acerca dos requisitos que serão tratados para que não se perca tempo definindo questões do negócio na presença do analista de requisitos. Finalmente, é importante saber quem chamar para cada tipo de reunião, assim os envolvidos não desperdiçam seu tempo em reuniões para as quais não são necessários. Assim, torna-se primordial auxiliar os *stakeholders* (tanto analistas quanto clientes) a planejarem suas reuniões para que sejam capazes de realizar encontros produtivos uma vez que o tempo tem um valor ainda maior para projetos distribuídos.

### 5.2.2 Elaboração de protótipos

Uma das recomendações que o modelo de processo proposto por Medeiros (2007) recomenda é que protótipos sejam elaborados durante a elicitação de requisitos. O estudo etnográfico apresentado neste trabalho apóia esta recomendação, pois conforme descrito na seção 4.3.1.1, um dos muitos problemas enfrentados pela empresa Alpha é a dificuldade de repassar os requisitos elicitados remotamente, junto ao cliente, para o restante do pólo de desenvolvimento. A análise dos dados mostra que a empresa passou a elaborar protótipos construídos durante a elicitação de requisitos e a relacionar regras de negócio a estes protótipos, pois percebeu que a ata da reunião era um artefato puramente textual e, portanto, passível de interpretações ambíguas o que dificultava o repasse das informações quando o analista retornava ao pólo. Constatou-se durante a análise dos dados que esta prática foi bem aceita pelo cliente, pelo coordenador de negócios e também pela equipe de desenvolvimento.

### 5.2.3 Disponibilização de informações de contexto

O modelo proposto por Lopes (2004) sugere que o processo de engenharia de requisitos possua atividades específicas para capturar informações de contexto dos usuários e clientes, registrar o vocabulário utilizado na aplicação em um glossário e disseminar informações relacionadas ao negócio, como o escopo do projeto e as necessidades dos clientes. O estudo etnográfico corrobora a ideia, entretanto na via contrária, isto é, no sentido pólo de desenvolvimento – cliente. De um lado, a análise dos dados indica que o líder disponibiliza ao cliente um cronograma “macro”, com datas relacionadas ao fim de cada etapa (requisitos, implementação e testes). Por outro lado, os dados também mostram que o cliente gostaria de estar informado sobre o andamento das atividades do pólo de desenvolvimento em um nível mais detalhado como mostrou a seção 4.3.2.4.

Como solução para este problema, sugere-se que o modelo de processo proposto por Lopes (2004) considere a utilização de PSEEs (*Process-centered Software Engineering Environment*) ou ADSs (Ambientes de Desenvolvimento de Software) (Gruhn, 2002). Os PSEEs são ambientes que permitem a modelagem e a execução de processos de desenvolvimento de software. Modelar um processo significa, de um modo geral, definir papéis para os membros de uma equipe, criar atividades que serão alocadas e realizadas pelos membros da equipe de acordo com seus papéis, criar relações de predecessão ou de sucessão entre estas tarefas, definir artefatos que serão gerados a partir de ou consumidos pelas tarefas

conforme elas sejam executadas (Gruhn, 2002). Executar um processo significa de fato realizar as tarefas (modeladas e alocadas) pelos membros da equipe.

Sabe-se, porém, que pode haver uma relutância em disponibilizar o processo da empresa Alpha para seus clientes, uma vez que o processo de desenvolvimento de software pode ser considerado propriedade intelectual da empresa. Além disso, a exposição dos detalhes da execução do processo da empresa Alpha pode ser um complicador para a empresa Alpha ao expor suas atividades de maneira inadequada. Desta forma, um interessante aspecto de pesquisa a ser explorado em trabalhos futuros seria entender o que exatamente os clientes da empresa Alpha precisam e/ou gostariam de estar cientes sobre o processo da empresa Alpha sem que isto expusesse de maneira inadequada a empresa Alpha. Finalmente, dentre tantos PSEEs existentes na literatura, optou-se por recomendar o uso do WebAPSEE (Lima et al, 2006) principalmente por ser um PSEE regional e por possuir uma versão livre.

#### **5.2.4 Reconhecimento da GC como um desafio para a ER distribuída**

Embora o modelo proposto por Lopes (2004) não pareça estar diretamente apoiado por atividades de Gerência de Conhecimento (GC), o autor reconhece que a GC é um desafio para a Engenharia de Requisitos distribuída. O estudo etnográfico descrito neste trabalho está alinhado com o modelo proposto por Lopes (2004), pois reforça que a GC é uma preocupação concreta e que deve ser considerada em projetos distribuídos. Alguns dos problemas relacionados à Gerência do Conhecimento (seção 4.1.2.2), enfrentados pelos analistas da empresa Alpha são:

- a) centralização do conhecimento em uma única pessoa e conseqüente interrupção contínua desta pelos demais membros da equipe para esclarecimento de dúvidas;
- b) entrada de novatos que não são imediatamente treinados no negócio do cliente; e
- c) ausência de mecanismos para gerenciar o conhecimento, como ferramentas e ou metodologias.

Por outro lado, a empresa Alpha adota boas práticas de gerência de conhecimento durante seu processo de engenharia de requisitos, tal como a seção 4.4.1.6 mostrou, tais como:

- a) a disseminação de conhecimento dos analistas mais experientes para os menos experientes;
- b) o registro de lições aprendidas;
- c) o registro do *rationale* por trás de uma solução técnica; e

d) finalmente, a existência de *brokers* informais tanto do lado do cliente quanto no pólo de desenvolvimento.

### 5.3 Pontos em que o estudo etnográfico corrobora as ferramentas analisadas

Existem várias ferramentas para amenizar os efeitos da distância e facilitar as atividades de requisitos realizadas em contextos distribuídos. A pesquisa bibliográfica em que este estudo etnográfico se baseia considerou as seguintes ferramentas:

- *Econference*: permite comunicação síncrona por *chat*, quadro de comunicação e apoio ao conceito de *awareness*, através do chat integrado à agenda (Calefato e Lanubile, 2005);
- CRETA: integra ferramentas de colaboração, promove o registro, compartilhamento e reuso de informação, facilita a comunicação, coordenação e cooperação, e apóia o conceito de *awareness* via *chat* (Togneri et al, 2002);
- CODIPSE-Req: promove colaboração e coordenação entre os *stakeholders* através do agendamento de reuniões e de eventos e mantém as alterações dos requisitos através de links com os fatores causadores da modificação (Brito e Vasconcelos, 2006);
- *TeamWave*: simula uma sala de reunião equipada com ferramentas de colaboração, como quadro de anotações, facilitando a ocorrência de reuniões de negociação de requisitos (Herlea e Greenberg, 1998); e
- *Centra Symposium*: fornece recursos visuais, como apresentação de slides e compartilhamento do navegador, que podem auxiliar a condução de reuniões virtuais (por exemplo, validação de requisitos) (Lloyd et al, 2002).

A análise dos dados mostrou que os analistas e os clientes da empresa Alpha utilizam bastante áudio-conferências. Entretanto, estas carecem de recursos visuais quando precisam discutir um documento via áudio (seção 4.3.2.3). Isto corrobora a importância da construção de ferramentas com funcionalidades de *quadro de anotações* e *compartilhamento de navegadores*, pois tais funcionalidades permitem que a revisão do documento ocorra com mais clareza já que permite que todas as partes envolvidas sejam capazes de visualizar o mesmo arquivo, ao mesmo tempo, ainda que remotamente.

Percebeu-se também que a empresa Alpha tem problemas de gerência de conhecimento, como centralização de conhecimento em uma só pessoa, ausência de mecanismos para disseminar o conhecimento centralizado e falta de *awareness* do contexto remoto. Este problema é causado principalmente porque apenas um analista é enviado para se reunir presencialmente com o cliente em razão de restrições orçamentárias. Isto também confirma a

contribuição de ferramentas que oferecem funcionalidades de *chats integrados à agenda dos stakeholders* e de *registro, compartilhamento e reuso do conhecimento*.

Ainda, segundo os dados analisados, a empresa Alpha atravessa problemas na gerência de seus requisitos uma vez que o artefato resultante da análise de impacto não é rico o suficiente para ilustrar aos clientes quais os reais impactos de uma modificação de requisitos. Isto pode colaborar para que os clientes se sintam excluídos do processo de desenvolvimento e assumam verdades de maneira precipitada, como uma falsa má vontade da empresa Alpha em realizar as modificações solicitadas. Em ambientes distribuídos, a distância dificulta a comunicação, aumenta as chances de ocorrerem mal-entendidos deste tipo e, conseqüentemente, pode abalar a relação de confiança e o sentimento de equipe entre os *stakeholders*. Isto reforça a contribuição de ferramentas que auxiliem *na manutenção e na rastreabilidade das modificações* realizadas nos requisitos.

Conclui-se que o trabalho etnográfico apresentado nesta dissertação corrobora a importância de todas as ferramentas citadas acima. Um trabalho futuro interessante seria justamente avaliar estas ferramentas em um contexto real de desenvolvimento, como o da empresa Alpha.

<b>Funcionalidades</b>	<b>Problemas</b>
Chat (e chat integrado à agenda de compromissos)	Falta de awareness do contexto remoto
Quadro de anotações e compartilhamento de navegador	Falta de recursos visuais quando precisam analisar um documento via áudio-conferência
Registro, compartilhamento e reuso de informação	Centralização de conhecimento
Rastreabilidade entre as alterações dos requisitos e as causas das modificações	Ilustrar claramente a análise de impacto ao cliente

## **5.4 Pontos que ficaram em aberto nos trabalhos analisados**

### **5.4.1 Formas de validação de requisitos em DDS**

Uma das recomendações encontradas no modelo de processo proposto por Medeiros (2007) diz respeito à tarefa de validação de requisitos ser realizada à distância e assincronamente. Esta revisão assíncrona está prevista como uma das atividades da etapa de validação de requisitos que o modelo propõe. Os resultados do estudo etnográfico, entretanto, indicam que a empresa Alpha enfrentou problemas com a validação dos requisitos à distância e assincronamente. Quando a validação era realizada remotamente, os documentos de requisitos eram elaborados no pólo de desenvolvimento e disponibilizados para que os clientes os revisassem remotamente. Entretanto, a distância prejudicava severamente a comunicação, além disso, os clientes dedicavam pouco tempo à tarefa de revisão uma vez que

não estavam alocados exclusivamente para a mesma. A pouca dedicação atrasava a validação de requisitos e refletia em todo o processo de desenvolvimento de software. Isto não acontece mais na empresa Alpha, uma vez que a validação de requisitos ocorre através de uma reunião presencial, com o deslocamento de um analista para a cidade do cliente. Neste caso, a leitura do documento ocorre de forma conjunta com o cliente, que se mostra mais dedicado para esta tarefa, e é seguida do registro das considerações apontadas pelos clientes (seção 4.2.1.1). Ainda assim, o estudo etnográfico mostrou que apesar dos ganhos adquiridos com as reuniões presenciais para validação de requisitos, a duração destes encontros ainda não parece ser suficiente para que os objetivos planejados sejam completamente atingidos (ver seção 4.1.1.5).

#### **5.4.2 Técnicas para gerenciar requisitos**

Esta subseção discutirá como os trabalhos citados acima abordam a gerência de requisitos em seus modelos de processo.

O modelo de processo proposto por Medeiros (2007) não contém atividades de gerência de requisitos. Já o modelo sugerido por Lopes (2004) possui uma atividade específica para gerenciar as modificações de requisitos, na qual se realiza a análise de impacto das modificações, a aprovação ou rejeição da análise de impacto e a modificação nos artefatos de requisitos, caso a análise de impacto seja aprovada. O modelo proposto por Brito (2006) também possui uma atividade específica para o gerenciamento de requisitos que serve para gerenciar as dependências através de uma matriz de rastreabilidade, analisar o impacto das mudanças e realizar de fato as alterações.

Nos modelos de processo por Lopes (2004) e por Brito (2006), apesar de os autores sugerirem atividades para avaliar o impacto de uma mudança, bem pouco se detalhou sobre como esta atividade deveria ser executada, por exemplo, que tipo de informação deveria estar contido no artefato resultante da análise de impacto. O estudo etnográfico identifica problemas relacionados à forma como o resultado da análise de impacto é disponibilizado ao cliente, e apóiam a importância de existirem atividades relacionadas a isto em um processo de ER. Diante de uma modificação de requisitos, a empresa Alpha disponibiliza um artefato de análise de impactos ao cliente, entretanto o estudo etnográfico indicou que o cliente não está satisfeito com este artefato, como mostra a seção 4.3.2.1, pois não consegue compreender quais os reais impactos resultantes da modificação. Mais do que isto, o cliente necessita de recursos visuais, por exemplo, uma árvore de dependências, para identificar onde a

modificação no requisito irá impactar em termos de telas ou mesmo no modelo de dados do sistema.

Como normalmente não existem reuniões presenciais formais cada vez que uma mudança de requisito é solicitada, a análise de impacto ocorre à distância e seu principal artefato de saída é confeccionado no pólo de desenvolvimento, isto é, sem a presença do cliente. Em resumo, os resultados deste trabalho sugerem que, de fato, a análise de impacto é um aspecto importante para a gerência de requisitos em projetos distribuídos, mas, mais importante, que os resultados desta análise devem ser detalhados e disponibilizados de forma adequada aos através de artefatos, métodos ou ferramentas que sejam ricos o suficiente ao ponto de assinalar de forma clara quais foram os reais impactos da mudança de requisitos e que, portanto, consigam diminuir a distância física existente entre os envolvidos no processo.

#### **5.4.3 Representação visual dos requisitos**

Mullick e colegas (2006) propuseram um guia de recomendações para a ER distribuída, incluindo práticas e ferramentas: wikis, fóruns e especificações de requisitos baseadas menos em diagramas UML e mais em descrições textuais – estes últimos por terem se mostrado pouco intuitivos para auxiliarem os desenvolvedores a entenderem o domínio do negócio e desenvolverem as funcionalidades do sistema.

Este estudo etnográfico, no entanto, sugere que os clientes são a favor de representações visuais (por acaso, diagramas UML). A análise dos dados mostrou que os diagramas auxiliam bastante na condução das reuniões de elicitação e validação de requisitos quando ilustram o fluxo de trabalho do cliente e asseguram que clientes e analistas possuem o mesmo entendimento do negócio que está sendo especificado (seção 4.4.1.7). Da mesma forma, os diagramas auxiliam o repasse dos requisitos para o restante da equipe que não pôde ouvir o cliente, o que diminui que os requisitos sejam compreendidos erroneamente e se transformem em funcionalidades incorretas.

Estes resultados estão alinhados com a Abordagem Unificada, proposta em (Berenbach e Gall, 2006), que estende a UML para suportar a integração entre características do sistema, casos de uso, requisitos funcionais e não-funcionais e riscos em um único modelo UML. Isto ocorre através da criação de estereótipos para representar, por exemplo, mitigação de riscos, impactos, restrições, relações de causa-efeito, etc.

## 5.5 Oportunidades de melhoria em DDS

Alguns pontos observados no estudo etnográfico não foram abordados por nenhum estudo considerado na revisão bibliográfica. Tais resultados sugerem pontos de atenção<sup>13</sup> que podem ser estudados com mais rigor em trabalhos futuros, não apenas na área de Engenharia de Requisitos. São eles:

### **Administradores de dados (ADs) em pouca quantidade, distantes geograficamente do pólo de desenvolvimento e com tarefas descoordenadas dos desenvolvedores.**

Grinter e colegas (1999) investigaram como o trabalho pode ser coordenado em contextos de desenvolvimento distribuído de software e propuseram quatro modelos de coordenação para DDS. Estes modelos estão de acordo com: áreas de habilidades, estrutura do produto, passos do processo e customização. No primeiro modelo, os especialistas de uma área específica do desenvolvimento do produto estão localizados em um único lugar. No segundo, o desenvolvimento do produto é distribuído em diversos lugares, normalmente orientado pela arquitetura do mesmo, isto é, cada componente é desenvolvido em um lugar diferente. Já no terceiro modelo, o trabalho é distribuído de acordo com os passos do processo, por exemplo, tarefas de requisitos, desenvolvimento e testes são realizadas em lugares diferentes. Finalmente, no último modelo o núcleo do código é desenvolvido em um lugar, enquanto que outros lugares se responsabilizam pela customização do código de acordo com as necessidades específicas de cada cliente.

O estudo etnográfico mostrou que a forma como os ADs estão distribuídos se ajusta ao primeiro modelo. Este modelo cita como uma de suas desvantagens a dificuldade em gerenciar projetos distribuídos em vários locais. Entretanto, a análise dos dados identificou outra desvantagem: esta configuração resulta em ADs sobrecarregados, que não dominam o negócio do cliente e que não têm tempo para atender as demandas do pólo de desenvolvimento. Por isso, os ADs são considerados um dos gargalos do processo de desenvolvimento da empresa Alpha. Uma sugestão seria aumentar o número de ADs e distribuí-los entre os pólos de desenvolvimento da empresa Alpha, para que trabalhassem junto aos desenvolvedores. Se a co-localização dos ADs não fosse possível, as demandas dos

---

<sup>13</sup> Alguns pontos serão enriquecidos com trechos de entrevistas e de notas de campo. O capítulo anterior não faz referência a estes pontos, pois os mesmos refletem sugestões baseadas na opinião da autora, o que poderia descaracterizar a natureza etnográfica do estudo que enfatiza o ponto de vista dos informantes.

ADs deveriam ser organizadas em tarefas planejadas de forma conjunta com as tarefas dos demais analistas.

### **Atendimento de demandas muito longas.**

O processo seguido pela empresa Alpha prevê que a homologação dos requisitos ocorra apenas ao fim de uma demanda. Entretanto, a demanda é quebrada em iterações e ao fim de cada iteração um conjunto de requisitos é desenvolvido e, teoricamente, estaria pronto para ser homologado. Como as demandas são muito longas, corre-se o risco de, ao fim de uma demanda, os requisitos das primeiras iterações terem sofrido alteração em virtude da mudança de algum normativo do cliente, o que é muito comum de acontecer com o cliente da empresa Alpha. Esta situação causa um desconforto geral, pois o cliente pode não homologar toda uma demanda porque as funcionalidades das primeiras iterações já não atendem mais as suas necessidades. Para isto, sugere-se a adoção de demandas curtas e com escopo reduzido. Quando as demandas e o escopo são menores, a equipe consegue gerenciá-los melhor: as homologações ocorrerão com mais frequência, o negócio provavelmente sofrerá menos mudanças e o desenvolvimento não será prejudicado ao ponto de ter que ser paralisado para que as tarefas da gerência de requisitos sejam executadas.

### **Realizar a elicitação de requisitos no pólo de desenvolvimento.**

Para casos em que o cliente é a parte distribuída, como na empresa Alpha, torna-se bastante interessante experimentar realizar a elicitação junto à equipe de desenvolvimento ao invés de deslocar apenas um analista para o contexto do cliente, pois muito mais pessoas ouvirão e poderão sugerir soluções diversas para as necessidades dos clientes. Quando a elicitação de requisitos é realizada na cidade do cliente, com o deslocamento de apenas um analista, o entendimento e, futuramente, o desenvolvimento do requisito são severamente comprometidos. Além disso, o analista deslocado, por ser o único a entrar em contato direto com o cliente, concentra conhecimento e torna-se sobrecarregado ao voltar ao pólo de desenvolvimento uma vez que será a principal fonte de conhecimento dos colegas.

### **Realizar a homologação no pólo de desenvolvimento.**

Quando a homologação ocorre no contexto do cliente, corre-se o risco de o cliente não estar totalmente comprometido com a atividade, portanto o ideal é que o cliente seja retirado totalmente de seu ambiente de trabalho e seja levado à cidade dos desenvolvedores, o que facilita para que ele esteja comprometido com as tarefas que deve desempenhar. Por outro

lado, quando a homologação ocorre de forma presencial, clientes e analistas trabalham com o objetivo comum de entregar as funcionalidades com êxito, portanto executar esta etapa com excelência interfere de maneira positiva na relação de confiança e no sentimento de equipe entre clientes e analistas (aspectos fundamentais em contextos de DDS), pois representa o fechamento de um ciclo com sucesso: a entrega de um conjunto de funcionalidades que atende as necessidades do cliente. Isto aprimora o relacionamento interpessoal dos clientes com a equipe de desenvolvimento e influencia indiretamente as tarefas de requisitos que correspondem às atividades que mais dependem do contato pessoal com o cliente.

### **Adoção de pontos de controle como pré-homologações.**

Embora a homologação não esteja diretamente relacionada com os requisitos, trata-se de uma atividade que resulta das atividades de requisitos e influencia na relação entre os clientes e os analistas. Os dados coletados mostram que alguns informantes atribuem o insucesso de uma não-homologação à ocorrência de testes de baixa qualidade.

O trecho abaixo foi retirado da entrevista com um líder de projeto. Ele fala sobre uma ocasião em que a homologação foi complicada em virtude da baixa qualidade dos testes. Os dados também mostram uma associação entre a entrada de novos analistas e a baixa qualidade da tarefa por eles desempenhada. É interessante perceber como uma questão aparentemente inofensiva no início do fluxo (ausência de uma metodologia para “receber” funcionários recém-contratados) pode adquirir proporções maiores e colocar todo um projeto em risco de não-homologação, por exemplo:

*[ent] “Recentemente nós tivemos problemas de homologação por conta de baixa qualidade nos testes. Chegou a acontecer isso em um projeto que não foi bem testado, eu acredito que, porque houve uma grande renovação do quadro da empresa Alpha. Foram chamadas muitas pessoas do concurso e muitas pessoas foram embora também. (...) Ficou meio que pela metade assim a equipe, então chegaram pessoas novas que ainda estavam se ambientando. E talvez por isso, a qualidade dos testes não tenha sido tão boa. Muitos erros passaram e chegaram até a fase de homologação e gerou uma certa reclamação no cliente. E com razão.”*

Os pontos de controle poderiam ser considerados como uma “pré-homologação” uma vez que disponibilizam de forma gradativa algumas funcionalidades para o cliente testar. Esta prática permite a identificação prematura de funcionalidades que tenham sido desenvolvidas de forma incorreta, logo permitem que tanto o código quanto os requisitos sejam corrigidos antes de irem para a homologação. Isto contribui para que uma homologação seja bem-sucedida e para melhorar o sentimento de equipe e a relação de confiança entre os

*stakeholders*. Conseqüentemente, uma homologação bem-sucedida influencia positivamente em todo o processo de desenvolvimento, sobretudo nas atividades da Engenharia de Requisitos que são as que mais requerem contato com o cliente e as que mais dependem de um bom relacionamento entre os *stakeholders*, especialmente quando estes indivíduos estão separados geograficamente.

### **Realizar *workshops* de requisitos.**

O capítulo 4 (seção 4.3.2.4) mostrou que a falta de percepção do contexto remoto pode diminuir, em longo prazo, o nível de confiança e o sentimento de equipe que devem existir entre os lados – e que já estão suficientemente comprometidos pela distância. Uma proposta, já sugerida anteriormente (seção 5.2.1.2) seria a adoção de um PSEE para auxiliar tanto os clientes quanto o coordenador de negócio no acompanhamento das tarefas. Outra recomendação, sugerida pelo cliente próprio cliente seria a realização de *workshops* anuais:

*“Se nós tivéssemos uma metodologia que permitisse a gente fazer um grande encontro por ano de planejamento do sistema, isso seria assim uma economia...”*.

Esta sugestão foi associada ao conceito de “*workshop* de requisitos”. O *workshop* de requisitos se assemelha a uma sessão de *brainstorming* e entre seus participantes devem estar tanto analistas quanto clientes. O principal objetivo de um *workshop* de requisitos, segundo o RUP, é promover o encontro entre a equipe de projeto e os envolvidos nele, elaborar uma “lista de desejos” dos envolvidos no projeto e priorizar requisitos.

No caso particular da empresa Alpha, o *workshop* de requisitos poderia servir para:

- a) aproximar analistas e clientes, propiciando a criação de laços de amizade e, conseqüentemente, uma relação de confiança entre eles;
- b) ilustrar de forma didática quais as etapas do processo de desenvolvimento, detalhando as atividades previstas e os artefatos produzidos em cada etapa;
- c) apresentar o processo de trabalho dos clientes, isto é, a forma como trabalham, suas atividades e suas necessidades, e;
- d) delimitar o escopo do projeto a partir da definição das soluções ou funcionalidades que atenderiam as necessidades dos clientes.

Alcançar estes objetivos ajudaria a compartilhar o conhecimento sobre as práticas e as políticas organizacionais dos *stakeholders* envolvidos e evitariam que quaisquer más impressões surgissem entre eles. Pode-se pensar em adotar, ao invés de *workshops* anuais, um

workshop por demanda ou por um grupo de demandas. Além disso, sugere-se despende algum esforço no momento de analisar quais analistas deslocar para o workshop (diante da questão orçamentária da empresa Alpha), ou ainda considerar a possibilidade de o workshop ser realizado no próprio pólo de desenvolvimento.

### **Formalização do papel de *knowledge broker*.**

Os problemas de falta de *awareness* presentes em contextos de DDS (e evidenciados neste estudo) podem ser amenizados pela criação e formalização do papel de *knowledge broker* dentro da equipe de desenvolvimento. O *knowledge broker* (Brössler, 1999) concentra uma grande quantidade de conhecimento consigo, porém sua função é exatamente disseminar este conhecimento dentro da organização de modo que os desenvolvedores.

No caso da empresa Alpha, um *knowledge broker* pode ser útil para disseminar aos demais membros da equipe de desenvolvimento como o contexto remoto (do cliente) funciona de modo que os demais analistas aprendam, por exemplo, a elaborar documentos de especificação de requisitos e de regras de negócio mais condizentes com a realidade de seus clientes. A análise dos dados coletados indica que os analistas deslocados para a cidade do cliente já assumem informalmente o papel do *broker*. Entretanto, suas atribuições oficiais continuam sendo as de um analista de sistemas, ou seja, seu tempo é totalmente alocado para tarefas relacionadas a requisitos, codificação e testes. Isto causa o problema das interrupções constantes ao analista deslocado que precisa parar suas atividades várias vezes por dia para prover informações a outros colegas, como visto no capítulo 4 (seção 4.3.2.2).

Tradicionalmente, o *knowledge broker* deve ser designado unicamente para disseminar conhecimento, isto é, participar das reuniões de monitoramento de projeto de todas as equipes da empresa, indicar especialistas em um determinado assunto quando solicitado, alimentar a base de conhecimento da empresa, etc. Sabe-se, contudo, que como a equipe de desenvolvimento é pequena para a carga de trabalho que a ela é atribuída, uma ideia seria que um analista atuasse como *knowledge broker* durante metade de seu expediente para que na outra metade estivesse livre para realizar tarefas de desenvolvimento junto aos demais, o que caracterizaria a formalização do papel dentro da empresa.

## 6 CONSIDERAÇÕES FINAIS

Este trabalho apresentou um estudo etnográfico na empresa Alpha. Esta empresa realiza atividades de Engenharia de Requisitos de maneira distribuída, uma vez que seu principal cliente está situado em outra região do país. Atividades como elicitação, especificação e validação de requisitos são consideravelmente afetadas em virtude da distância existente entre a equipe de desenvolvimento e o cliente.

O método etnográfico foi escolhido por se tratar do único método empírico capaz de permitir que o pesquisador esteja em campo para entender as práticas de uma cultura a partir do ponto de vista dos informantes. A adoção deste método visa eliminar uma das maiores limitações dos estudos empíricos levantados durante a revisão bibliográfica: a coleta de dados de apenas um dos contextos de trabalho envolvidos no projeto. Esta limitação pode comprometer a validade dos resultados obtidos já que as atividades de Engenharia de Requisitos dependem sempre da interação entre as duas partes: clientes e/ou usuários e equipe de desenvolvimento. Em projetos distribuídos, as atividades do processo de Engenharia de Requisitos são severamente impactadas, sobretudo pela distância existente entre os *stakeholders*.

O principal objetivo deste estudo etnográfico foi identificar os problemas enfrentados por engenheiros de software envolvidos em projetos de Desenvolvimento Distribuído de Software na empresa Alpha. Além das dificuldades, este estudo também identificou as “boas práticas” de trabalho realizadas pela empresa, isto é, quais estratégias eram utilizadas pelas equipes para diminuir os efeitos causados pela distância.

Finalmente, este trabalho confronta os resultados do estudo etnográfico com os trabalhos elencados durante a pesquisa bibliográfica, com o intuito de apontar suas limitações e sugerir

avanços nos trabalhos pesquisados sobre a Engenharia de Requisitos em contextos distribuídos.

As próximas seções apresentarão as contribuições do trabalho bem como futuros estudos que podem ser iniciados a partir dos resultados obtidos nesta dissertação.

## 6.1 Principais contribuições

As principais contribuições desta dissertação são:

- Realização de um estudo etnográfico em um projeto real de Desenvolvimento Distribuído de Software, envolvendo coleta de dados a partir de entrevistas não e semi-estruturadas e observações diretas e análise destes dados utilizando técnicas da *Grounded Theory*, como codificação aberta, codificação axial e codificação seletiva;
- Identificação dos problemas enfrentados e das boas práticas utilizadas por profissionais de uma empresa que realiza Engenharia de Requisitos em um contexto de Desenvolvimento Distribuído de Software;
- Criação de discussões a partir dos problemas e boas práticas identificadas durante a análise dos dados e comparação destas teorias com os trabalhos correlatos que foram propostos pela academia.
- Avanço do conhecimento em métodos qualitativos de pesquisa na Engenharia de Requisitos em contextos de Desenvolvimento Distribuído de Software, e
- Publicação de artigos em conferências com os resultados preliminares do trabalho (Lemos e de Souza, 2008) (Lemos e de Souza, 2008).

## 6.2 Limitações do estudo

Uma das principais restrições do estudo etnográfico apresentado é a incapacidade de generalização dos resultados obtidos uma vez que apenas uma equipe de apenas uma empresa foi intensamente estudada. Entretanto, deve-se enfatizar que o tipo do método de pesquisa utilizado – a etnografia – possui natureza qualitativa e especialmente exploratória e, portanto, não se propõe a provar hipóteses, mas compreender o contexto observado para, só então, discutir a ampla pergunta de pesquisa que o trabalho deseja responder.

Outra limitação deste estudo etnográfico é o viés possivelmente introduzido pela influência da autora na análise dos dados. Assume-se que esta influência existe, pois, todo estudo empírico possui algum tipo de viés.

Outras limitações do trabalho transcorrem dos problemas enfrentados durante a coleta de dados, como a dificuldade em realizar observações no contexto da empresa Alpha (dado o desconforto visível que isto causava aos analistas) e o pouco tempo que os informantes podiam dedicar às entrevistas.

### **6.3 Trabalhos futuros**

Como possíveis estudos futuros, sugerem-se:

- A realização de estudos etnográficos em outras empresas com o intuito de comparar os resultados encontrados neste estudo com os encontrados nos demais;
- A realização de estudos quantitativos visando avaliar quantitativamente as teorias criadas neste estudo etnográfico. Por exemplo, um dos resultados deste estudo é que a elicitação de requisitos deveria ser realizada no pólo de desenvolvimento. Um estudo quantitativo poderia ser conduzido para investigar se esta teoria criada é válida;
- Inclusão das boas práticas propostas na seção 5.6 deste trabalho aos modelos de processo citados objetivando a evolução destes modelos;
- Elaboração e avaliação de um modelo de processo adequado ao contexto ilustrado neste trabalho, em que os clientes estão distantes de todo o resto da equipe de desenvolvimento;
- Investigação sobre o que exatamente os clientes precisariam ou gostariam de saber sobre o processo da empresa Alpha sem que isto expusesse de maneira inadequada a mesma; e
- Finalmente, um estudo sobre a possibilidade de implantação do uso de um PSEE na empresa Alpha dada a necessidade dos clientes em estarem cientes sobre partes do processo de desenvolvimento da empresa;

## REFERÊNCIAS BIBLIOGRÁFICAS

- Anderson, B. Work, Ethnography and System Design. In A. Kent e J.G. Williams (eds.): The Encyclopedia of Microcomputing. New York: Marcel Dekker, pp. 159-183, 1997.
- Aranda, G., Vizcaíno, A., Cechich, A. e Piattini, M. Technology Selection to Improve Global Collaboration. Proc. of the IEEE ICGSE. Florianópolis, Brasil, 2006.
- Audy, J. e Prikladnicki, R. Desenvolvimento Distribuído de Software: Desenvolvimento de Software com Equipes Distribuídas, Série Livros Didáticos Campus-SBC, Editora Campus/Elsevier, 2007.
- Berenbach, B. e Gall, M.: Toward a Unified Model for Requirements Engineering. International Conference on Global Software Engineering (ICGSE 2006). IEEE Computer Society, Costão do Santinho, Florianópolis, Brazil (2006).
- Blomberg, J. Social aspects of operability: Ethnography of photocopiers. Paper presented at the meeting of the American Association for the Advancement of Science, Boston, 1998.
- Blomberg, J., Giacomi, J., Mosher, A., e Swenton-Wall, P. Ethnographic Field Methods and Their Relation to Design. In: Participatory Design: Principles and Practices. Editado by Douglas Dchuler and Aki Namioka. Lawrence Erlbaum Associates Inc., USA, 1993.
- Boehm, B., Grünbacher, P. e Briggs, R., Developing Groupware for Requirements Negotiation: Lessons Learned, IEEE Software, v.18 n.3, p.46-55, May 2001.
- Brito, R. A. e Vasconcelos, Lins M. A. Integrando Groupware e Gerenciamento de Requisitos no Suporte à Engenharia de Requisitos Distribuída. Anais do WER06 - Workshop em Engenharia de Requisitos, Rio de Janeiro, RJ, Brasil, Julho 13-14, 2006, pp 84-92.
- Brito, Regiane Andrade. Suporte à engenharia de requisitos no desenvolvimento distribuído de software. Recife, 2006. Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIN. Ciência da Computação, 2006.

- Brössler, P. Knowledge management at a software house: An Experience report, in Learning Software Organizations: Methodology and applications; Proc. 11th SEKE '99, Kaiserslautern, Germany, June 16-19, 1999, pp. 163-170.
- Calefato, F. e Lanubile, F. Using The Econference Tool For Synchronous Distributed Requirements Workshops. Proc. of the 1st International Workshop on Distributed Software Development (DiSD 2005), Paris, France, Austrian Computer Society, August, 2005.
- Carmel, E. e Agarwal, R. Tactical Approaches for Alleviating Distance in Global Software Development. IEEE Software, March/April, 2001, pp.22-29.
- Carmel, E. Global Software Teams: A Framework for Managerial Problems and Solutions. Global Information Technology and Electronic Commerce: Issues for the New Millennium, Ivy League Publishing, Marietta GA, 2000.
- Cheng, B. e Atlee, J. Research Directions in Requirements Engineering. Proc. of the Conference on the Future of Software Engineering. p.285-303. May, 2007.
- Curtis, B., Krasner, H., Iscoe, N. A Field Study of the Software Design Process for Large Systems. Magazine: Communications of ACM, Volume 31 Issue 11. Novembro, 1988. Páginas 1268-1287. Publisher: ACM Nova York, NY, EUA, 1988.
- Damian, D. e Zowghi, D. Requirements Engineering challenges in multi-site software development organizations. Requirements Engineering Journal, 8, 149-160, 2003.
- Damian, D. e Zowghi, D. The impact of stakeholders' geographical distribution on requirements engineering in a multi-site development organization. Proc. of the 10th IEEE Int'l Conference on Requirements Engineering (RE'02). p.319-328. Essen, Germany. 2002.
- Damian, D., Eberlein, A., Shaw, M. e Gaines, B. Using Different Communication Media in Requirements Negotiation. IEEE Software, v.17 n.3, p.28-36, May 2000.
- Damian, D., Lanubile, F. e Mallardo, D. The role of asynchronous discussions in increasing the effectiveness of remote synchronous requirements negotiations. Proc. of International Conference on Software Engineering, Shanghai, May, 2006.
- Damian, D., Shaw, M., Eberlein, A., Gaines, B. e Woodward, B., An Empirical Study of Facilitation of Computer-Mediated Distributed Requirements Negotiations, Proceedings of the Fifth IEEE International Symposium on Requirements Engineering, p.128, August 27-31, 2001.

- Daniela Damian , Deependra Moitra, Guest Editors' Introduction: Global Software Development: How Far Have We Come?, IEEE Software, v.23 n.5, p.17-19, September 2006.
- de Souza, C., Sharp, H., Dittrich, Y., Singer, J. Cooperative and Human Aspects of Software Engineering (CHASE 2009). In Proceedings of the 31st International Conference on Software Engineering, Vancouver, Canada, ICSE 2009.
- De Souza, K., Awazu, Y. e Baloh, P. Managing Knowledge in Global Software Development Efforts: Issues and Practices, IEEE Software, v.23 n.5, p.30-37, September, 2006.
- Dewalt, B. e Dewalt, K. Participant Observation: A Guide for Fieldworkers. Blue Ridge Summit: AltaMira Press, 2002.
- Dourish, P. Where the Action Is: The Foundations of Embodied Interaction. 2004.
- Ebert, C. Global Software Engineering. IEEE ReadyNote (e-Book), IEEE Computer Society, Los Alamitos, USA, 2006.
- eGroupware, <http://www.egroupware.org/Home?lang=pt-br>, 2000.
- Espinosa, J., Nan, N. e Carmel, E. Do Gradations of Time Zone Separation Make a Difference in Performance? A First Laboratory Study. Proc. of Int'l Conference on Global Software Engineering, Munich, Germany. IEEE, August, 2007.
- Finkelstein, A. Requirements Engineering: an overview. 2<sup>nd</sup> Asia-Pacific Software Engineering Conference (APSEC'93), Tokyo, Japan, 1993.
- Glaser, B. Strauss, A. The discovery of grounded Theory. Aldine de Gruyter, NY, 1967.
- Grinter, R., Herbsleb, J., et al. (1999). The Geography of Coordination: Dealing with Distance in R&D Work. ACM Conference on Supporting Group Work (GROUP '99), Phoenix, AZ, ACM Press.
- Gruenbacher, P. Collaborative Requirements Negotiation with EasyWinWin. Proc. of the Second International Workshop on the Requirements Engineering Process - Innovative Techniques, Models, Tools to support the RE Process, 2000.
- Gruhn, V. Process-Centered Software Engineering Environments, A Brief History and Future Challenges. Journal Annals of Software Engineering Volume 14 Issue 1-4, December 2002.

- Heath, C. e Luff, P. (1991) Collaborative Activity and Technological Design: Task coordination in the London Underground Control Rooms. In *Proceedings of ECSCW'91*, páginas 65-80. Kluwer Academic Publishers.
- Herbsleb, J.D. e Moitra, D. Global software development, *IEEE Software*, March/April, 2001, pp.16-20.
- Herlea, D. e Greenberg, S., Using a Groupware Space for Distributed Requirements Engineering. Proc. of the Seventh International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, 1998.
- Hughes, J., King, V., Rodden, T. e Andersen, H. (1994) Moving out from the control room: ethnography in system design. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, páginas 429-439, ACM Press.
- Illes-Seifert, T., Herrmann, A., Geisser, M. e Hildenbrand, T. The Challenges of Distributed Software Engineering and Requirements Engineering: Results of an Online Survey. Proc. of GREW'07: 1st International Global Requirements Engineering Workshop, Munich, Germany, August, 2007.
- Jorgensen, D. L. Participant Observation: A Methodology for Human Studies. SAGE Publications, Thousand Oaks, CA, 1989.
- Kotlarsky, J. e Oshiri, I. (2005). Social ties, knowledge sharing and succes.ful collaboration in globally distributed system development projects, *European Journal of Information Systems*, 14, páginas 37-48.
- Kraut, R., Egidio, C. et al. Patterns of Contact and Communication in Scientific Research Collaborations. In: *Intellectual Teamwork: Social and Technological Foundations of Cooperative Work*. J. Galegher, C. Egidio e R. Kraut, Lawrence Erlbaum: 149-172, 1990.
- Kraut, R., Streeter, L. 1995. Coordination in software development. Magazine: *Communications of the ACM*, Volume 38 Issue 3. Março, 1995. Páginas 69-81. Publisher: ACM Nova York, NY, EUA.
- Lemos, A. e de Souza, C. Desafios da Gerência de Conhecimento no Desenvolvimento de Software: Resultados de um Estudo Etnográfico. SBSC 2008a: 77-87.
- Lemos, A. e de Souza, C. Engenharia de requisitos em ambientes distribuídos de desenvolvimento de software: resultados preliminares de um estudo etnográfico. WER 2008b.

- Lima, A., Costa, A., França, B., Reis, C. e Reis, R. Gerência Flexível de Processos de Software com o Ambiente WebAPSEE. 19º. Simpósio Brasileiro de Engenharia de Software – Sessão de Ferramentas, Outubro, 2006.
- Lloyd, W. J., Tools and Techniques for Effective Distributed Requirements Engineering: An Empirical Study. Masters Thesis, Virginia Tech, 2001. Disponível em: <http://scholar.lib.vt.edu/theses/available/etd-07262001-110924/>
- Lloyd, W., Rosson, M. e Arthur, J. Effectiveness of Elicitation Techniques in Distributed Requirements Engineering. Proc. of the 10th Anniversary IEEE Joint Int'l Conference on Requirements Engineering, p.311-318, September 09-13, 2002.
- Lopes, L., Prikladnicki, R., Audy, J. e Majdenbaum, A. Requirements Specification in Distributed Software Development - A Process Proposal. Proc. of the 38th Hawaii International Conference on System Sciences - HICSS. Hawaii, USA, 2005.
- Lopes, Leandro Teixeira. Um modelo de processo de engenharia de requisitos para ambientes de desenvolvimento distribuído de software. Porto Alegre, 2004. Dissertação (Mestrado) – Fac. de Informática, PUCRS, 2004.
- Macaulay L., Requirements Capture as a Cooperative Activity, Proceedings of IEEE Symposium on Requirements Engineering, IEEE Computer Society Press 174-18 I, 1993.
- MAXqda2, <http://www.maxqda.com/>, 1995.
- McGrath, J. E. Methodology Matters: Doing Research in the Behavioral and Social Sciences, 1994.
- Medeiros, Leonardo Melo de. Proposta de Processo de Documentação e Validação dos Requisitos para Equipes de Desenvolvimento Distribuído de Software. Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIN. Ciência da Computação, 2007.
- Millen, D. Rapid ethnography: time deepening strategies for HCI field research. In *Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques*, p.280-286, August 17-19, 2000, New York City, New York, United States, 2000.
- Mockus, A. e Herbsleb, J. Challenges of global software development, Proc. of the 17th Software Metrics Symposium, London, IEEE Computer Society, April – 2001, pp.182-184.

- Mullick, N., Bass, N., Houda, Z., Paulish, P. e Cataldo, P. Siemens Global Studio Project: Experiences Adopting an Integrated GSD Infrastructure. Proc. of the IEEE ICGSE. Florianópolis, Brasil, 2006.
- Nardi, B. The use of ethnographic methods in design and evaluation. In Handbook of Human-Computer Interaction, M. Helander, T. K. Landauer, e P. Prabhu, Eds., pp. 361-366. Elsevier Science, 1997.
- Nuseibeh & Easterbrook. Requirements Engineering: A Roadmap – Nuseibeh, Easterbrook, 2000.
- Pfleegler, Shari Lawrence. Engenharia de Software – teoria e prática, Prentice Hall, 2004.
- Pressman, R. S. “Engenharia de Software”. 6ª Edição, Editora McGraw-Hill, 2004.
- Prikladnicki, R. e Audy, J. Requirements Engineering in Global Software Development: Preliminary Findings from a Case Study in a SW-CMM context. Proc. of the 5th SIMPROS – Simpósio Internacional de Melhoria de Processo de Software, Pernambuco, 2003.
- Randall, D., Harper, R. e Rouncefield, M. Livro: Fieldwork for Design: Theory and Practice. Capítulo: Ethnography and How to Do It. Publisher: Springer London, 2007.
- Rose, A., Shneiderman, B. e Plaisant, C. An applied ethnographic method for redesigning user interfaces. In Proceedings of the conference on Designing interactive systems: processes, practices, methods, & techniques, p.115-122, August 23-25, 1995, Ann Arbor, Michigan, United States.
- Schmidt, K. e Carstensen, P. Bridging the Gap: Requirements Analysis for System Design, COMIC Working Paper, COMIC-RIS0-2-2, Available from Department of Computing Lancaster University, Lancaster LA1 4YR, 1993.
- Shull, F., Singer, J. e Sjøberg, D. Guide to Advanced Empirical Software Engineering, Springer-Verlag New York, Inc., Secaucus, NJ, 2007.
- Sommerville, I. Software Engineering. 7 ed. Addison-Wesley, 2004.
- Sommerville, I., Rodden, T., Sawyer, P., Bentley, R. e Twidale, M. Integrating ethnography into the requirements engineering process. Proc. of IEEE International Symposium on Requirements Engineering, San Diego, CA, USA, 1993.
- Sommerville, I., Rodden, T., Sawyer, P., Bentley, R., e Twidale, M. (1993). Integrating Ethnography into the Requirements Engineering Process. In Proceedings of the IEEE

- International Symposium on Requirements Engineering. January 4-6, 1993, San Diego, California, IEEE Computer Society Press, Los Alamitos, California, 165-173.
- Strauss, A e Corbin, J. Basics of Qualitative Research: Grounded Theory Procedures and Techniques, Sage Publications, Newbury Park, CA.), 1990.
- Suchman, L. Plans and situated actions: the problem of human-machine communication. Cambridge, Cambridge University Press, 1987.
- TeamWave Software Ltd., <http://www.teamwave.com/>.
- Togneri, D.F., Falbo, R.A. e Menezes, C.S., Supporting cooperative requirements engineering with an automated tool. In: Proceedings of the Workshop on Requirements Engineering (WER), pp. 240-254. (2002)
- Wild, C. J. e Seber, G. Chance Encounters: A First Course in Data Analysis and Inference, John Wiley & Sons, 1999.
- Yin, R. Case Study Research, Design and Methods, 3rd edition, Thousand Oaks, CA, Sage Publications, 2003.
- Zowghi, D. 2002. Does global software development need a different requirements engineering process? In Proceedings of the International Workshop on Global Software Development, Orlando, Florida, (In conjunction with the Int'l Conference on Software Engineering, ICSE 2002).

## APÊNDICE I – GUIAS DE ENTREVISTAS

### Entrevista realizada com o cliente:

#### Grand Tour: como a distância afeta suas atividades?

1. Como (por que) você foi escolhida para ser **fornecedor de requisitos** do SISCAD?
2. Você é **usuário** do SISCAD?
3. Quem mais é o **usuário**? Você tem **contato** com estes usuários?

#### Elicitação de requisitos:

1. Como acontecem as reuniões de **elicitación de requisitos**?
2. O que você achou do **uso de protótipos** durante a reunião de elicitação de requisitos?
3. Em que momento você acha necessário rabiscar interfaces no papel? (**paper prototyping**)
4. Como você acha que a **análise de requisitos funciona melhor**: quando o **analista** sai de Belém e vai à Brasília ou quando os **clientes** se deslocam à Belém e entram em contato com a equipe de desenvolvimento? Por quê? (*pedir para ele falar sobre as experiências de idas à Belém e vindas à BSB*).

#### Documentação de requisitos:

1. Você tem acesso aos documentos elaborados pela empresa Alpha? De que forma?
  - a. O que é o QoS RF? Qual a relação dele com o modelo de processo da empresa Alpha? Espelhamento um do outro?
2. Quais documentos são enviados do empresa Alpha para o cliente?
  - a. O que você acha da especificação de requisitos? (é suficiente, clara,

intuitiva?) Se pudesse, o que mudaria nela?

- b. O que você acha do documento de regras de negócio? Qual a diferença dele para a especificação de requisitos? Se pudesse, mudaria algo nele? O quê?
  - c. O que você acha dos diagramas? (são mais intuitivos?)
3. Você participa da documentação (especificação de requisitos, regras de negócio, glossários, etc)?
  4. Como é a sua participação?
    - a. Alterou/revisou algum documento? Qual? Em sua opinião, esta prática **ajudou ou atrapalhou** o processo de análise de requisitos?
    - b. Depois de alterar/revisar, o documento é enviado de volta à empresa Alpha? Como ele é enviado? O que acontece em seguida? (*A empresa Alpha acata as sugestões dadas pelo cliente?*)

#### **Validação de requisitos:**

1. Você já participou de alguma reunião de fechamento?
2. Como ela funciona?
 

Obs.: Se ela falar sobre conflitos, desentendimentos, discussões, perguntar: em sua opinião, qual é a **causa** destes conflitos? (analistas não entendem os requisitos? Requisitos estão mal especificados? Mal escritos? **O documento de especificação é confuso?**).
3. O que é **mais comum** de acontecer: o **analista** ir à BSB ou o **cliente** vir à Belém? Em sua opinião, o que é **mais vantajoso**? Por quê?
4. Antes o **fechamento** dos requisitos era realizado à distância:
  - a. **Como** isto acontecia?
  - b. Qual a forma de **comprometimento** do cliente para com a empresa Alpha?

#### **Gerência de requisitos:**

1. Como funcionam as **demandas**?
  - a. Quem as cria?
  - b. Quem modifica a prioridade das demandas?
  - c. A prioridade das demandas é alterada com muita frequência? Por quê?

2. Existe alteração / adição de requisitos com frequência? Por quê?
  - a. O cliente entende o custo destas alterações em termos de custo financeiro e cronograma?
  - b. A equipe envia algum documento para os clientes ficarem cientes do que está sendo alterado ou quais artefatos serão impactados?
  - c. Qual documento é esse? (*pedir para descrever*).

**Informações gerais:**

1. O **cliente é informado** sobre as atividades da equipe de analistas? Há **interesse** por parte do cliente em ser informado sobre isso?  
  
*Ex.: em qual fase do processo de desenvolvimento do software os analistas estão? Saber as etapas do processo pode permitir que o cliente entenda por que uma alteração / adição de requisitos demora tanto.*
2. Em que (quais) momento (s) você **entra em contato** com a equipe da empresa Alpha?
  - a. Como ocorrem estes contatos? (Áudio-conferência? Videoconferência?)
  - b. Você acha que este contato é **satisfatório**?
    - i. Vantagens/desvantagens da áudio/videoconferência.
  - c. Se pudesse, o que você **mudaria**?
3. Você já esteve em **contato com os desenvolvedores** responsáveis pelo SISCAD? Como foi a experiência? (**vantagens, desvantagens**).
4. Como ocorre o **relato de erros/defeitos** no SISCAD? Você acha que esse método é eficaz?
5. Você participa das homologações?
6. É comum ocorrer uma **não-homologação**? Quais os **motivos** para uma não-homologação? Você poderia citar **um caso** que já tenha presenciado uma não-homologação?
7. Em sua opinião, quais os principais impactos da distância?
  - a. Atrasos?
  - b. Restrição orçamentária gera problemas? Por quê?

- c. Dedicção do cliente para a atividade é suficiente?
  - d. Ida de uma única pessoa à BSB? Soluções: ir mais pessoas? Realizar a reunião no local onde está a equipe de analistas?
8. Como você solucionaria estes problemas?
9. Você, de alguma forma, procura **amenizar** os impactos causados pela distância?  
**Como?**

### **Entrevista realizada com o coordenador de negócios:**

#### **Grand Tour: como a distância afeta as atividades de requisitos?**

#### **Contextualização:**

1. Qual é o seu **cargo** na empresa Alpha?  
*Resposta esperada: coordenador de negócios.*
2. Há quanto **tempo** você exerce este cargo?
3. Você já exerceu algum **outro cargo**?
4. No cargo atual, quais **funcionalidades** são atribuídas a você?
  - a. Qual e como é a sua relação com o **cliente** (empresa Beta)?
  - b. Qual e como é a sua relação com a **equipe** técnica?
5. *No caso de ela tocar em “mediar encontros entre analistas e clientes”: Quando é necessário realizar uma intervenção? Em que consiste uma intervenção?*
6. Com quais **sistemas** você trabalha atualmente?  
*Resposta esperada: SISCAD, SISCAC, SISCO, E-FAU.*
7. **Quem decide onde** serão as reuniões com o cliente?
8. **Quem decide qual o pólo** será responsável por atender uma determinada demanda? **Quais fatores** são levados em consideração?
  - a. Qual o pólo que já trabalha com **tributos**?
  - b. Qual o pólo que já trabalha com **imposto de renda**?
9. **Quem** normalmente **se desloca** de seu local de trabalho de origem?
  - a. Se ela tocar no assunto sobre **restrição orçamentária**, perguntar **quem** paga as passagens e diárias dos analistas (empresa Alpha?) e dos clientes

(empresa Beta?).

10. Como você acha que a **análise de requisitos funciona melhor**: quando o **analista** sai de Belém e vai à Brasília ou quando os **clientes** se deslocam à Belém e entram em contato com a equipe de desenvolvimento? Por quê?

### **Etapas de Engenharia de Requisitos:**

#### 1. Elicitação de requisitos:

- a. Sua presença é exigida na etapa de **elicitação** de requisitos? Por quê?
- b. O que você achou da elaboração de **protótipos** junto ao cliente (ida do analista à BSB)?
- c. Você sabe como a empresa Beta escolhe a pessoa responsável por informar os requisitos à empresa Alpha?

(É de acordo com a experiência da pessoa, ou seja, o mais experiente no negócio é escolhido? A pessoa que informa os requisitos também é usuária do sistema a ser desenvolvido? Em caso negativo, qual o grau de aproximação da pessoa com o usuário real?)

- i. Prestar atenção se ela vai falar sobre **centralizar dúvidas** de usuários.

#### 2. Documentação de requisitos:

- c. Você já participou alguma vez da **reunião de repasse**?
- d. Você participa da **documentação**? Os clientes participam?
- e. O que você acha de os **clientes poderem alterar** o documento de especificação de requisitos? Em sua opinião, esta prática **ajuda ou atrapalha** o processo de análise de requisitos?
- f. Já chegou a **revisar** algum documento? Em caso afirmativo – **por que** lhe foi exigido que você revisasse um documento?

#### 3. Validação de requisitos (fechamento):

- a. Como é uma reunião de **fechamento**?

Obs.: Se ela falar sobre conflitos, desentendimentos, discussões, perguntar: em sua opinião, qual é a causa destes conflitos? (analistas não entendem os requisitos? Requisitos estão mal especificados? Mal escritos? **O documento de especificação é confuso?**).

- b. Sua **presença** é exigida no **fechamento**? Por quê?

- c. Antes o **fechamento** dos requisitos era realizado à distância. Como isto acontecia?

Obs.: Procurar entender por que a validação passou a acontecer presencialmente. Em que momento a necessidade deste segundo encontro presencial foi detectada? Quem detectou?

- d. Nas reuniões de fechamento, o que é **mais comum** de acontecer? O **analista** ir à BSB ou o **cliente** vir à Belém? Em sua opinião, o que é **mais vantajoso**? Por quê?
- e. No caso de fechamento à distância, existe alguma **garantia** de que o cliente leu o documento elaborado pela empresa Alpha? Qual a forma de **comprometimento** do cliente para com a empresa Alpha? O **analista de negócios** pode atuar no sentido de garantir este comprometimento? **Como**?

#### **Outras perguntas:**

2. O cliente é **informado** sobre as atividades realizadas pela equipe? (Por exemplo, quando o analista vai à BSB e volta à Belém, o cliente sabe que o próximo passo da equipe é escrever um documento de especificação?).
- a. Em caso positivo, **como o cliente é informado** sobre o andamento do trabalho da equipe?
- b. Em caso negativo, em sua opinião, você acha que o cliente **deveria receber feedback** da empresa Alpha quanto ao andamento das tarefas? De que forma seria o ideal? (Via e-mail? Via alguma ferramenta?)
3. Como você lida com situações em que os clientes pedem **alterações (ou adição) de requisitos**? O que é a **portaria 1450**? Os clientes têm **conhecimento** desta portaria? – **Gerência de requisitos**.
4. O que é o QoS RF? Qual a relação dele com o modelo de processo da empresa Alpha?
5. Você participa das **homologações**?
6. É comum ocorrer uma **não-homologação**? Você poderia citar **um caso** que você já tenha presenciado de não-homologação? Quais os motivos para a não-homologação?
7. Pra você, quais são os principais **impactos** causados pela distância?
- a. Atrasos?
- b. Restrição orçamentária → problemas?

- c. Dedicção do cliente para a atividade?
- d. Ida de uma única pessoa à BSB? Soluções: ir mais pessoas? Realizar a reunião no local onde está a equipe de analistas?

8. Como você **solucionaria** estes problemas?

Você, de alguma forma, procura **amenizar** os impactos causados pela distância?  
**Como?**

### **Entrevista realizada com os analistas:**

#### **Grand Tour: como a distância afeta as atividades de requisitos?**

##### **1. Processo seguido pela empresa:**

- a. Como ele está caracterizado? (Cascata? Incremental e iterativo?)
- b. Quais são as etapas seguidas?
  - i. Depois da análise dos requisitos, existe alguma etapa para projeto e arquitetura do sistema?

##### **2. Elicitação dos requisitos:**

- a. Quantas reuniões ocorreram?
- b. Quando elas aconteceram?
- c. Onde? (Locais / remotas)
- d. Sobre qual sistema?
  - i. Quem decide o pólo de desenvolvimento responsável pelo software?
- e. Quem eram os participantes das reuniões?
  - i. Quantos?
  - ii. Quais eram os papéis?
- f. Quais os documentos resultantes destas reuniões?
  - i. Definição do escopo (isto é um documento? O que vem contido nele?)
  - ii. Documento de visão (O que vem contido nele? Requisitos? Casos de uso? Regras de negócio? Como você diferencia requisitos de casos de uso? E estes de regras de negócio?)

- iii. Ambos sempre são elaborados? (Mesmo que o negócio já seja conhecido pela equipe, há necessidade de definir o escopo?)
  - iv. O que é o DRS?
  - v. Qual a ordem em que esses documentos devem ser elaborados? (Primeiro DV, depois UCs, depois...?). Vocês elaboram diagramas de casos de uso? Eles são mostrados ao cliente?
- g. Negócio novo x negócio conhecido para a equipe:
- i. A natureza do negócio tem a ver com a quantidade de reuniões realizadas?
  - ii. Como você conclui que não há mais necessidade de reuniões para definição de escopo?
  - iii. Como você conclui que não há mais necessidade de reuniões para elaboração do documento de visão?
- h. Como proceder em caso de dúvidas?
- i. Telefone?
  - ii. Novo encontro presencial?
  - iii. Como escolher um ou outro? O que caracteriza a necessidade de um e de outro? (Complexidade: o que é simples e o que é complexo?)

### **3. Reuniões de repasse:**

- a. Quando elas ocorrem? (Sempre que um analista volta do cliente?)
- b. Quanto tempo em média elas demoram?
- c. Quais documentos são mostrados para os analistas de Belém? (Isto é, quais documentos começam a ser elaborados ainda em Brasília?)

### **4. Reuniões remotas x reuniões locais:**

- a. Qual a diferença entre elas?
- b. Qual a melhor? Qual a pior? Por quê?
  - i. **Remota:** O que você acha sobre a dedicação dos funcionários da empresa Beta durante a elicitación de requisitos?
  - ii. **Local:** Os desenvolvedores devem ter contato com os clientes? Por

quê? (Importância / problema no contato entre desenvolvedores e clientes). Em que momento este contato deve ocorrer?

#### **5. Fechamento de requisitos:**

- a. Quais documentos são vistos pelo cliente quando ele assina o Anexo 2? (DV, glossário, casos de uso, regras de negócio, requisitos não-funcionais, etc).
- b. Como você acha que deveria ocorrer o fechamento de requisitos? (Local ou remoto?)

Obs: (o que a última áudio representou? Foi uma forma de fechar requisitos? Algum documento resultou daquela áudio? Qual? Este documento considera o comprometimento dos demais? Como você avalia as áudios do SISCAC? Quais as ações seguintes àquela áudio? Implementação?)

#### **6. Qual a etapa posterior aos requisitos?**

- a. Sua equipe realiza etapa de projeto e arquitetura?
  - i. Em que consiste esta etapa? (Elaboração de quais diagramas? E quais documentos?)
  - ii. É fácil mostrar ao cliente que esta etapa é importante?
- b. Sua equipe costuma elaborar protótipos?
  - i. Quando e quem decide elaborar protótipos? (em que etapa?)
  - ii. Eles fazem parte da documentação oficial?
  - iii. A construção de protótipos está prevista no modelo de processo da empresa Alpha?
  - iv. Você já utilizou protótipos enquanto elicitava requisitos? Como foi a experiência?

#### **7. Quais as dificuldades criadas pelo fato de o cliente estar remoto?**

- a. Atentar para os seguintes aspectos:
  - i. O cliente (de alguma forma) está ciente das etapas em que os analistas se encontram? (monitoração do processo)
  - ii. Em sua opinião, os clientes estão satisfeitos com os documentos elaborados? (Isto é, os clientes conseguem interpretar bem os

documentos? Compreendem os documentos?)

iii. Os documentos elaborados ficam disponíveis aos clientes? De que forma?

b. Dificuldades já identificadas:

i. Dedicção do cliente não é satisfatória.

Ida de apenas uma pessoa para a reunião (por que o coordenador de negócios não pode sair de BSB?).

## APÊNDICE II – CÓDIGOS

A estrutura abaixo foi gerada com a ferramenta MaxQDA2 e representa os códigos utilizados durante a análise de dados.

### Code System

O que funciona

Soluções metodológicas

“Clientes estão conscientes”

Impacto de uma alteração de requisito

Elicitação no pólo de desenvolvimento

Economizar tempo

Cliente tem requisitos “fechados”

Manter o foco na reunião remota

Otimizar tarefas

Primeiro conversar, depois prototipar

Fechamento junto aos clientes

Requisitos melhor compreendidos

Dedicação dos clientes

Reunião dinâmica

Gerência de conhecimento

Disseminação do conhecimento

Existência de *brokers*

Registro de lições aprendidas

Registro de *rationale*

Homologação

Em BSB

Coordenação controla melhor a situação

No pólo de desenvolvimento

Disseminação de conhecimento

Dedicação dos clientes

Triagem de erros

Melhorar a comunicação e a coordenação

Registrar o máximo possível

AD registra novas entidades / relacionamentos

Analista registra pendências para BEL

Cliente registra tópicos resolvidos

- Nas áudios
- Nas reuniões remotas
- Melhorar a qualidade da documentação
  - Adotar boas práticas de documentação
  - Cultura de documentação - Analistas
  - Cultura de documentação - Clientes
  - Diminuir escopo da demanda
  - Leitura conjunta do documento
  - Revisar documentos
- Melhorar a qualidade dos testes
  - Melhorar os testes
  - Testar qualidade dos documentos
- Melhorar o nível de confiança
- Uso de metodologias
  - Mantis
  - Diagrama de atividades
  - Paper prototyping*
  - Ponto de controle remoto
  - Pontos de controle
  - Processo
  - Protótipos com cliente
- Soluções administrativas
  - Acessar própria máquina remotamente
  - Analista de requisitos proativo
  - Esforço dos clientes
    - Disponibilidade dos clientes
    - Assumir requisitos
    - Aprender aspectos técnicos
    - Fazer reuniões prévias
    - Organizar requisitos pra empresa Alpha
  - Participação ativa do coordenador
  - Realocação de pessoas
  - Relação de confiança entre empresa Alpha e seus clientes
    - Comprometimento do desenvolvimento com o cliente
    - Contato entre desenvolvimento e usuário final (satisfação)
    - Esforço da equipe de desenvolvimento
  - Tirar líder de seu ambiente de trabalho
  - Várias reuniões de elicitação
- Soluções tecnológicas
  - Uso de áudio
    - Envolvimento de todos
    - Registro de decisões
    - Controle da homologação
    - Solução de dúvidas
  - Uso de emails
    - Enviar artefatos
    - Registro de decisões
  - Uso de sistemas internos para armazenamento de arquivos
    - Controle de demandas

- Armazenamento de arquivos
- Uso de videoconferência
- O que não funciona
  - ADs concentrados no RJ
    - Atividades de baixa qualidade
    - Falta de acesso aos documentos de BD - engenharia reversa
    - Demora
    - Sobrecarga de atividades
    - Pouco conhecimento sobre o negocio
  - Análise de impacto
  - Artefatos disponíveis estão desatualizados
  - Atendimento de demandas
    - Troca da ordem das demandas
    - Escopo da demanda
  - Cientes da área de informática
  - Cientes em contato com os analistas
  - Cientes insatisfeitos
    - Regras de negocio desatualizadas
    - Sistema com bugs
  - Cientes não estão conscientes
    - Importância do documento de UCs (*use cases*)
    - Importância do processo
  - Comunicação
    - Ausência da videoconferência
    - Coordenação administra muitos emails
    - Áudio
      - Cansativa
      - Desvio do foco
      - Falta de recursos visuais
    - Envio de anexos muito grandes
    - Falta de contato com cliente durante implementação
    - Demora
  - Contato exagerado com usuário final
  - Coordenador da empresa Alpha “pensa que sabe”
  - Deslocamento de clientes custoso (tempo e dinheiro)
  - Elicitação à distância
    - Analista nem sempre consegue tirar dúvidas do cliente
    - Cientes não estão em consenso
    - Estimativas
    - Falta de dedicação dos clientes
    - Falta roteiro a ser seguido / reunião desorganizada
    - Quando o AD deve estar presente?
    - Requisitos mal entendidos / mal especificados
    - Reunião curta
    - Solicitação de mudanças
  - Elicitação no pólo de desenvolvimento
    - Falta de atendimento do cliente
    - Falta de comprometimento da equipe
  - Especificação à distância

- Demora
- Documento pouco claro
- Documentos inconsistentes
- Falta de dedicação dos clientes
- Requisitos / regras de negócio confusos
- Sobrecarga no cliente
- Falta de awareness / comunicação informal
  - Cliente -> Desenvolvimento
  - Desenvolvimento -> Cliente
  - Falta de contato após elicitação
    - Coordenador não acompanha o processo
    - Cliente não acompanha o processo
  - Prazo das atividades
  - Template* do artefato de caso de uso
- Falta de contato com usuário final
- Fechamento à distância
  - Insatisfação do analista de negócios
  - Requisitos pouco claros / errados
  - Conhecimento tácito
  - Falta de dedicação dos clientes
  - Comunicação ruim (áudio-conferência)
- Fechamento presencial
  - Pouco tempo
  - Falta de dedicação dos clientes
  - Mudança de requisitos
- Gerencia de conhecimento
  - Conhecimento centralizado / interrupções
  - Entrada de novatos
    - Atividades de baixa qualidade
    - Ausência de contato com clientes
    - Falta de padrão na implementação
    - Mentoring*
    - Pouco conhecimento do negócio
  - Não existe mecanismo para gerenciar conhecimento
  - Não existe registro de *rationale*
  - Não tem acesso a determinados documentos
  - Registro de lições aprendidas
- Homologação
  - Anexo de homologação
  - Ausência de pontos de controle
  - Baixa qualidade dos testes
  - Cliente insatisfeito
    - Documentação “ruim”
    - Funcionalidades incorretas
  - Clientes separados geograficamente
  - Conhecimento tácito
  - Homologação com ressalva
  - Homologação em BEL
    - Contato direto entre clientes e desenvolvedores

- Homologação em BSB
  - Falta de dedicação dos clientes
- Muitos erros vazados
- Não-homologação
- Pontos de controle à distância
  - Falta de controle sobre o ambiente do cliente
  - Falta de dedicação dos clientes
- Pouco tempo
- Solicitação de mudanças indevidas
- Ida de apenas um analista
  - Entendimento comum dos requisitos
  - Demais analistas não entendem corretamente os requisitos
  - Dificuldades no repasse
  - Falta de foco
  - Atrasos na reunião
- Ida de vários analistas
- Implicações no cronograma
- Matriz de rastreabilidade
- Modelo de documentação de requisitos
  - Analista insatisfeito
  - Requisitos x regras de negocio
- Pessoas indisponíveis
- Problema de ambiente no contexto remoto
- Problemas orçamentários
- Processo inadequado
  - Estimativas incorretas
  - Não prevê elaboração de diagramas
  - Anexo de homologação
  - Anexo de especificação
- Relação de confiança entre empresa Alpha e seu cliente
  - Omissão do cronograma
  - Impasse entre o que fica acordado entre analistas e clientes
  - Indicadores da empresa Alpha
  - Questões políticas
- Sistemas de controle interno são lentos
- Uso de protótipo
- Uso do *Rational Rose*
- Uso do *RequisitePro*
- Processo
  - Papeis
    - Administrador de dados
    - Analistas de sistema
    - Auxiliar
    - Cliente
    - Coordenador do cliente da empresa Alpha
    - Coordenador da empresa Alpha
    - DBA
    - Setor de TI do cliente da empresa Alpha
    - SYS

## Técnico

## Tipo

## Atividades

- Abertura da demanda
- Elicitação de requisitos
  - Criação de protótipo
  - Escolha do analista
  - Definição de escopo
- Especificação de requisitos
- Fechamento de requisitos
- Homologação
  - Reporte de erros
  - Teste do sistema
- Manter modelo de dados
- Pontos de controle
- Repasse de requisitos
- Revisão de requisitos
- Solução de dúvidas
  - Via email
  - Reunião presencial
  - Via áudio

## Artefatos

- Anexos (todos)
- Atas
- Casos de uso
- Documentos de BD
- Documento de visão
- Relatório de análise de impacto
- Roteiro de teste

## Sugestões

- Adaptar o processo
- Aplicar práticas de gerencia de portfólio
- Aprimorar os pontos de controle
- Aumentar a qualidade dos testes
- Facilitar a negociação de requisitos
  - Explicar aspectos técnicos pro cliente
- Gerência de conhecimento
  - Externalização do conhecimento
    - Elaborar documentações
    - Realizar palestras
    - Utilizar base de soluções
  - Facilitar a entrada de novatos
  - Mapeamento de habilidades
  - Realizar repasses eficazes
    - Disponibilizar documentação
    - Tornar o *liaison* disponível
  - Tratar a rotatividade da empresa Alpha
- Melhorar a documentação
  - Criar diagramas de navegação

- Criar links entre documentos
- Detalhar bastante os casos de uso
- Detalhar o relatório de análise de impacto
- Ouvir a opinião do cliente
- Protótipo deveria fazer parte da documentação
- Reorganizar a documentação
- Reorganizar os fluxos alternativos
- Separar documentação
- Melhorar comunicação/confiança
  - Enviar mais analistas
  - Marcar pontos de controle em BEL
  - Motivar os analistas a se alinharem ao cliente
  - Realizar um workshop anual de requisitos
  - Utilizar recursos audiovisuais
- Permitir que o cliente acompanhe o processo
- Realizar escrita colaborativa