



PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL  
FACULDADE DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO  
CURSO DE MESTRADO



**Leandro Teixeira Lopes**

***Um Modelo de Processo  
de Engenharia de Requisitos para  
Ambientes de Desenvolvimento  
Distribuído de Software***

Porto Alegre, 2004

LEANDRO TEIXEIRA LOPES

**Um Modelo de Processo de Engenharia de Requisitos para Ambientes de  
Desenvolvimento Distribuído de Software**

Dissertação apresentada como requisito parcial  
à obtenção do grau de Mestre, pelo Programa  
de Pós-Graduação em Ciência da Computação  
da Pontifícia Universidade Católica do Rio  
Grande do Sul.

Orientador: Prof. Dr. Jorge Luis Nicolas Audy

Porto Alegre, 2004



## Dados Internacionais de Catalogação na Publicação (CIP)

L864m Lopes, Leandro Teixeira  
Um modelo de processo de engenharia de requisitos para  
ambientes de desenvolvimento distribuído de software/  
Leandro Teixeira Lopes. – Porto Alegre, 2004.  
142 f.  
  
Dissertação (Mestrado) – Fac. de Informática, PUCRS, 2004.  
  
1. Engenharia de Software. 2. Engenharia de Requisitos.  
I. Título.  
  
CDD 005.1

**Ficha Catalográfica elaborada pelo  
Setor de Processamento Técnico da BC-PUCRS**



## TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada “Um modelo de Processo de Engenharia de Requisitos para Ambientes de Desenvolvimento Distribuído de Software” apresentada por **Leandro Teixeira Lopes**, como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Área de Sistemas de Informação, aprovada em 31/01/2005 pela Comissão Examinadora:

Prof. Dr. Jorge Luis Nicolas Audy  
Orientador

PPGCC/PUCRS

Prof. Dr. Marcelo Blois Ribeiro

PPGCC/PUCRS

Prof. Dr. Ricardo Melo Bastos -

PPGCC/PUCRS

Prof. Dr. Sérgio Crespo Coelho da Silva Pinto

UNISINOS

Homologada em 28/06/06, conforme Ata N° 17 pela Comissão Coordenadora.

Prof. Dr. **Avelino Francisco Zorzo**  
Coordenador.

*“À minha família - meu porto seguro, meu “Sul”...”*

## **AGRADECIMENTOS**

Ao Prof. Dr. Jorge Audy, pela amizade, parceria e orientação. Aprendi contigo muito mais que questões técnicas, recebi lições de dedicação, caráter e paixão pelo ensino e pesquisa que jamais esquecerei.

Aos professores Dr. Ricardo Bastos, Dr. Marcelo Blois e Dra. Karin Becker por todo acompanhamento dado, críticas e sugestões que muito auxiliaram na evolução desta pesquisa.

Ao Prof. Dr. Julio C. S. do Prado Leite, pela oportunidade de conviver e aprender com seu grupo de pesquisas, reconhecido internacionalmente na área de engenharia de requisitos.

Ao Prof. Dr. Toacy Cavalcante, pelo acompanhamento dedicado durante o estágio de docência.

Aos amigos e colegas Sabrina Marczak e Rafael Prikladnicki, pela contribuição dada a este trabalho, disponibilidade para discussões e revisões. Ao Azriel Majdenbaum, pela amizade e colaboração com este trabalho.

Aos grandes companheiros Ênio e Ilvio, pelo grande apoio e amizade que sempre dedicaram.

À minha família, que sempre me fez acreditar no amor e nos sonhos. Aos meus pais, meu exemplo de vida, pelo constante apoio, carinho e amizade sempre dedicados. Às minhas irmãs e cunhados, por toda a força e alegria. Aos sobrinhos, por terem compreendido os momentos em que o “tio lê” não podia brincar porque estava trabalhando.

À Suelen, pelo amor, alegria e compreensão, és minha namorada agora e sempre.

Ao Centro de Desenvolvimento e Pesquisa Dell/PUCRS (CDPe), por ter financiado esta pesquisa e meu curso de mestrado.

“O mais importante é não parar de questionar. A curiosidade tem sua própria razão de ser. Não podemos fazer nada senão contemplar extasiados os mistérios da eternidade, da vida, da maravilhosa estrutura da realidade. É mais que suficiente tentarmos simplesmente compreender um pouco desse mistério a cada dia. Nunca perca a sagrada curiosidade.”

(Albert Einstein)

## RESUMO

A crescente globalização do ambiente de negócios tem afetado diretamente o mercado de desenvolvimento de software. Em busca de vantagens competitivas como baixos custos, produtividade e qualidade na área de desenvolvimento de sistemas, diversas organizações optaram por distribuir o processo de desenvolvimento de software dentro de seu país, ou em outros países, como Índia, China e Brasil. Entretanto, os desafios apresentados pela distribuição da equipe envolvida no processo de software são significativos. Nesse contexto, a engenharia de requisitos é uma atividade influenciada pela distribuição das equipes. O processo de requisitos, mesmo em ambientes co-localizados, é crítico no desenvolvimento de software. Ao lidar com a distância entre os envolvidos, as dificuldades com requisitos tendem a se exacerbar. Para tratar questões como dispersão geográfica, diferenças culturais e dificuldades de comunicação, torna-se necessária a definição de novos processos, padrões e ferramentas, de forma a reduzir o impacto da dispersão das equipes na engenharia de requisitos. Nesse sentido, esta dissertação de mestrado tem como objetivo propor um modelo de processo de engenharia de requisitos visando tratar questões relacionadas ao desenvolvimento distribuído de software. O principal método de pesquisa utilizado foi o estudo de caso e a base empírica da pesquisa envolve uma unidade de desenvolvimento de software de uma empresa multinacional de grande porte localizada no Brasil. A pesquisa contribui no sentido de propor um modelo de processo de engenharia de requisitos adequado à realidade de equipes dispersas.

**Palavras-chave:** Engenharia de Software, Processo de Desenvolvimento de Software, Desenvolvimento Distribuído de Software, Desenvolvimento Global de Software, Engenharia de Requisitos.



## **ABSTRACT**

Crescent globalization in business environments has affected the software development market. Aiming competitive advantages as low costs, high productivity and quality in systems development, several organizations decided to distribute their development process inside or outside their countries. However, team dispersion introduces several challenges to process development. In this context, requirements engineering is one activity highly influenced by team dispersion. Requirements process, even in co-located environments, is critical. When dealing with distance among stakeholders, requirements difficulties tends to be exacerbated. It is necessary new processes, patterns and tools to reduce the impact of team dispersion in requirements engineering and address geographical dispersion, cultural differences and communication difficulties, for instance. In this sense, the objective of this master thesis is to propose a requirements engineering process model that addresses questions related to distributed software development. The main research method used was case study, conducted in a software development unit of a multinational organization located in Brazil. This research contributes by proposing a requirements engineering process adequate to distributed teams.

**Keywords:** Software Engineering, Software Process Development, Distributed Software Development, Global Software Development, Requirements Engineering.

## LISTA DE FIGURAS

Figura 1 - Modelo de processo de engenharia de requisitos (adaptado de [KOT98]).	23
Figura 2 - Principais razões envolvidas no DDS.	30
Figura 3 - Demanda por software em relação ao número de microcomputadores e profissionais disponíveis (adaptado de [KAR98]).	31
Figura 4 - Custos de coordenação e interdependência [CAR99] [THO 76].	36
Figura 5 - Níveis de interação contextual [MAR 2001].	38
Figura 6 - Número de canais de comunicação em uma equipe.	40
Figura 7 - Modelo de impacto dos desafios e atividades afetadas da engenharia de requisitos devido a problemas com o DGS (adaptado de [DAM02]).	41
Figura 8 - Desenho de pesquisa.	47
Figura 9 - Exemplo de cenário de dispersão considerado no processo preliminar.	57
Figura 10 - Distribuição dos engenheiros de requisitos no exemplo de cenário.	58
Figura 11 - Interação entre engenheiros de requisitos, usuários, clientes e desenvolvedores.	58
Figura 12 - Processo preliminar de engenharia de requisitos para ambientes de DDS.	59
Figura 13 - EC1 - Função dos respondentes no projeto 1.	64
Figura 14 - EC1 - Função dos respondentes no projeto 2.	69
Figura 15 - EC2 - Formação dos respondentes.	80
Figura 16 - EC2 - Equipe de engenheiros de requisitos distribuída em relação a usuários e clientes.	80
Figura 17 - EC2 - Usuários e clientes distribuídos entre si e em relação aos engenheiros de requisitos.	84
Figura 18 - EC2 - Engenheiros de requisitos distribuídos globalmente em relação à equipe de desenvolvimento.	85
Figura 19 - EC2 - Equipe de desenvolvimento distribuída entre si e em relação ao engenheiro de requisitos.	88
Figura 20 - Modelo de processo de ER em DDS proposto.	94
Figura 21 - Fase 1 - Definições iniciais.	96

Figura 22 - Fase 2 - Mapeamento do contexto .....	101
Figura 23 - Fase 3 - Criação da especificação de requisitos .....	104
Figura 24 - Aspectos enfocados no trabalho .....	122
Figura 25 - Aspectos enfocados no trabalho .....	128
Figura 26 - Engenheiro de requisitos distribuído em relação a usuários e clientes.....	130
Figura 27 - Usuários e clientes distribuídos entre si e em relação aos Engenheiros de Requisitos .....	131
Figura 28 - Engenheiros de Requisitos distribuído globalmente em relação à equipe de desenvolvimento .....	132
Figura 29 - Equipe de desenvolvimento distribuída entre si e em relação aos engenheiros de requisitos.....	133

## LISTA DE TABELAS

Tabela 1 - Relação entre dificuldades da engenharia de requisitos e autores referenciados....	24
Tabela 2 - Principais desafios ao desenvolvimento distribuído de software .....	33
Tabela 3 - Lições aprendidas (adaptado de [PRI03]) .....	43
Tabela 4 - Fatores culturais visíveis e ocultos (adaptado de [MAH98]) .....	44
Tabela 5 - EC1 - Resultados da dimensão 3 no projeto 1 .....	64
Tabela 6 - EC1 - Resultados da dimensão 4 no projeto 1 .....	66
Tabela 7 - EC1 - Resultados da dimensão 5 no projeto 1 .....	68
Tabela 8 - EC1 - Resultados da dimensão 3 no projeto 2 .....	70
Tabela 9 - EC1 - Resultados da dimensão 4 no projeto 2 .....	71
Tabela 10 - EC1 - Resultados da dimensão 5 no projeto 2 .....	73
Tabela 11 - EC1 - Pontos fortes do processo preliminar .....	74
Tabela 12 - EC1 - Pontos fracos do processo preliminar.....	75
Tabela 13 - EC2 - Síntese das respostas para Informação Ambígua no cenário 1 .....	81
Tabela 14 - EC2 - Síntese das respostas para Informação Faltante no cenário 1 .....	82
Tabela 15 - EC2 - Síntese das respostas para Informações Inconsistentes no cenário 1 .....	83
Tabela 16 - EC2 - Síntese das respostas para Informação Incorreta no cenário 1 .....	83
Tabela 17 - EC2 - Síntese das respostas para Informação Ambígua no cenário 3 .....	85
Tabela 18 - EC2 - Síntese das respostas para Informação Faltante no cenário 3 .....	86
Tabela 19 - EC2 - Síntese das respostas para Informações Inconsistentes no cenário 3 .....	87
Tabela 20 - EC2 - Síntese das respostas para Informação Incorreta no cenário 3 .....	88
Tabela 21 - EC2 - Principais fatores que ampliam problemas na especificação.....	90
Tabela 22 - Atividades relacionadas ao estabelecimento da infra-estrutura .....	96
Tabela 23 - Artefatos de entrada e saída da atividade de definição do idioma da especificação .....	97
Tabela 24 - Artefatos de entrada e saída da atividade de definição de dicionário .....	98
Tabela 25 - Artefatos de entrada e saída da atividade de definição de responsabilidades.....	98

Tabela 26 - Artefatos de entrada e saída da atividade de definir pontos focais e/ou embaixadores .....	99
Tabela 27 - Artefatos de entrada e saída da atividade de definição da forma de interação. ..	100
Tabela 28 - Artefatos de entrada e saída da atividade de definição dos padrões de especificação.....	100
Tabela 29 - Atividades relacionadas ao mapeamento do contexto.....	101
Tabela 30 - Artefatos de entrada e saída da atividade de criar/atualizar base de informações culturais e de contexto.....	102
Tabela 31 - Artefatos de entrada e saída da atividade de criar/atualizar base de conceitos do universo de informações (UdI). .....	103
Tabela 32 - Artefatos de entrada e saída da atividade de distribuição de informações de negócio. ....	103
Tabela 33 - Atividades relacionadas à criação da especificação de requisitos .....	104
Tabela 34 - Artefatos de entrada e saída da atividade de criar e distribuir artefatos iniciais de requisitos.....	105
Tabela 35 - Artefatos de entrada e saída da atividade de evoluir artefatos de requisitos. ....	106
Tabela 36 - Artefatos de entrada e saída da atividade de inspecionar artefatos de requisitos. ....	106
Tabela 37 - Artefatos de entrada e saída da atividade de validar artefatos de requisitos. ....	107
Tabela 38 - Atividades relacionadas ao gerenciamento de requisitos .....	108
Tabela 39 - Artefatos de entrada e saída da atividade de gerenciar artefatos de requisitos... ..	108
Tabela 40 - Artefatos de entrada e saída da atividade de gerenciar base de informações culturais e de contexto.....	109
Tabela 41 - Artefatos de entrada e saída da atividade de gerenciar base de conceitos do UdI. ....	110

## LISTA DE SIGLAS

DDS	Desenvolvimento Distribuído de Software
DGS	Desenvolvimento Global de Software
EC1	Estudo de Caso 1
EC2	Estudo de Caso 2
ER	Engenharia de Requisitos
GSD	<i>Global Software Development</i>
IEEE	<i>Institute of Electrical and Electronic Engineers</i>
MSF	<i>Microsoft Solutions Framework</i>
PMI	<i>Project Management Institute</i>
PPGCC	Programa de Pós-Graduação em Ciência da Computação
PUCRS	Pontifícia Universidade Católica do Rio Grande do Sul
RUP	<i>Rational Unified Process</i>
SW-CMM	<i>Software Capability Maturity Model</i>
SRS	<i>Software Requirements Specification</i>
UdI	Universo de Informações

# SUMÁRIO

1.	INTRODUÇÃO .....	16
1.1.	JUSTIFICATIVA.....	17
1.2.	QUESTÃO DE PESQUISA .....	18
1.3.	OBJETIVOS .....	18
1.4.	ORGANIZAÇÃO DO VOLUME.....	19
2.	BASE TEÓRICA .....	20
2.1.	ENGENHARIA DE REQUISITOS .....	20
2.1.1.	Conceitos.....	20
2.1.2.	Processo de Engenharia de Requisitos .....	23
2.1.3.	Dificuldades .....	24
2.2.	DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE.....	27
2.2.1.	Conceitos.....	28
2.2.2.	Razões que levam ao DDS.....	29
2.2.3.	Desafios ao desenvolvimento distribuído de software .....	32
2.3.	TRABALHOS RELACIONADOS .....	40
2.3.1.	Abordagem de Damian e Zowghi (2002) .....	40
2.3.2.	Abordagem de Lloyd, Rosson e Arthur (2002).....	41
2.3.3.	Abordagem de Prikladnicki, Audy e Evaristo (2003) .....	42
2.3.4.	Abordagem de Mahemoff e Johnston (1998) .....	43
2.3.5.	Abordagem de Zowghi (2002) .....	44
3.	MÉTODO DE PESQUISA.....	46
3.1.	DESENHO DE PESQUISA .....	46
3.2.	ESTUDOS DE CASO .....	48
3.2.1.	Estudo de Caso 1 .....	48
3.2.2.	Estudo de Caso 2 .....	49
4.	CATEGORIAS E FATORES ASSOCIADOS À ENGENHARIA DE REQUISITOS NO DDS.....	51
4.1.	COMUNICAÇÃO .....	52
4.2.	CULTURA.....	53
4.3.	GESTÃO DE CONHECIMENTO .....	54
4.4.	ASPECTOS TÉCNICOS .....	55
5.	PROCESSO PRELIMINAR.....	57
5.1.	CONTEXTO .....	57
5.2.	PROCESSO PRELIMINAR .....	59

5.3. MODELO DE ESPECIFICAÇÃO DE REQUISITOS EM LINGUAGEM NATURAL .....	61
6. RESULTADOS DOS ESTUDOS DE CASO .....	62
6.1. ESTUDO DE CASO 1 .....	62
6.1.1. Descrição do Instrumento de Coleta de Dados .....	62
6.1.2. Análise de Dados.....	63
6.1.3. Consolidação dos resultados .....	73
6.1.4. Lições aprendidas .....	76
6.2. ESTUDO DE CASO 2 .....	78
6.2.1. Instrumento de Coleta de Dados .....	78
6.2.2. Análise de resultados .....	79
6.2.3. Consolidação dos resultados .....	89
6.2.4. LIÇÕES APRENDIDAS .....	90
7. MODELO DE PROCESSO PROPOSTO .....	93
7.1. PAPÉIS E RESPONSABILIDADES .....	94
7.2. FASE 1 - DEFINIÇÕES INICIAIS.....	95
7.2.1. Definir idioma da especificação .....	96
7.2.2. Definir dicionário .....	97
7.2.3. Definir responsabilidades.....	98
7.2.4. Definir pontos focais e/ou embaixadores.....	98
7.2.5. Definir a forma de interação .....	99
7.2.6. Definir padrões de especificação.....	100
7.3. FASE 2 - MAPEAMENTO DO CONTEXTO .....	100
7.3.1. Criar/atualizar base de informações culturais e de contexto.....	101
7.3.2. Criar/atualizar base de conceitos do universo de informações (UdI).....	102
7.3.3. Distribuição de informações de negócio.....	103
7.4. FASE 3 - CRIAÇÃO DA ESPECIFICAÇÃO DE REQUISITOS .....	103
7.4.1. Criar e distribuir artefatos iniciais de requisitos.....	105
7.4.2. Evoluir artefatos de requisitos.....	105
7.4.3. Inspeccionar artefatos de requisitos .....	106
7.4.4. Validar artefatos de requisitos.....	106
7.5. FASE 4 - GERENCIAMENTO DE REQUISITOS.....	107
7.5.1. Gerenciar artefatos de requisitos .....	108
7.6. ATIVIDADES DE SUPORTE .....	108
7.6.1. Gerenciar base de informações culturais e de contexto.....	109
7.6.2. Gerenciar base de conceitos do UdI .....	109
8. CONSIDERAÇÕES FINAIS .....	111
8.1. CONTRIBUIÇÕES .....	111
8.2. LIMITAÇÕES DO ESTUDO .....	112
8.3. ESTUDOS FUTUROS .....	112
REFERÊNCIAS BIBLIOGRÁFICAS.....	113
APÊNDICE 1 - PROTOCOLO PARA ESTUDO DE CASO 1.....	120
APÊNDICE 2 - PROTOCOLO PARA ESTUDO DE CASO 2.....	127
APÊNDICE 3 - DADOS CONSOLIDADOS DO ESTUDO DE CASO 2 .....	134
APÊNDICE 4 - ARTIGOS PUBLICADOS .....	138



# 1. INTRODUÇÃO

A crescente globalização do ambiente de negócios tem afetado diretamente o mercado de desenvolvimento de software [HER01]. Em busca de vantagens competitivas em termos de custos, produtividade ou qualidade na área de desenvolvimento de sistemas [MCC96], por exemplo, diversas organizações optaram por distribuir o processo de desenvolvimento de software dentro de seu país, ou em outros países, como a Índia e o Brasil. Estas regiões oferecem, muitas vezes, incentivos fiscais ou possuem grande concentração de massa crítica em determinadas áreas [PRI02b].

Embora o desenvolvimento de software tenha evoluído consideravelmente nas últimas décadas, ainda são enfrentadas diversas dificuldades. Muitos projetos de software têm sido entregues atrasados ou com custos superiores ao esperado. Frequentemente, o software não realiza as funcionalidades desejadas por seus usuários. Ao optar por distribuir o processo de desenvolvimento, além de todas as dificuldades inerentes ao desenvolvimento co-localizado, uma organização começa a enfrentar diversos desafios de adaptação, diferenças culturais, planejamento do trabalho, comunicação, entre outros.

No processo de desenvolvimento de software, a engenharia de requisitos destaca-se como um ponto fundamental para o sucesso dos projetos. Estudos mostram que os maiores problemas no desenvolvimento de software são relacionados com requisitos [STA95]. Uma incorreta engenharia de requisitos pode exercer um impacto negativo em um projeto de software, como a necessidade de um novo ciclo de especificação, projeto, codificação e teste, afetando diretamente os custos e prazos envolvidos.

Em ambientes de desenvolvimento distribuído de software (DDS), dificuldades como distância, comunicação e cultura causam um aprofundamento dos problemas inerentes ao processo de engenharia de requisitos, que adquire um caráter ainda mais crítico [ZOW02]. Como a distância envolvida pode compreender países distantes, comumente a linguagem e a cultura são diferentes. Com isso, os problemas causados por ambigüidade e falta de clareza nos requisitos são intensificados. A compreensão dos requisitos ao serem lidos em uma língua diferente da nativa é mais limitada, levando a interpretações incorretas. Diferenças culturais como atitude em relação à hierarquia, riscos e valores culturais podem ampliar a possibilidade de conflitos. Sem o conhecimento presencial do ambiente onde o software será inserido, a compreensão das razões e expectativas motivadoras do software por parte da equipe de desenvolvimento é reduzida [DAM02].

A comunicação também apresenta novos desafios. Com a perda de contato face-a-face entre a equipe de desenvolvimento e os usuários, existe uma maior dificuldade de esclarecimento em caso de dúvidas. Além disso, com a utilização de meios de comunicação de baixo contexto, como correio eletrônico, o contato informal entre os membros dos diversos grupos é limitado, reduzindo a confiança entre eles. Em casos de grupos separados por diversos fusos horários, em geral, ocorre uma maior demora na tomada de decisões. Uma simples troca de mensagens por correio eletrônico para esclarecimento de um requisito pode levar dias se os horários de trabalho não coincidirem. Reuniões por vídeo ou teleconferência podem perder a efetividade. O comprometimento entre as equipes pode ser reduzido, se não houver confiança entre elas.

Considerando-se a crescente adoção do desenvolvimento distribuído de software, ainda existem poucos estudos abordando seu impacto na engenharia de requisitos [DAM02]. Nos estudos encontrados, os aspectos técnicos não são avaliados em detalhe. Dessa forma, mostra-se necessário novos estudos sobre processos, padrões e ferramentas para compensar as dificuldades causadas pela distribuição das equipes na engenharia de requisitos.

Com base nesta contextualização, o tema escolhido para esta dissertação de mestrado é o processo de engenharia de requisitos em ambientes de desenvolvimento distribuído de software. Este assunto é considerado de grande relevância por diversos autores [DAM02][LLO02][ZOW02], se constituindo em uma boa área para pesquisa e aplicação.

## **1.1. JUSTIFICATIVA**

É cada vez mais significativo o número de empresas que estão distribuindo seus processos de desenvolvimento de software ao redor do mundo. Por isso, o desenvolvimento distribuído tem atraído um maior volume de pesquisas na área de engenharia de software [PRI03b]. Os engenheiros de software têm reconhecido a grande influência desta nova forma de trabalho e estão em busca de modelos que facilitem o desenvolvimento de software com equipes geograficamente dispersas.

Segundo Carmel [CAR99], as principais características que diferenciam o desenvolvimento distribuído de software do desenvolvimento co-localizado são a dispersão geográfica, a dispersão temporal e as diferenças culturais.

Problemas fundamentais da engenharia de requisitos são exacerbados quando as equipes de desenvolvimento de software estão distribuídas geograficamente [ZOW02]. A engenharia de requisitos, por ser uma atividade largamente dependente da comunicação, sofre com o impacto da dispersão das equipes. Com as equipes distribuídas, a comunicação torna-se

menos freqüente e mais restrita. A dispersão temporal, principalmente advinda da diferença de fuso-horário, afeta atividades como a elicitação e a negociação de requisitos, introduzindo demoras e dificultando o contato síncrono entre as equipes. A engenharia de requisitos, por depender de um bom relacionamento entre os envolvidos, também é influenciada pelas diferenças culturais.

Estes desafios apresentados pela dispersão das equipes na engenharia de requisitos conduzem à necessidade de novas abordagens para redução de seu impacto. Segundo Zowghi [ZOW02], o desenvolvimento distribuído de software necessita claramente de um processo de engenharia de requisitos diferenciado, tema central deste estudo.

## **1.2. QUESTÃO DE PESQUISA**

Com base na contextualização apresentada na seção anterior, o objetivo principal desta pesquisa é a definição de um modelo de processo de engenharia de requisitos para ambientes de desenvolvimento distribuído de software, respondendo à seguinte questão de pesquisa: *Como desenvolver a engenharia de requisitos em um ambiente de desenvolvimento distribuído de software visando melhorar a qualidade das especificações geradas e simplificar as tarefas de gerenciamento?*

## **1.3. OBJETIVOS**

A engenharia de requisitos é uma das fases mais importantes no desenvolvimento de software, pois nela são identificados, analisados e definidos os propósitos, funcionalidades e o escopo do software [ZOW02]. A distribuição das equipes envolvidas no processo de desenvolvimento de software apresenta diversos desafios à engenharia de requisitos. Por ser uma atividade rica em comunicação a engenharia de requisitos é afetada diretamente pela distância geográfica. Ambigüidades e problemas de entendimento são exacerbados com as diferenças de idioma e cultura. Da mesma forma, questões como gestão do conhecimento e diferenças de fuso horário exercem impacto nas atividades da engenharia de requisitos em ambientes de desenvolvimento distribuído de software.

Dessa forma, o objetivo geral desta pesquisa é propor um modelo de processo de engenharia de requisitos visando tratar questões relacionadas ao desenvolvimento distribuído de software.

Para atender a este objetivo geral emergem os seguintes objetivos específicos:

- Aprofundar o conhecimento em engenharia de requisitos, DDS e tópicos relacionados;

- Identificar as principais categorias e fatores relacionados à engenharia de requisitos em ambientes de desenvolvimento distribuído de software;
- Elaboração de uma proposta preliminar de processo de engenharia de requisitos para ambientes de desenvolvimento distribuído de software;

#### **1.4. ORGANIZAÇÃO DO VOLUME**

Este volume está organizado em 8 capítulos. No capítulo 2 apresenta-se o referencial teórico desta pesquisa, envolvendo os principais conceitos e implicações das áreas do estudo: desenvolvimento distribuído de software e engenharia de requisitos. A apresentação deste referencial teórico é feita de forma abrangente, em virtude da natureza exploratória desta pesquisa e de métodos qualitativos, que determinam a necessidade de uma larga e consistente fundamentação teórica [YIN01].

No capítulo 3, apresenta-se a metodologia de pesquisa, descrevendo cada uma das etapas do estudo, e apresentando o desenho de pesquisa em detalhe. As principais categorias e fatores associados à engenharia de requisitos no desenvolvimento distribuído de software são descritos no capítulo 4, com base na teoria existente na área.

O capítulo 5 apresenta a proposta preliminar de processo de engenharia de requisitos para ambientes de DDS, avaliado em um dos estudos de caso descritos no capítulo 6.

No capítulo 6 descrevem-se detalhadamente os dois estudos de caso desenvolvidos em organizações que realizam desenvolvimento distribuído de software.

O modelo de processo proposto é apresentado no capítulo 7, como consequência do processo de pesquisa como um todo, apoiado na revisão teórica desenvolvida (Capítulo 2) e nos resultados obtidos com o estudo de caso (Capítulo 6).

Finalmente, no capítulo 8 apresentam-se as considerações finais sobre o tema e enfocam-se os aspectos relacionados às contribuições e limitações deste estudo. Conclui-se destacando rumos para futuras pesquisas na área.

## **2. BASE TEÓRICA**

Coerente com estudos de base qualitativa, o referencial teórico representa importante etapa da pesquisa [YIN 01]. Na seção 2.1 apresenta-se a base teórica em engenharia de requisitos, incluindo conceituação e descrição do processo. Após, na seção 2.2 é abordado o desenvolvimento distribuído de software, descrevendo os principais conceitos e motivações. Ao final são descritos os trabalhos relacionados (seção 2.3).

### **2.1. ENGENHARIA DE REQUISITOS**

O sucesso no desenvolvimento de um software é medido principalmente pela forma com que ele realiza a tarefa para qual foi proposto [NUS00]. O esforço de desenvolvimento é total ou parcialmente desperdiçado se o software, por melhor que seja a qualidade de sua codificação, não cumpre com a tarefa a que foi destinado. Da mesma forma, se a base tecnológica (hardware, software e dispositivos) necessária ao software em questão não for compatível com a base existente onde ele será utilizado, todo (ou a maior parte) o trabalho de desenvolvimento pode se tornar inútil.

Para que o sucesso possa ser atingido, é fundamental que seja realizada uma tarefa de identificação e documentação das necessidades e propósitos de um software. Esta tarefa, muitas vezes, exige uma compreensão do ambiente onde o software será inserido, considerando as características do negócio, as possíveis modificações futuras e as necessidades reais envolvidas no processo.

Pressman [PRE01] destaca que não existe uma forma incontestável de assegurar que a especificação de um sistema está propriamente de acordo com as necessidades do cliente, e que satisfaz suas necessidades. Este é um desafio complexo enfrentado pelos engenheiros de software, e o melhor modo de encará-lo é através de um processo consistente de engenharia de requisitos.

Neste capítulo são apresentados os conceitos básicos e é destacada a importância da engenharia de requisitos no processo de desenvolvimento de software, de acordo com a visão de diversos autores. Em seguida são apresentadas algumas das dificuldades e riscos da engenharia de requisitos, como forma de caracterização do assunto em estudo.

#### **2.1.1. Conceitos**

A engenharia de requisitos possui uma diversidade de termos que recebem diferentes definições dependendo dos autores e do momento em que foram escritos. Existem críticas

inclusive à utilização do termo engenharia de requisitos, pois alguns autores não consideram esta uma verdadeira “engenharia” [NUS00].

Em 1995, Pressman [PRE95] utilizava o termo análise de requisitos, para referir-se ao processo como um todo. Atualmente [PRE01], adotou o termo engenharia de requisitos, considerando a análise de requisitos como uma de suas etapas.

Considerando esta diversidade de definições e termos existentes, esta seção busca reunir a visão de diversos autores com relação aos principais conceitos envolvidos na engenharia de requisitos, oferecendo um embasamento inicial do assunto.

### **Software**

Durante o desenvolvimento deste texto, o termo software será utilizado com significado equivalente a sistema, cujo significado é mais abrangente e de aplicação a variados contextos. Sendo assim, software deve ser entendido como uma ferramenta de suporte à solução de problemas com o uso da informática [ZAN99].

### **Requisito**

De acordo com Dorfmann e Thayer [THA00] um requisito é uma característica do software necessária para o usuário solucionar um problema de forma a atingir um objetivo. Uma característica de software que deve ser realizada ou implementada por um sistema ou componente de sistema para satisfazer um contrato, padrão, especificação ou outra documentação formalmente imposta.

Siddiqi [SID96] define requisitos como uma declaração completa do que o software irá fazer sem referir-se a como fazê-lo. Kruchten [KRU00] define um requisito como uma condição ou capacidade que um software deve realizar.

Estas definições caracterizam os requisitos como forma de expressar as características necessárias e restrições impostas a um software, para que ele satisfaça as expectativas existentes. Gougen [GOG96] acrescenta mais um elemento a esta definição, que tem se mostrado importante na utilização de requisitos. Em sua visão, requisitos são propriedades que um software deve possuir para funcionar com sucesso no ambiente onde será utilizado. Neste caso, o autor considera tanto o contexto social como o técnico do ambiente em que o software será utilizado. Esta contextualização é necessária, pois muitas das informações para os requisitos estão ligadas ao ambiente social dos usuários e gerentes, e podem ser informais ou tácitas.

Diversos autores [THA00][NUS00][FRA98][SID96][KRU00] dividem os requisitos em dois tipos: funcionais (o que o software deve fazer) e não-funcionais (como o software se comporta em relação a alguns atributos observáveis, como desempenho, confiabilidade, etc.).

De acordo com Kruchten [KRU00], requisitos funcionais são utilizados para expressar o comportamento de um software através da especificação das condições de entrada e saída que deve possuir. Requisitos não-funcionais são atributos desejados de qualidade que não são descritos pelos requisitos funcionais.

Thayer e Dorfman [THA00] complementam: “Os requisitos funcionais capturam a natureza da interação entre o componente e seu ambiente. Os requisitos não-funcionais restringem os tipos de solução que podem ser consideradas”.

### ***Stakeholder***

A definição mais utilizada para *stakeholder* [SHA99][POU99] provavelmente é a de Freeman, na literatura de gerência estratégica ([FRE84] apud [SHA99][POU99]): “Um *stakeholder* em uma organização é (por definição) qualquer grupo ou indivíduo que pode afetar ou ser afetado pela obtenção dos objetivos da organização”.

Em uma definição aplicada a sistemas de informação, Leffingwell e Widrig [LEF00] caracterizam *stakeholder* como qualquer pessoa que possa ser afetada pela implementação de um sistema ou aplicação.

Kruchten [KRU00] define *stakeholder* como qualquer pessoa ou representante de uma organização que possua um *stake* - um grande interesse - no resultado de um projeto. O *stakeholder* mais óbvio é o usuário final, mas devemos lembrar de considerar outros *stakeholders* importantes: compradores, contratantes, desenvolvedores, gerentes de projeto ou quaisquer outros que se importem ou cujas necessidades devam ser satisfeitas pelo projeto.

### **Engenharia de requisitos**

Segundo Zave [ZAV97], a engenharia de requisitos é a área da engenharia de software que se preocupa com as metas reais, funções e restrições de software. Ela também se preocupa com o relacionamento destes fatores com uma precisa especificação do comportamento do software e sua evolução através do tempo e das famílias de software.

Na visão de Pressmann [PRE01], a engenharia de requisitos fornece os mecanismos apropriados para entender o que o cliente quer, analisar necessidades, analisar o que é possível ser feito, negociar uma solução razoável, especificar uma solução não ambígua, validar a especificação e gerenciar os requisitos conforme eles se transformam em um software operante.

Thayer e Dorfman [THA00] definem engenharia de requisitos como a ciência e disciplina preocupada com a análise e documentação dos requisitos, incluindo análise das necessidades e análise e especificação dos requisitos. Além disso, a engenharia de requisitos

fornece mecanismos apropriados para facilitar as atividades de análise, documentação e verificação.

### 2.1.2. Processo de Engenharia de Requisitos

Um processo de engenharia de requisitos é um conjunto estruturado de atividades que são seguidas para derivar, validar e manter um documento de requisitos de um sistema. Uma descrição completa de um processo deve incluir as atividades que devem ser conduzidas, sua estrutura ou agenda destas atividades, as entradas e saídas de cada atividade e as ferramentas utilizadas para suportar a engenharia de requisitos [KOT98].

Não existe um processo único que possa ser utilizado por todas as organizações. Cada organização deve desenvolver seu próprio processo para adaptar-se aos tipos de sistema em desenvolvimento, cultura organizacional, experiência dos envolvidos, entre outros. De forma abstrata, a maioria dos processos de engenharia de requisitos podem ser descritos em um modelo de atividades de alta granularidade, como apresentado na Figura 1.

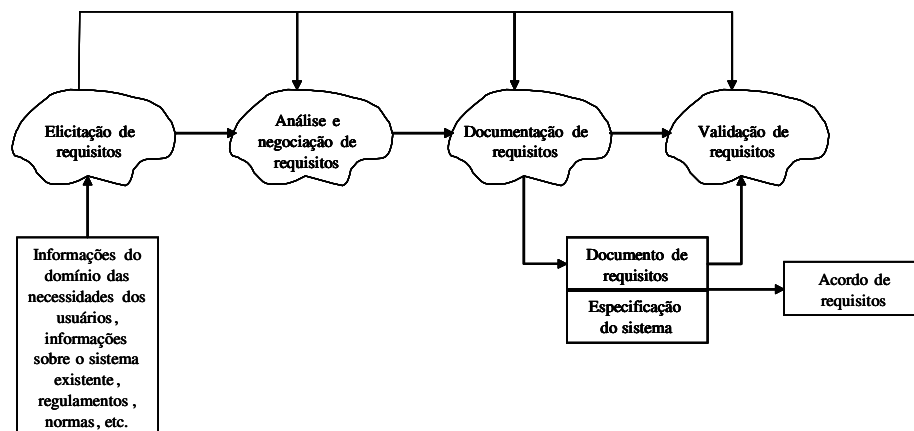


Figura 1 - Modelo de processo de engenharia de requisitos (adaptado de [KOT98]).

As atividades apresentadas no modelo são descritas a seguir [KOT98] [SOM97]:

**Elicitação dos requisitos:** nesta fase são identificadas as expectativas e necessidades dos *stakeholders* com relação ao software a ser desenvolvido;

**Análise e negociação dos requisitos:** depois de obtidos os requisitos iniciais, estes são utilizados como base para análise dos requisitos. A análise distribui os requisitos em categorias, explora as relações entre eles, e classifica a importância de cada um dos requisitos de acordo com as necessidades dos *stakeholders*. Os requisitos são negociados para decidir quais devem ser aceitos, de forma a obter consenso;



**Documentação dos requisitos:** os requisitos são documentados em um nível apropriado de detalhe. Em geral, é produzido um documento de especificação de requisitos, de forma que todos os *stakeholders* possam entendê-lo;

**Validação dos requisitos:** esta etapa examina a especificação do software, de forma a assegurar que todos os requisitos foram definidos sem ambigüidades, inconsistências ou omissões, e que todos os erros foram detectados e corrigidos.

Em paralelo com estas atividades está o processo de **gerência dos requisitos**, voltado ao endereçamento de modificações nos requisitos. A gerência de requisitos busca manter registro das modificações e assegura que elas ocorram de forma controlada no documento de requisitos.

### 2.1.3. Dificuldades

A maior dificuldade na construção de um software é decidir precisamente o que construir. Nenhuma outra parte do trabalho conceitual é mais difícil quanto estabelecer detalhadamente os requisitos técnicos. Nenhuma outra parte é mais difícil de ser corrigida tardiamente [BER98]. Embora esta dificuldade tenha sido percebida há mais de 10 anos, ainda hoje ela está presente no processo de software [PRE01][MCC96][SAW99]. A engenharia de requisitos é a área da engenharia de software onde a distância entre teoria e prática está mais evidente [BER98].

A Tabela 1 apresenta uma compilação das principais dificuldades relatadas na literatura da área de engenharia de requisitos, detalhadas ao longo da seção.

Dificuldade	Autores
Identificação dos Stakeholders	[SHA99][POU99][GOT95][DAM02]
Ambigüidade e falta de clareza	[IEE98][OSB96][KOG00]
Rastreamento dos Requisitos	[GOT97][RAM95][DAV95]
Dificuldades de comunicação	[NUS00][CAR99][DAM03]
Diferenças culturais	[DAM03][PET97]

**Tabela 1 - Relação entre dificuldades da engenharia de requisitos e autores referenciados.**

#### A) Identificação de Stakeholders

Durante o processo de ER, costuma-se dedicar pouca atenção à identificação dos *stakeholders* [SHA99]. Esta deficiência já foi percebida por diversos autores [POU99][SHA99][GOT95], onde as abordagens foram criticadas por assumirem que a identificação dos *stakeholders* é óbvia, ou por fornecerem categorias abrangentes que são muito genéricas para uso prático.

A dificuldade de identificação de *stakeholders* torna-se ainda mais crítica em ambientes de desenvolvimento global. Sem a percepção presencial do ambiente onde o

software em desenvolvimento será utilizado, a capacidade de descoberta das pessoas envolvidas é prejudicada [DAM02].

Na Engenharia de Requisitos, os *stakeholders* exercem um papel fundamental na definição dos requisitos. Se forem selecionadas pessoas com pequeno grau de envolvimento, ou um número insuficiente ou excessivo de *stakeholders*, o resultado final pode ser desastroso.

### ***B) Ambigüidade e Falta de Clareza***

Tendo em vista que as especificações de requisitos servirão como base para contratos e desenvolvimento do software em questão, esta especificação de funcionamento do sistema deve ser clara. A especificação dos requisitos de software não deve ser ambígua tanto para aqueles que a criaram quanto para aqueles que a utilizarão. Entretanto, estes grupos freqüentemente não têm o mesmo nível de conhecimento e tendem a descrever os requisitos de forma diferente [IEE98].

Em ambientes geograficamente distribuídos as diferenças de língua e cultura podem facilmente levar a ambigüidades e redução na clareza dos requisitos. A utilização de gírias ou palavras de pouco uso dificulta o entendimento e a correta especificação dos requisitos. Estes erros ainda podem surgir de uma compreensão diferenciada de um termo entre participantes de países diferentes. Além disso, os meios de comunicação utilizados (computadores, videoconferência, teleconferência) em geral não possuem mecanismos que permitam perceber componentes não-verbais na interação. Alternativas, como *emoticons*, em geral são insuficientes ou de pouca eficácia para melhorar a percepção e o entendimento da comunicação [KOG00].

A utilização de linguagem natural para especificação simplifica o entendimento por parte dos *stakeholders*, mas possibilita a expressão de características ambíguas, inconsistentes, e provavelmente de difícil gerenciamento devido ao tamanho [OSB96]. Além disso, a linguagem natural pode ser pouco clara, levando os desenvolvedores a criar um produto com funcionamento incorreto. Por outro lado, a expressão dos requisitos utilizando técnicas mais formais, como máquinas de estados finitos, pode tornar claro para os desenvolvedores, mas será de difícil compreensão por parte dos clientes.

É necessária a definição de uma linguagem de especificação comum ao pessoal técnico e não técnico envolvido no projeto, que contribua com a clareza e a redução de ambigüidades na especificação dos requisitos.

### ***C) Rastreamento dos Requisitos***

O rastreamento dos requisitos [GOT97][RAM95] refere-se à habilidade para descrever e seguir a vida de um requisito em ambas as direções, para frente e para trás. Isto é, desde a raiz, desenvolvimento e especificação (para trás), no subsequente emprego e uso e durante períodos de refinamento e iteração em qualquer das fases relatadas (para frente). É uma técnica fundamental no apoio às diversas atividades do projeto, assegurando que sistemas e software estão em conformidade às mudanças dos requisitos. Entretanto, é citada como uma área problema pelos desenvolvedores.

O rastreamento dos requisitos é necessário para garantir a qualidade do processo de software (mas não suficiente) e relativamente simples (em conceito e implementação), mas mesmo assim muitas empresas não o realizam. A principal justificativa é o esforço necessário para o rastreamento dos requisitos [DAV95], que se manifesta em aumento no custo e no tempo de um projeto.

E ambientes distribuídos, a dificuldade de se rastrear requisitos pode ser ampliada se existirem diferentes tratamentos para os requisitos pelas equipes de desenvolvimento. Determinadas equipes podem utilizar rastreamento, enquanto que outras não, reduzindo (ou eliminando) os benefícios deste processo.

Sem a utilização de mecanismos de rastreamento, podem surgir inconsistências durante o processo de adição, remoção ou substituição de requisitos na ER, que são resultado de enganos ou conflitos entre os requisitos.

### ***D) Dificuldades de Comunicação***

A engenharia de requisitos envolve grande interação social, entre os diversos participantes do processo. O desenvolvimento de um novo software sempre afeta as pessoas (usuários e gerentes) envolvidas, tanto em aspectos comportamentais como processuais. Conseqüentemente, a ER deve ser sensível à forma com que as pessoas percebem e entendem o mundo a sua volta, como eles interagem, e como a sociologia do local de trabalho pode afetar suas ações. Neste sentido, a comunicação entre os participantes do projeto torna-se crítica. A consciência deste tipo de dificuldade no processo de ER tem levado alguns analistas a procurar apoio em outras áreas do conhecimento, tais como psicologia, sociologia e lingüística [NUS00].

A compreensão do ambiente sociológico em que a ER está inserida nem sempre é uma tarefa simples. Se os participantes no processo não analisarem corretamente o ambiente e se

posicionarem de acordo, podem surgir diversas dificuldades de comunicação, provavelmente causando uma especificação incompleta ou incorreta, e a um produto final não satisfatório.

Em caso de ambientes geograficamente distribuídos, as dificuldades de comunicação se acentuam, uma vez que a comunicação entre os grupos não ocorre pessoalmente, e sim através de ferramentas de e-mail, teleconferência e trocas de mensagens, por exemplo.

Interações mais frequentes levam a uma maior credibilidade das pessoas. As pessoas de maior crédito são aquelas mais acessíveis ou disponíveis. Enquanto equipes de mesma localização podem ampliar a credibilidade através de interações formais e informais, a distância é um fator impeditivo na construção de relações de credibilidade [CAR99]. A tarefa da conversa no cafezinho é importante novamente, uma vez que à distância é difícil tornar-se uma equipe. O contato pessoal, o conhecimento das personalidades e valores é necessário para obter o engajamento, credibilidade e o espírito de equipe [DAM02].

#### *E) Diferenças Culturais*

A diversidade cultural existente na maioria das empresas atuais pode gerar problemas na etapa de ER. Os diferentes valores, expectativas e dinâmicas podem levar a desentendimentos entre os participantes do processo. Por outro lado, toda esta diversidade, se corretamente conduzida, introduz visões e possibilidades que poderiam não ser percebidos em ambientes de cultura homogênea. Os resultados obtidos da interação em ambientes culturalmente diversos podem ser surpreendentes [PET97].

Em ambientes distribuídos, as dificuldades causadas pelas diferenças culturais são ampliadas, devido à distribuição geográfica em questão. Por exemplo [DAM02], para algumas culturas, a estabilidade é muito importante. Então, ao solicitar requisitos para um novo release, eles podem solicitar alguns requisitos puramente porque eles estavam em uma versão anterior do software. Clientes de outros ambientes culturais, entretanto, podem solicitar características completamente novas, apenas porque querem estar atualizados e em progresso na sua abordagem tecnológica. Estes valores culturais conflitantes influenciam na ordem, priorização e negociação dos requisitos.

Esta dificuldade dependendo de como é tratada, pode introduzir elementos prejudiciais ou benéficos, isto torna a heterogeneidade cultural um fator importante a ser explorado em estudos futuros.

## **2.2. DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE**

Nos últimos anos, o software se tornou um componente vital de negócios. O sucesso de uma organização cada vez mais depende da utilização do software como um diferencial

competitivo. Ao mesmo tempo, a economia tem convertido os mercados nacionais em mercados globais, criando novas formas de competição e colaboração [HER01].

Entretanto, o mercado global de software vem passando por diversas crises. Por um lado, um grande número de falhas em projetos. De outro, uma demanda crescente, atingida pela escassez de recursos capacitados. Nesse contexto, muitas organizações perceberam o desenvolvimento distribuído de software como uma alternativa, experimentando o desenvolvimento em locais remotos.

Atualmente um grande número de organizações realiza desenvolvimento distribuído de software, e o assunto está cada vez mais presente em congressos, workshops e grupos de pesquisa internacionais, como o *International Workshop on Global Software Development*<sup>1</sup>, o *Global Software Development Working Group*<sup>2</sup> e o grupo de pesquisa MuNDDoS<sup>3</sup>.

Esta seção apresenta uma revisão bibliográfica em desenvolvimento distribuído de software. Inicialmente são apresentados os principais conceitos envolvidos. Em seguida são descritas as razões que levam ao uso do DDS. Ao final, são apresentados os principais desafios impostos pelo desenvolvimento distribuído de software.

### **2.2.1. Conceitos**

O desenvolvimento distribuído de software, considerada sua recente aplicação e grande abrangência, possui uma diversidade de conceitos que auxiliam sua caracterização. Dentro do contexto de desenvolvimento distribuído de software, foram aplicados conceitos de equipes globais [MAR01] e organizações virtuais [KAR98], por exemplo, ampliando a extensão do conhecimento envolvido.

Considerando a diversidade conceitual atingida pelo DDS, esta subseção reúne a visão de diversos autores [CAR99][KAR98][MCC96][MAR01][PRI02b] com relação aos principais conceitos envolvidos no desenvolvimento distribuído de software, oferecendo um embasamento inicial do assunto.

#### **Desenvolvimento Distribuído de Software (DDS)**

O desenvolvimento distribuído de software caracteriza-se pela distância física e/ou temporal entre alguns elementos (cliente, usuário e desenvolvedores, por exemplo) envolvidos no processo de desenvolvimento [PRI02b].

---

<sup>1</sup> <http://gsd2004.uvic.ca/>.

<sup>2</sup> <http://gsd2004.uvic.ca/gsd/home.htm>.

<sup>3</sup> <http://www.inf.puers.br/munddos/>.

### ***Outsourcing***

*Outsourcing* é a prática de contratar uma organização externa para desenvolver um software, ao invés de desenvolvê-lo na própria sede (*in-house*)[MCC96].

### **Desenvolvimento Global de Software (DGS)**

O desenvolvimento global de software (*Global Software Development - GSD*) ocorre quando a distância física entre os elementos de um DDS envolve mais de um país.

### ***Offshore outsourcing***

*Offshore outsourcing* é uma opção do *outsourcing* que tem se tornado bastante popular nos últimos anos. Organizações *offshore* são empresas localizadas em outro país, geralmente oferecendo um custo menor de desenvolvimento [MCC96][PRI02b].

### ***Offshore insourcing***

O *offshore insourcing* é caracterizado pela contratação de uma subsidiária da própria organização, localizada em outro país [PRI03b].

### **Organização virtual de desenvolvimento de software**

Uma organização virtual de desenvolvimento de software é caracterizada por realizar partes de um projeto em locais distintos, comportando-se como se estivesse no mesmo local [KAR98].

### **Equipes globais**

Uma equipe global, de acordo com Marquardt [MAR01], refere-se a um grupo de pessoas de diferentes nacionalidades que trabalham unidas em um projeto comum, através de culturas e fusos horários distintos, por um extenso período de tempo.

### **Equipes Globais de Desenvolvimento de Software**

Uma equipe global de desenvolvimento de software (*Global Software Team*) é distribuída em dois ou mais países, colaborando ativamente em um projeto comum de desenvolvimento de software [CAR99].

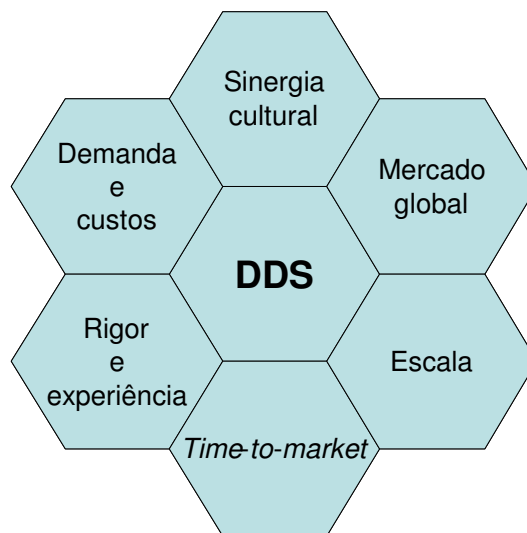
## **2.2.2. Razões que levam ao DDS**

O desenvolvimento de software costumava ser realizado apenas por pessoas com altíssimo grau de especialização, trabalhando em centros de processamento de dados de países avançados [CAR99]. Hoje, entretanto, o desenvolvimento de software, cada vez mais, ocorre de forma distribuída. Em 2000, 203 das empresas americanas na *Fortune 500*<sup>4</sup> realizavam *outsourcing offshore* [CAR01].

---

<sup>4</sup> Fortune 500 é o ranking anual das 500 maiores empresas nos Estados Unidos, promovido pela revista Fortune (<http://www.fortune.com/fortune/>).

Não podemos citar apenas uma razão para a aplicação do desenvolvimento distribuído de software, existem diversas. Essas razões, ou um subconjunto delas, motivam um crescente número de organizações a desenvolverem software de forma distribuída. Nesta subseção são apresentados em detalhe alguns fatores que motivam o uso de DDS nas organizações. A Figura 2 identifica as principais razões envolvidas no DDS.



**Figura 2 - Principais razões envolvidas no DDS**

#### ***A) Demanda e custos***

A demanda por serviços de software tem superado historicamente a disponibilidade de pessoas que os realizam (Figura 3). Como conseqüência, o custo dos profissionais de desenvolvimento de software cresceu, conforme as empresas competiam por suas contratações [KAR98]. No mercado americano, responsável por grande parte da demanda, as cotas de vistos de trabalho temporário para área de tecnologia se esgotavam antes do final do ano fiscal, gerando uma maior escassez de profissionais [CAR99]. Dessa forma, com um número de funcionários insuficiente e custos de contratação altíssimos, a disponibilidade de recursos equivalentes em outras localidades a um custo mais baixo, tornou-se um grande atrativo para as empresas de software, motivando o desenvolvimento distribuído.

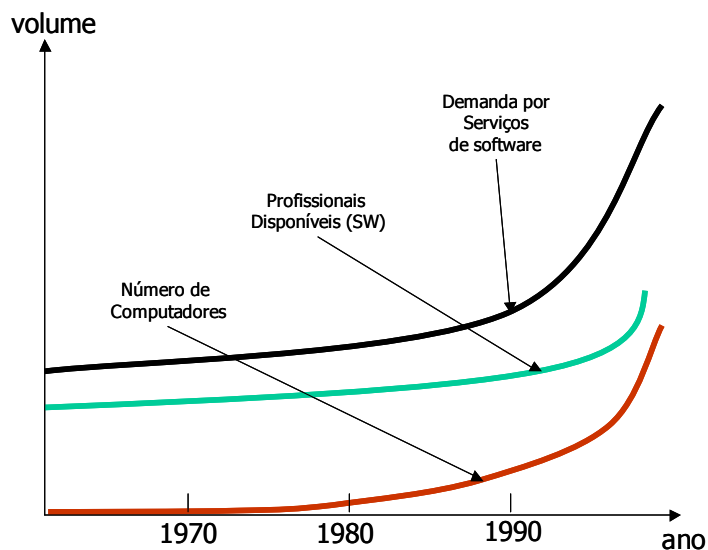


Figura 3 - Demanda por software em relação ao número de microcomputadores e profissionais disponíveis (adaptado de [KAR98])

#### **B) “Time-to-market”**

As pressões para reduzir o tempo necessário para colocar um produto em mercado (*time-to-market*) também auxiliam no crescimento do DDS [HER01]. A possibilidade de desenvolvimento *follow-the-sun* é um grande atrativo para empresas que visam reduzir o *time-to-market*. No desenvolvimento *follow-the-sun* equipes são distribuídas ao redor do globo, de forma que há sempre pelo menos uma equipe disponível realizando o trabalho [MAR01]. Dessa forma, uma desvantagem do DDS (distância geográfica) transforma-se em uma vantagem competitiva, permitindo trabalhar 24 horas por dia [CAR99].

#### **C) Mercado e presença global**

Outro fator importante é o crescente mercado de software externo à América do Norte e Europa. Conforme os custos se reduzem e o poder computacional aumenta, o mercado global de informática cresce, apresentando grande demanda por soluções de software [KAR98]. Em conjunto, para melhor satisfazer o mercado consumidor, a presença das corporações se torna necessária para venda, projeto e manutenção do software. Dessa forma, muitas empresas optam pelo DDS para atingir o mercado global e ficarem próximas a seus consumidores. Além disso, obtém-se o diferencial de se tornar uma empresa global, um bom atrativo de marketing [CAR99].

#### **D) Rigor e experiência no desenvolvimento**

Equipes de desenvolvimento co-localizadas tendem a utilizar mais de mecanismos informais, descuidando no uso de metodologias formais de desenvolvimento e práticas de



qualidade. Equipes de DDS, por outro lado, na tentativa de melhorar a comunicação entre os locais de desenvolvimento, tendem a melhorar significativamente a documentação utilizada [CAR99].

Além disso, em muitos casos, determinados locais desenvolvem experiência e habilidade em áreas pouco difundidas em outros pontos da organização. Nestes casos, a experiência de um local específico pode ser um diferencial para que o desenvolvimento seja realizado nele.

#### ***E) Sinergia cultural***

A diversidade amplia a criatividade e a inspiração na organização de desenvolvimento de software. Uma equipe global cria uma sinergia cultural, encontrando novas formas de resolver problemas, projetar produtos, ou pensar sobre os processos de desenvolvimento [CAR99]. Além disso, a sinergia cultural, ligada à demanda e aos desafios envolvidos, amplia a capacidade de aprendizado da organização [MAR01].

#### ***F) Escala***

Conforme os centros de desenvolvimento de software aumentam, tornam-se grandes e difíceis de gerenciar. Com isso, faz-se necessário distribuir o desenvolvimento de forma a atender a demanda [CAR99].

### **2.2.3. Desafios ao desenvolvimento distribuído de software**

O desenvolvimento de software sempre se apresentou de forma complexa. Existem uma série de problemas e desafios inerentes ao processo [PRI02]. O DDS ao acrescentar fatores como dispersão física, distância temporal e diferenças culturais, acentuou alguns dos desafios existentes e acrescentou novos desafios ao processo de desenvolvimento.

Nesta seção são apresentados os principais desafios gerados pelo DDS, de acordo com a visão de diversos autores. A Tabela 2 apresenta uma compilação dos principais desafios e questões envolvidas com o DDS.

<b>Desafio</b>	<b>Questões envolvidas</b>
Culturais	Liderança
	Formação de grupos
	Estilo de comunicação
	Resolução de problemas
	Perspectiva de tempo
Dispersão geográfica	Gerenciamento
	Fusos Horários
	Legislação
Coordenação e controle	Interdependência
	Gerenciamento de Configuração
	Consciência
Comunicação	Contexto
	Meio
Espírito de equipe	Coesão
	Confiança
	Tamanho

**Tabela 2 - Principais desafios ao desenvolvimento distribuído de software**

#### **A) Culturais**

O gerenciamento da diversidade cultural é fundamental para a efetividade de uma equipe distribuída, principalmente em âmbito global. A cultura, consciente ou inconscientemente, forma os valores, percepções e o comportamento de um grupo, bem como de cada um de seus membros [MAR01]. Culturas diferem em muitas dimensões críticas, como a necessidade de estrutura, atitudes com relação à hierarquia, senso de tempo e estilos de comunicação [HER01]. Dessa forma, membros de uma equipe podem ver a realidade de forma diferente, cada um acreditando que sua própria percepção é a mais correta. O desafio de equipes globais é atingir um equilíbrio entre encorajar o livre fluxo de idéias, e o controle das diferenças culturais entre seus membros [MAR01].

Questões como liderança, por exemplo, costumam fazer com que o processo decisório em equipes distribuídas seja mais demorado ou complexo [KAR98]. Líderes de algumas culturas costumam tratar a equipe de forma participativa e democrática, encorajando a manifestação de opiniões. Em contraste, líderes de outras culturas podem tratar a liderança de forma mais hierárquica, tomando decisões sem consultar os subordinados. Pode haver uma hierarquia clara baseada em status, como família, idade, sexo ou títulos, desencorajando os subordinados a expressarem suas opiniões, por questões de respeito [MAR01].

Problemas relacionados à formação de grupos também são comuns em ambientes de DDS. Enquanto algumas culturas valorizam a independência e os direitos individuais, outras são coletivistas, subordinando os interesses do indivíduo para o bem do grupo [MAR01]. Essas diferenças podem ser fonte de conflitos e rancor entre as equipes [FAV01].

A comunicação entre membros de um grupo pode ser praticada de diversas formas, dependendo de sua cultura. O estilo de comunicação pessoal, por exemplo, pode variar de expressivo a instrumental [MAR01]. Um estilo de comunicação *expressivo* preocupa-se com o estabelecimento e manutenção de conexões pessoais e sociais, deixando a precisão da comunicação em segundo plano. As pessoas são encorajadas a expressar emoções. Abraços e toques são comuns. Em comunicações de estilo *instrumental*, a preocupação maior é com o que está sendo dito. Este estilo de comunicação é mais impessoal, centrado em problemas e orientado a objetivos. A precisão é mais importante que um formato apropriado.

A resolução de problemas também pode ser vista de forma diferente de acordo com a cultura: pode ser *linear*, onde problemas são dissecados em pequenas partes ligadas em cadeias de causa e efeito, ou *sistêmico*, onde problemas são vistos de forma mais holística, com relacionamentos e caos como componentes comuns [MAR01]. Além disso, a cultura difere na forma com que as soluções são feitas. Algumas culturas buscam soluções genéricas, com a aplicação consistente de regras, procedimentos e generalização. Outras culturas têm ênfase em diferenças, relativismo e individualidade. A solução depende totalmente da situação e do contexto.

Diversas outras diferenças culturais podem afetar o DDS, como a forma de tratar conflitos, tempo, flexibilização, entre outros [CAR99][KAR98][MAR01]. Todos estes fatores devem ser cuidadosamente analisados, de forma a minimizar seu impacto no andamento dos projetos.

### **B) Dispersão geográfica**

A tarefa de gerenciamento de grupos adquire maior complexidade com a dispersão [CAR99]. A distância física e psicológica é considerada a segunda maior dificuldade em equipes distribuídas [MAR01]. Embora a tecnologia permita que as organizações optem pela descentralização, ainda existem diversas limitações devido ao reduzido contato face a face entre os membros da equipe e entre a equipe e seus contatos externos.

Obter consenso e aprovação na tomada de decisões pode ser mais difícil. Encontrar membros distantes do projeto, especialmente quando trabalham para diferentes organizações, e obter aprovação adequada de todos os locais envolvidos pode consumir bastante tempo [HER03][LAY00].

Além da dispersão física, a dispersão temporal exerce efeitos sobre o desenvolvimento distribuído de software. Membros de uma equipe estão dispersos no tempo quando há diferença nos horários de trabalho, fusos horários, e/ou ritmos de trabalho que diminuem o

tempo disponível para interação síncrona (isto é, ao mesmo tempo). Por exemplo, equipes separadas no eixo terrestre leste-oeste têm menor número de horas de trabalho coincidentes que equipes separadas no eixo norte-sul, tornando mais difícil a coordenação e comunicação. Mesmo equipes co-localizadas podem ser separadas no tempo se seus membros trabalham em diferentes turnos [ESP03]. A dispersão temporal também afeta a comunicação devido aos estados físicos e mentais dos participantes. Elementos em um local podem estar iniciando o dia, enquanto outros estão no final do expediente.

Equipes distribuídas globalmente também podem sofrer com o sentimento de isolamento. Viagens constantes e ausências demandadas pelas atividades da equipe podem ser desmoralizantes e prejudicar a convivência familiar. Equipes globais podem operar vinte quatro horas por dia ao redor do mundo, com decisões importantes ocorrendo durante a madrugada de alguns países envolvidos. Dias de trabalho em algumas regiões (como sábados e domingos no golfo arábico) podem ser dias de descanso em outras [MAR01].

Outras questões importantes dizem respeito às diferenças legais entre os locais envolvidos (principalmente em casos de distribuição global). Essas diferenças implicam em maior complexidade no estabelecimento de contratos, bem como na necessidade de cautela com questões de sigilo e propriedade intelectual [KAR98].

A co-localização e a dispersão estão ligadas ao antigo paradigma da centralização e descentralização das organizações. A centralização concentra a tomada de decisão em um único ponto da organização. De acordo com antigas noções de gerenciamento, a descentralização da autoridade leva a perda de controle e aumento de custos, por exemplo, em telecomunicações. Com novas tecnologias e redução nos custos, estes problemas foram amenizados, mas não removidos [CAR99].

### ***C) Coordenação e controle***

Coordenação (integração das tarefas e unidades organizacionais de forma que o esforço da equipe contribua para o objetivo geral) e controle (o processo de adesão a metas, políticas e padrões) têm ainda maior importância em equipes distribuídas. As dificuldades e desafios na coordenação e controle em ambientes distribuídos são ampliados devido aos problemas de cultura, língua e tecnologia [CAR99][MAR01].

Equipes distribuídas apresentam dificuldades nos mecanismos de coordenação e controle, principalmente os informais. Devido à distância, o gerenciamento não pode ser feito por observação enquanto se caminha pelo local de trabalho, ou reunindo a equipe em reuniões informais [CAR99][MAR01].

O grau de dependência das tarefas exerce um papel fundamental na coordenação. Quando dois membros de uma equipe, com tarefas fortemente relacionadas, colaboram, a diferença de fuso horário pode dificultar a coordenação. Não poder utilizar o telefone para resolver rapidamente questões simples pode desacelerar o progresso do grupo. Frequentemente as requisições não são claras, necessitando de novas comunicações. Quando membros da equipe estão trabalhando face a face, a clarificação necessária pode ser obtida quase instantaneamente. Entretanto, quando os membros da equipe estão distantes, a clarificação pode introduzir demoras [ESP03].

Conforme a necessidade de coordenação em equipes distribuídas de desenvolvimento de software aumenta, cresce a carga em todas as formas de comunicação, do telefone ao software de gerenciamento de projetos [CAR99].

O custo de coordenar o trabalho aumenta quando tarefas são novas ou incertas, ou quando as unidades de trabalho tornam-se mais interdependentes. Isso significa que conforme a dependência entre as equipes aumenta, a necessidade de coordenação também aumenta [CAR99], conforme Figura 4.

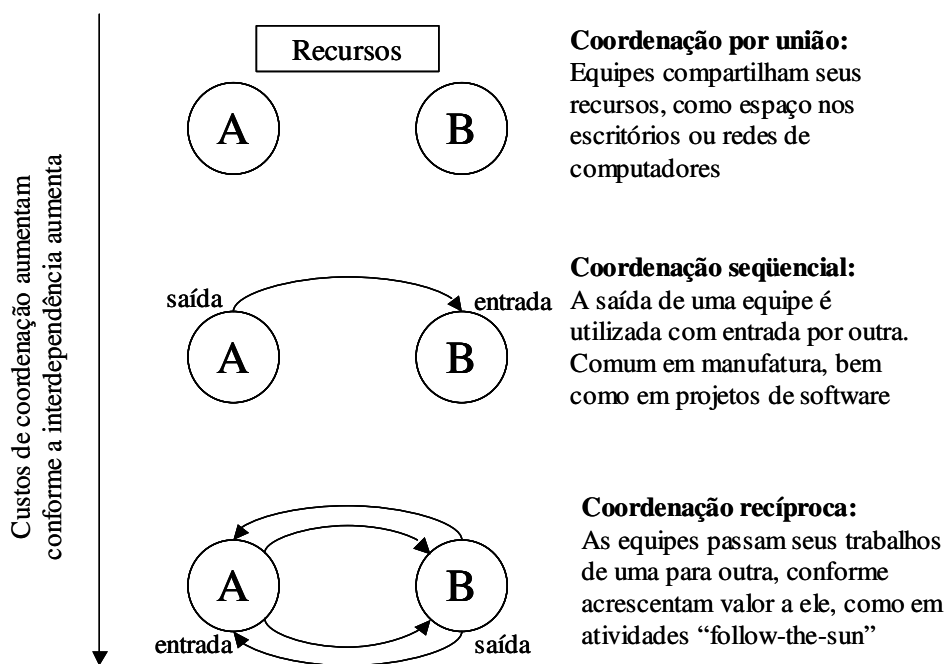


Figura 4 - Custos de coordenação e interdependência (adaptado de [CAR99] [THO76])

Outro fator que torna a coordenação e o controle mais difíceis é o tamanho das equipes distribuídas. Uma equipe distribuída é, em geral, maior, para uma mesma tarefa, que equipes co-localizadas. Quanto mais pessoas e papéis envolvidos, mais desafiadores os problemas de coordenação e controle [MAR01].

O gerenciamento da configuração de software (*Software Configuration Management - SCM*) também apresenta novos desafios. Controlar modificações nos artefatos em cada uma das localidades e coordenar o processo de modificação destas localidades com o processo de desenvolvimento de todo o produto pode ser bastante complexo. A manutenção da consistência de padrões, como formato de documentação também é mais difícil em ambientes distribuídos. Além disso, coordenar e tornar disponíveis as diferentes versões do software em um tempo reduzido pode ser uma tarefa desafiadora [KAR98].

A consciência das atividades desenvolvidas também apresenta desafios ao desenvolvimento distribuído de software. No desenvolvimento de software é importante que se tenha consciência sobre o que está acontecendo, quem está realizando e onde está acontecendo, principalmente sobre dois itens: documentação e código [DAM03]. Participantes necessitam saber os resultados dos outros, de forma a ampliar a colaboração [KOB03].

#### ***D) Comunicação***

A dispersão geográfica tem impacto direto em todas as formas de comunicação. A comunicação informal é reduzida drasticamente. As pessoas deixam de se comunicar devido às dificuldades impostas. Mais reuniões são necessárias, e a complexidade de coordená-las torna-se maior [CAR99].

Comunicação clara e efetiva é absolutamente essencial para o sucesso de equipes distribuídas. Entretanto, em muitos casos é necessário comunicar-se indiretamente (devido à distância temporal), prejudicando a riqueza de contexto [MAR01]. Comunicações podem variar de alto a baixo contexto, e diferentes culturas comunicam de forma diferente de acordo com a necessidade de contexto.

Culturas de alto contexto (ou ricas em contexto) têm a habilidade de compartilhar experiências e tornar inteligíveis determinadas mensagens sem precisar declarar explicitamente. Regras de comportamento e expressão estão implícitas no contexto. O sentido de uma frase não está somente nas palavras. Ele pode estar no tom de voz, linguagem corporal, expressões faciais, contato visual ou uso do silêncio, por exemplo. O significado tende a ser implícito, indireto, não literal [MAR01].

Culturas de baixo contexto, por outro lado, enfatizam a troca de fatos e informações. A mensagem é mais importante que a forma. A informação está primariamente nas palavras, e o significado é explícito. Nessas culturas, videoconferência e e-mails são usualmente aceitos como substitutos eficientes para comunicações face a face.

Da mesma forma, os meios de comunicação podem ser classificados de acordo com o nível de interação. Meios de comunicação ricos permitem interações nos dois sentidos envolvendo mais de um canal sensorial [CAR99]. A Figura 5 apresenta os diversos níveis de interação contextual.

Dados os diferentes estágios do ciclo de desenvolvimento de software, algumas tarefas necessitam de comunicação mais rica que outras. O contato com o cliente deve ser o mais próximo de face a face possível durante a elicitação dos requisitos. Análise e projeto de software necessita de meios ricos para que a colaboração ocorra. De forma geral, qualquer tarefa que necessite de intensa cooperação requer mais comunicação, e quanto mais rica, melhor [CAR99].

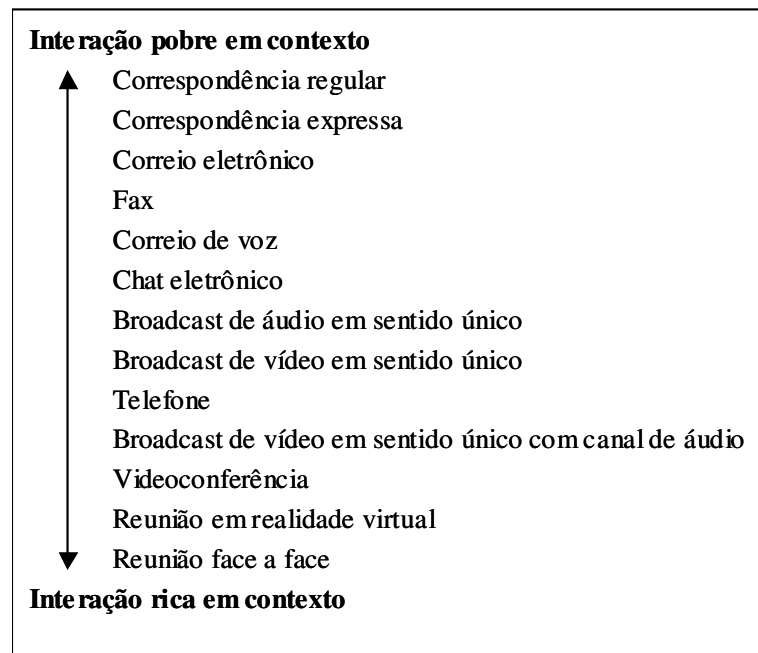


Figura 5 - Níveis de interação contextual [MAR 2001]

Determinadas tarefas exigem muito cuidado na escolha do meio de comunicação a ser utilizado. Por exemplo, um gerente de desenvolvimento distribuído de software deve, regularmente, transmitir a visão de equipe para todos os grupos e culturas envolvidos, de forma que seja entendida. Uma vez que a “visão” deve utilizar níveis altos de motivação e emoção, um meio de comunicação adequado a essas características deve ser utilizado [CAR99].

Alguns dos impactos da dispersão são: o uso excessivo de comunicação, falta de comprometimento e desconforto ao utilizar alguns meios [LAY00].

### ***E) Espírito de equipe***

A palavra equipe, em geral, tende a subentender co-localização, homogeneidade cultural, confiança, padrões de comunicação, um pequeno número de membros e, mais importante, coesão. Equipes são unidades sociais frágeis, que podem facilmente ser quebradas [MAR01]. Quando apresentadas dificuldades como distância e diferenças culturais e de fuso horário, o espírito de equipe geralmente desaparece. O espírito de equipe é o efeito sinérgico que torna a equipe uma unidade coesa [CAR99].

Compartilhar uma visão das metas e ações da equipe pode ser difícil inclusive em ambientes com uma única cultura. Em ambientes globais, multiculturais, é difícil até mesmo afirmar que todos os membros entendem o que é uma equipe. Algumas culturas têm pouca experiência com trabalho em equipe, em algumas línguas a palavra equipe nem mesmo existe [MAR01].

A coesão é uma característica fundamental o sucesso das equipes. Equipes coesas têm maior motivação, moral e produtividade, além de melhor comunicação e satisfação com o trabalho. No DDS a coesão é mais difícil de ser obtida, por diversos motivos. Pessoas podem desconfiar de outras devido à formação de estereótipos incorretos, falta de comunicação externa ao grupo, ou mesmo pelos problemas causados pela diferença de linguagem [CAR99].

A confiança, fundamental para o espírito de equipes, torna-se de difícil manutenção no DDS. A construção desse tipo de relacionamento é impedida pela distância [CAR99]. Equipes co-localizadas adquirem confiança uns nos outros através de encontros informais e face a face. Como esse tipo de encontro não pode ser realizado devido a distância, muitas organizações investem em viagens para integração da equipe.

Além disso, equipes globais são, comumente, maiores que equipes co-localizadas, reduzindo o espírito de equipe. A intimidade que surge da co-localização é perdida, e a complexidade da comunicação aumenta.

O tamanho do grupo é importante para assegurar que a comunicação de todos os membros ocorra de forma efetiva. Quanto menor número de membros de uma equipe, menor o número de canais de comunicação necessários. Com nove membros, uma equipe precisa gerenciar 36 canais de comunicação, que crescem de forma geométrica com a adição de membros ao grupo (Figura 6). Por isso é importante que as equipes sejam pequenas, preferencialmente não contendo mais de dez pessoas [CAR99].



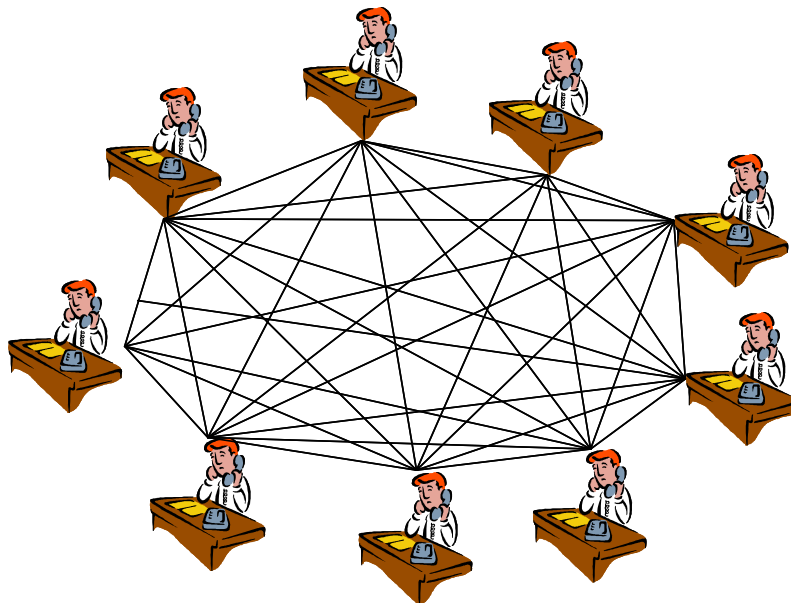


Figura 6 - Número de canais de comunicação em uma equipe.

## 2.3. TRABALHOS RELACIONADOS

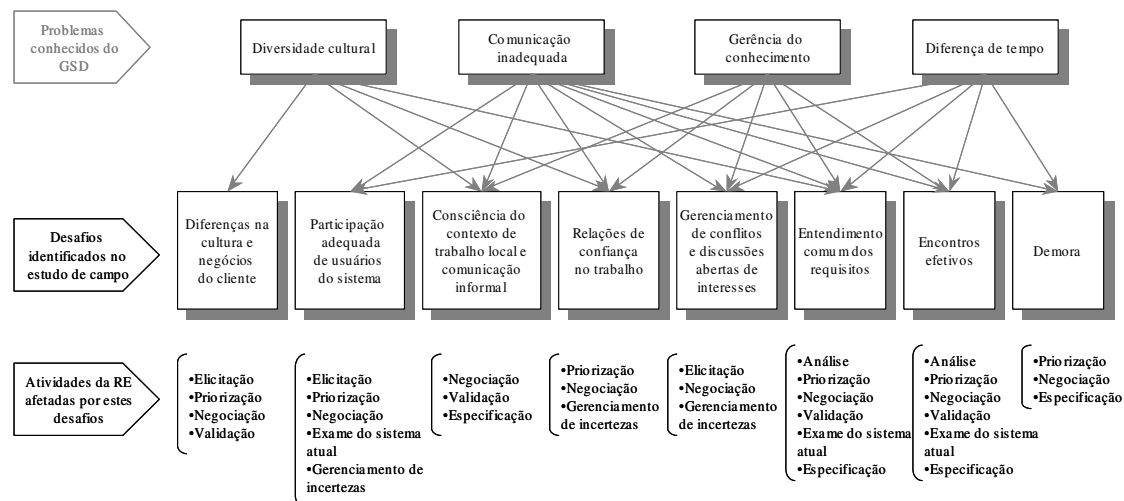
O desenvolvimento distribuído de software apresenta algumas características que o diferenciam fundamentalmente do desenvolvimento co-localizado [ZOW02]. A engenharia de requisitos possui diversas tarefas que necessitam de alto nível de coordenação e comunicação, o que tende a ampliar as dificuldades quando em ambientes distribuídos[DAM02][ZOW02].

Embora diversos estudos reconheçam a necessidade de ampliar o conhecimento em engenharia de requisitos em ambientes distribuídos, após uma pesquisa extensa, um número reduzido de artigos no tema foi encontrado. Nesta seção é apresentada uma compilação dos principais artigos relacionados à engenharia de requisitos em ambientes distribuídos.

### 2.3.1. Abordagem de Damian e Zowghi (2002)

Os estudos de Damian e Zowghi têm como foco principal o entendimento e descrição do impacto exercido pela distância na negociação de requisitos de software, compreendendo grande parte do conhecimento atual sobre engenharia de requisitos em ambientes de desenvolvimento global de software. Em um artigo recente [DAM02], foi apresentada a relação entre cultura, conflitos e distância na negociação de requisitos de forma distribuída globalmente. Para isso foi conduzida uma pesquisa baseada em estudos de caso, visando esclarecer o impacto causado pela distribuição dos *stakeholders* nas atividades de engenharia de requisitos em ambientes de DGS.

Como resultado foi construído um modelo dos desafios apresentados pela distribuição dos *stakeholders* à engenharia de requisitos, conforme apresentado na figura 5. A camada superior do modelo apresenta os quatro maiores problemas da distribuição geográfica dos *stakeholders*: comunicação inadequada, gerência do conhecimento, diversidade cultural e diferença temporal.



**Figura 7 - Modelo de impacto dos desafios e atividades afetadas da engenharia de requisitos devido a problemas com o DGS (adaptado de [DAM02]).**

Esses problemas se alinham com estudos anteriores de DGS [CAR99]. A segunda camada da figura mostra as dificuldades específicas encontradas, decorrentes dos problemas identificados. Na terceira camada, são apresentadas as atividades da engenharia de requisitos afetadas por cada uma das dificuldades encontradas.

O estudo, além da construção do modelo, pôde obter também diversos resultados, relacionados com conflito, cultura e distância.

### 2.3.2. Abordagem de Lloyd, Rosson e Arthur (2002)

Este artigo [LLO02] apresenta um estudo empírico sobre como ferramentas de *groupware* podem ser utilizadas no auxílio da engenharia de requisitos em ambientes distribuídos. Os principais objetivos do estudo foram: identificar os fatores que levam os grupos a produzirem uma especificação de requisitos de software (*Software Requirements Specification* - SRS) de alta qualidade; avaliar a efetividade do *groupware* utilizado como suporte as atividades de engenharia de requisitos; e avaliar a efetividade das diversas técnicas de elicitação de requisitos quando utilizadas em ambientes distribuídos.

O estudo foi realizado com grupos de estudantes universitários, divididos nos papéis de clientes e engenheiros, realizando especificação remotamente. Como suporte, foi utilizado um conjunto de ferramentas de *groupware*.

Através da correlação entre pesquisa, observação e medição da qualidade dos documentos produzidos, os grupos participantes foram classificados em grupos de alta e de baixa performance.

Os grupos classificados como baixa performance tenderam a reclamar mais sobre a falta de participação dos clientes no processo de engenharia de requisitos (56% dos membros do grupo). Os grupos classificados como alta performance registraram menor volume de reclamações (12% dos membros do grupo). A percepção de participação dos membros do grupo pôde ser relacionada, embora fracamente, de forma direta com a qualidade do SRS.

Os resultados desse estudo sugerem que a engenharia de requisitos em ambientes distribuídos é mais efetiva quando os *stakeholders* participam ativamente nas atividades síncronas do processo de requisitos.

O estudo aponta diversas possibilidades de trabalho futuro, ressaltando que a engenharia de requisitos em ambientes distribuídos é uma grande área para pesquisa futura.

### **2.3.3. Abordagem de Prikladnicki, Audy e Evaristo (2003)**

O artigo escrito por Prikladnicki, Audy e Evaristo [PRI03] apresenta resultados iniciais de um estudo de caso do processo de gerência de requisitos no DGS, sob um contexto SW-CMM (*Capability Maturity Model*) [PAU93].

O método de pesquisa utilizado foi estudo de caso, com o acompanhamento de dois projetos em uma organização de DGS. Os projetos foram estudados sob o ponto de vista do centro de desenvolvimento localizado no Brasil. A organização utiliza como base para o processo de desenvolvimento de software o MSF (*Microsoft Solutions Framework*) e o SW-CMM. O estudo tinha como objetivo identificar os problemas, vantagens e desvantagens da gerência de requisitos em projetos geograficamente distribuídos.

A implementação do nível 2 do SW-CMM no centro de desenvolvimento brasileiro, deu origem a algumas diferenças de processo com relação às equipes de outras localidades, que não estavam envolvidas na certificação. Isto ampliou o tempo de algumas atividades, devido à necessidade de explanações a respeito do processo. Por outro lado, a implementação do SW-CMM auxiliou significativamente no processo de gerência dos requisitos e padronização das atividades.

Como impactos do DGS no gerenciamento de requisitos, concluiu-se que essa pode ser uma tarefa árdua se os processos de software não estiverem bem definidos e se as equipes não estiverem preparadas para trabalhar neste cenário. O trabalho envolvendo o SW-CMM nível 2 foi considerado favorável e auxiliou a minimizar alguns dos problemas encontrados. O

treinamento em aspectos não técnicos auxiliou na redução de problemas com cultura e confiança, por exemplo. As lições aprendidas com os estudos de caso estão descritas na Tabela 3.

Nº	Lição
1	Treinar a equipe em assuntos como confiança, cultura, comunicação e colaboração, por exemplo, é essencial;
2	A padronização do trabalho é obrigatória;
3	Encontros freqüentes com pessoas geograficamente distantes são importantes para rastrear o projeto;
4	Um processo bem definido é a chave para o sucesso;
5	Se possível, viagens devem ser feitas para que as equipes envolvidas se conheçam;
6	A diferença de fuso horário pode ser, ao mesmo tempo, uma vantagem e uma desvantagem;
7	O uso de ferramentas como correio eletrônico, teleconferência e videoconferência é muito importante;
8	Um processo de certificação como SW-CMM nível 2 pode ampliar o <i>overhead</i> ;
9	O modelo SW-CMM nível 2 auxiliou a definir uma forma padrão de trabalho;
10	É muito importante conhecer as pessoas com quem se está trabalhando, considerando as formas de comunicar, as diferenças culturais, etc.

**Tabela 3 - Lições aprendidas (adaptado de [PRI03])**

#### **2.3.4. Abordagem de Mahemoff e Johnston (1998)**

O artigo de Mahemoff e Johnston [MAH98] embora não trate explicitamente do impacto do DDS na engenharia de requisitos, apresenta um estudo interessante quanto ao relacionamento da internacionalização de software e engenharia de requisitos. No estudo é realizada uma revisão da bibliografia existente sobre o assunto e proposta a utilização de um repositório de informações culturais como forma de auxílio à internacionalização de software. Os pontos principais estão destacados na seqüência.

Em geral, o desenvolvimento de software direcionado a diversas culturas envolve dois estágios: preparar o sistema principal de forma a permitir o processo de internacionalização (chamado de *localization-enabling*), e incluir os dados específicos de localização na estrutura internacionalizada (*localization*). Quando apenas as características dependentes das necessidades específicas dos usuários são adaptadas, o esforço de desenvolvimento é otimizado. As demais partes do sistema são comuns a todas as versões.

Separar os requisitos específicos de uma cultura dos requisitos que se aplicam a todas as versões é uma tarefa complexa. O autor propõe categorizar os fatores culturais, permitindo a criação de um repositório central de informações culturais, de forma a auxiliar no processo.

No estudo, os fatores foram divididos em visíveis e ocultos, e classificados em subcategorias, conforme apresentado na Tabela 4.

<b>Categoria</b>	<b>Subcategoria</b>	<b>Exemplos</b>
<b>Visível</b>	Fatores de tempo	Calendário
	Problemas de escrita	Conjunto de caracteres, direção do texto
	Problemas de linguagem	Uso de jargão, ordenação de palavras
	Medidas	Moedas, unidades de comprimento
	Formatação	Números, data e hora, arredondamento
<b>Oculto</b>	Sistemas externos	Tamanho padrão de página
	Disposição mental	Alguns grupos qualificam software com base na usabilidade, outros em utilidade
	Percepção	O ícone de pastas pode não ter o mesmo significado em países onde documentos são armazenados em caixas de papelão
	Regras de interação social	Não é educado “bipar” no Japão, pois chama a atenção para um possível erro do usuário
	Contexto de uso	Terminais ATM ( <i>Automatic Teller Machines</i> ) na Suécia são construídos com botões grandes para permitir o uso por pessoas vestindo luvas

**Tabela 4 - Fatores culturais visíveis e ocultos (adaptado de [MAH98])**

Atualmente, pouco trabalho tem sido feito para documentar esse tipo de informação. Muitos textos de internacionalização focam em problemas de implementação ao invés de conceitos de mais alto nível, como usabilidade. Um repositório de informações poderia ser útil para desenvolvedores, projetistas, criadores de protótipos e testadores, por exemplo, adquirindo ainda mais importância para casos em que não fosse possível contratar um consultor local.

### **2.3.5. Abordagem de Zowghi (2002)**

Zowghi [ZOW02] apresenta um estudo cuja questão central é identificar se o desenvolvimento global de software necessita de um processo de engenharia de requisitos diferenciado. O impacto do DGS no processo de engenharia de requisito é descrito, conduzindo à resposta da questão apontada, conforme apresentado na seqüência.

A engenharia de requisitos é a atividade mais rica em comunicação do desenvolvimento de software, uma vez que neste estágio ocorre a maioria das interações entre as pessoas que possuem o problema a ser resolvido e as pessoas que resolverão o problema. A distância geográfica entre os locais de desenvolvimento tem um impacto direto em todas as formas de comunicação, dessa forma, afetando diretamente a engenharia de requisitos.

A distância também afeta mecanismos de comunicação e controle que são ingredientes centrais para o gerenciamento efetivo das equipes. Durante o processo de engenharia de requisitos, os participantes devem buscar obter uma visão comum sobre os requisitos do

sistema a ser construído. A distância não somente desacelera este processo, como dificulta a obtenção de consenso.

Outro desafio para engenharia de requisitos que é influenciado pela distância é a gestão de conhecimento. Informações de diversas origens devem ser compartilhadas com todos os *stakeholders*. Muito do conhecimento que necessita ser compartilhado é tácito e não documentado, principalmente nas fases iniciais da engenharia de requisitos.

A diferença de fuso horário afeta as atividades de elicitação, negociação e priorização da engenharia de requisitos. Demora é introduzida durante o processo, devido à dificuldade de participação dos *stakeholders*. Além disso, ainda existe o impacto causado pelas diferenças culturais. Equipes com diversas culturas tem um potencial maior para produtividade, bem como para problemas [CAR99].

Dessa forma, percebe-se que questões sociais são o centro de muitos dos problemas na engenharia de requisitos e como não pode ser endereçados somente pelos métodos atuais, novas abordagens e paradigmas devem ser procurados. Dessa forma, a resposta para a questão “O desenvolvimento global de software necessita de um processo de engenharia de requisitos diferenciado?” é claramente “Sim”.

### 3. MÉTODO DE PESQUISA

Após ampla revisão teórica, percebeu-se que o problema apresentado ainda não foi abordado sob a mesma perspectiva. Assim, esta pesquisa se caracteriza como um estudo predominantemente exploratório. Segundo Yin [YIN01], a pesquisa exploratória tem como principal finalidade desenvolver, esclarecer e modificar conceitos e idéias, com vistas à formulação de novas teorias, modelos e hipóteses pesquisáveis em estudos posteriores. A pesquisa exploratória muitas vezes constitui-se na primeira etapa de uma investigação mais ampla, que é o caso deste estudo.

Nesse sentido, o método de pesquisa utilizado é o estudo de caso, adotado conforme proposto por Yin [YIN01]. Foram desenvolvidos dois estudos de caso em um centro de desenvolvimento de software de uma multinacional de grande porte. Por tratar-se de uma pesquisa qualitativa, devem-se ter claras as limitações, principalmente quanto à generalização dos resultados obtidos.

#### 3.1. DESENHO DE PESQUISA

O desenho de pesquisa contempla os componentes básicos de uma pesquisa qualitativa, quais sejam: objetivo, unidade de análise e critérios para interpretar os resultados (protocolo de análise). O objetivo geral da pesquisa foi propor um modelo de processo de engenharia de requisitos visando tratar questões relacionadas ao desenvolvimento distribuído de software.

Na seqüência são detalhadas as principais fases do desenho de pesquisa, conforme a Figura 8.

**Fase 1:** nesta fase foi realizado um extenso estudo da base teórica nos assuntos relacionados, como engenharia de requisitos, desenvolvimento distribuído de software, qualidade de software e melhoria de processo de software. Esta fase foi fundamental para formação de uma base conceitual consistente para a continuidade do estudo.

**Fase 2:** após a consolidação de uma base teórica consistente no assunto, foi definido um processo preliminar de engenharia de requisitos para ambientes de DDS. O processo foi definido apoiado pela teoria estudada, bem como em estudos empíricos realizados anteriormente no PPGCC (dissertações de mestrado, estudos de caso, etc.).

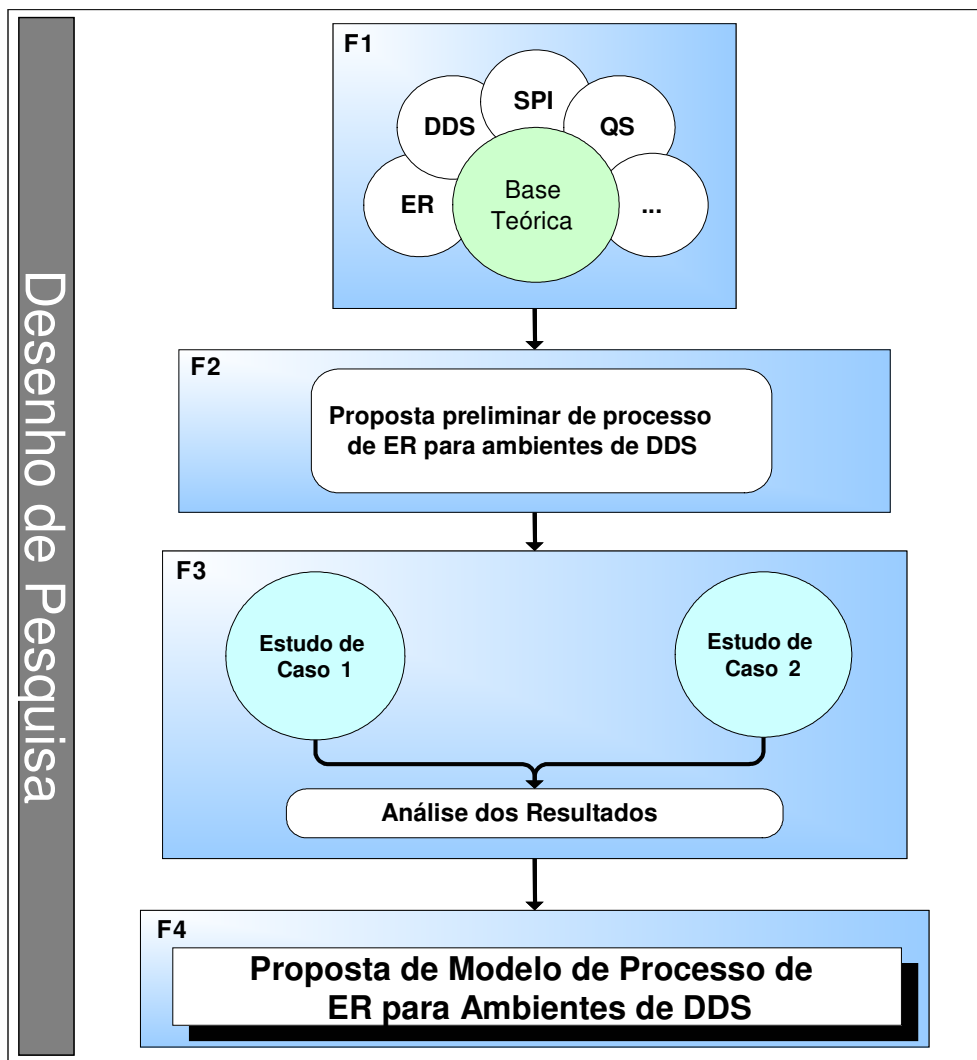


Figura 8 - Desenho de pesquisa

**Fase 3:** nesta fase foram desenvolvidos dois estudos de caso. O primeiro estudo de caso visou avaliar a percepção dos envolvidos no desenvolvimento de software em relação à utilização do processo preliminar. Um segundo estudo de caso foi conduzido buscando capturar a percepção de engenheiros de requisitos em relação aos problemas da engenharia de requisitos em ambientes de DDS. Seu principal objetivo foi a identificação da influência da distribuição das equipes na especificação de requisitos.

**Fase 4:** a etapa final do desenho de pesquisa é a elaboração de uma proposta de modelo de processo de engenharia de requisitos para ambientes de desenvolvimento distribuído de software. Esta proposta foi elaborada com base nos resultados dos estudos de caso, à luz da teoria existente.



## **3.2. ESTUDOS DE CASO**

Durante o processo de pesquisa foram conduzidos dois estudos de caso visando explorar o problema escolhido, bem como obter dados da aplicação do processo preliminar de engenharia de requisitos. A descrição dos estudos de caso conduzidos é realizada a seguir.

### **3.2.1. Estudo de Caso 1**

De acordo com o desenho de pesquisa, foi conduzido o primeiro estudo de caso, visando avaliar a percepção dos envolvidos na aplicação do processo preliminar de engenharia de requisitos para ambientes de desenvolvimento distribuído de software. Inicialmente foi desenvolvido um protocolo de estudo de caso, definindo objetivo, escopo, unidade de análise, procedimento, dimensões e questões (Apêndice 1 - Protocolo de estudo de caso - Estudo de caso 1).

A unidade de análise do estudo de caso 1 é composta de projetos de desenvolvimento de software que utilizaram o processo preliminar de engenharia de requisitos para ambientes de DDS. Nesse sentido, foi selecionada uma organização que realiza desenvolvimento global de software, de forma a acompanhar projetos de desenvolvimento de software desde seu início. A organização selecionada trabalha com manufatura e suporte de computadores, possuindo três unidades de desenvolvimento de software localizadas em dois continentes, responsáveis por atender a demanda de projetos internos da mesma em âmbito mundial. Também é considerada uma empresa de grande porte para o seu segmento e a sua matriz está localizada nos Estados Unidos. O processo de desenvolvimento de software é baseado no MSF, e em metodologias conhecidas, como RUP (*Rational Unified Process*) e PMI (*Project Management Institute*). A unidade de desenvolvimento onde foi conduzido o estudo é reconhecida como uma organização SW-CMM nível 2.

Foram selecionados dois projetos na organização para utilização do processo preliminar de engenharia de requisitos para ambientes de DDS, doravante chamados de Projeto 1 e Projeto 2. Estes projetos seguiram o processo preliminar durante toda a engenharia de requisitos. Após todos os envolvidos terem contato com a versão final dos artefatos de requisitos, foi aplicado um questionário de avaliação.

Como instrumento de coleta de dados, foi utilizado um questionário composto principalmente de questões em escala Lickert de cinco níveis. O instrumento de coleta de dados passou por validação de face e conteúdo por dois pesquisadores (doutores), além de um gerente de projetos da empresa, passando por refinamentos decorrentes das contribuições obtidas na validação de face e conteúdo, até sua versão preliminar.

Um pré-teste foi conduzido com dois líderes técnicos da organização selecionada, sobre a versão preliminar do instrumento, de forma a descobrir os inconvenientes, eliminar equívocos e ambigüidades e escolher a formulação mais adequada das perguntas para a finalidade da pesquisa. Depois de adequar o instrumento à luz do resultado do pré-teste e valendo-se de sugestões decorrentes das modificações realizadas, foram enviados os questionários aos respondentes por correio eletrônico.

No projeto 1 o questionário foi respondido por sete pessoas envolvidas no processo de desenvolvimento: o gerente de projetos, o analista de aplicação, o líder técnico, um desenvolvedor e três testadores. Neste projeto, apenas um dos desenvolvedores teve acesso a todos os artefatos de requisitos utilizados, e por este motivo ele foi o único respondente nesse papel. Os demais utilizaram como base para o desenvolvimento somente o documento de projeto do software (*Software Design Document*), não estando aptos a responder o questionário.

No projeto 2 o questionário foi respondido por seis pessoas: o gerente de projetos, o analista de aplicação, o arquiteto do sistema e três desenvolvedores. Neste projeto o analista de aplicação também realizou o papel de líder técnico. O teste desta aplicação foi considerado um projeto separado e foi realizado em outro país. Embora o questionário tenha sido enviado para o líder de teste, não foi obtido retorno.

Após o recebimento das respostas dos questionários enviados, foi realizada a análise de resultados. Para isso, foram utilizados a técnica de análise de conteúdo e o módulo estatístico do Excel, triangulando-se os dados do questionário com documentos de projeto, de forma a obter maior confiabilidade nos resultados.

### **3.2.2. Estudo de Caso 2**

Um segundo estudo de caso foi conduzido, de acordo com o desenho de pesquisa, visando capturar a percepção de engenheiros de requisitos com experiência no desenvolvimento distribuído de software com relação aos problemas que a distribuição tende a ampliar na especificação de requisitos. Um protocolo de estudo de caso foi desenvolvido de forma a guiar a condução do estudo (veja Apêndice 2 - Protocolo Para Estudo De Caso 2).

Foi selecionada uma empresa que realiza desenvolvimento global de software, para que fossem entrevistados os engenheiros de requisitos de forma a obter sua opinião com relação aos problemas na especificação de requisitos causados por fatores relacionados ao DDS. A organização selecionada no estudo de caso 2 foi a mesma escolhida no estudo de caso 1. Foram estabelecidos quatro principais cenários, de acordo com a distribuição dos grupos de

engenheiros de requisitos, usuários, clientes e equipe de desenvolvimento. O instrumento de coleta de dados utilizado foi um artefato possibilitando relacionar os principais fatores da engenharia de requisitos no desenvolvimento distribuído de software com os problemas na especificação de requisitos. Os fatores da engenharia de requisitos no DDS foram obtidos com base na teoria existente [DAM00][DAM02][DAM03][LLO02][MAH98][ZOW02] e nas categorias e fatores apresentado no capítulo 4. A taxonomia para problemas na especificação foi obtida com base no estudo de Shull [SHU00].

O instrumento de coleta de dados passou por validação de face e conteúdo por dois pesquisadores. Com base nas considerações realizadas foram feitas alterações no instrumento. Em seguida, foi conduzido o pré-teste com um engenheiro de requisitos da organização em estudo. Após estabelecimento da versão final do instrumento de coleta de dados, deu-se início às entrevistas.

Seis engenheiros de requisitos da organização foram entrevistados. Todas as entrevistas foram gravadas, possibilitando consulta posterior. Os entrevistados, durante o período previamente agendado, relacionavam os principais fatores do desenvolvimento distribuído de software com os problemas nos requisitos, de forma a apontar os problemas ampliados por cada fator.

Após a conclusão das entrevistas, foi realizada a análise de resultados. Para isso, foram utilizados a técnica de análise de conteúdo e o módulo estatístico do Excel, relacionando os dados obtidos com a literatura no assunto.

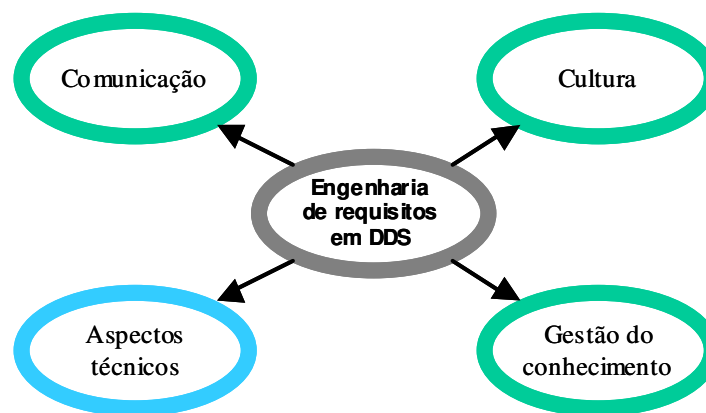
## 4. CATEGORIAS E FATORES ASSOCIADOS À ENGENHARIA DE REQUISITOS NO DDS

Neste capítulo, são identificadas as principais categorias e fatores relacionados à engenharia de requisitos em ambientes de desenvolvimento distribuído de software, com base na teoria existente no assunto [DAM02][LLO02][MAH98][PRI03].

Em uma primeira análise do tema, foram identificadas as categorias comunicação, cultura e gestão do conhecimento. Entretanto, observando a divisão realizada por Carmel [CAR99] da forças centrífugas e centrípetas em equipes globais de software, podemos perceber que as forças centrífugas relacionam-se a aspectos não técnicos, e as forças centrípetas, em sua maioria, a aspectos técnicos.

Traçando um paralelo com as abordagens apresentadas, podemos perceber um domínio de estudos sobre aspectos não técnicos na engenharia de requisitos em ambientes de DDS. Considerando a escassez de estudos que abordam a influência de aspectos técnicos, decidiu-se incluir, além dos aspectos não-técnicos (comunicação, cultura e gestão do conhecimento), uma categoria voltada aos aspectos técnicos da engenharia de requisitos em DDS.

Desse modo, foram identificadas quatro categorias, conforme a Figura 9. Cada uma das categorias foi estudada em maior detalhe, destacando seus principais fatores, de acordo com a influência apresentada pelas abordagens.



**Figura 9 - Categorias relacionadas à engenharia de requisitos em DDS**

Essas categorias compreendem diversos fatores e se relacionam de diversas maneiras, sendo, muitas vezes, difícil de se definir a fronteira entre cada uma. Outros fatores poderiam

ter sido relacionados, mas para efeito desse estudo, foram considerados apenas os principais, devido à importância que adquiriram nas abordagens estudadas. A análise visa encontrar e detalhar os fatores que devem ser endereçados por um modelo de processo de engenharia de requisitos para ambientes distribuídos.

#### 4.1. COMUNICAÇÃO

O processo de engenharia de requisitos depende largamente da comunicação entre os envolvidos. Nas diversas abordagens estudadas, a comunicação aparece como um ponto crítico para o sucesso da engenharia de requisitos em ambientes distribuídos. Dentre os fatores relacionados com a comunicação destacam-se idioma, diferença de fuso-horário e meio de comunicação, conforme exposto na Figura 10.

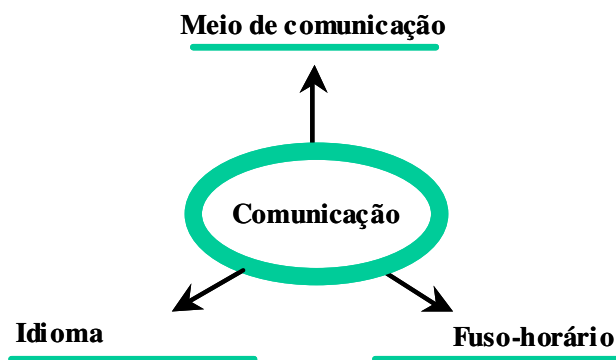


Figura 10 - Fatores associados à comunicação

Ambientes de DDS comumente atingem nível global, dessa forma o idioma utilizado pelas equipes passa a exercer influência na engenharia de requisitos em ambientes DDS. Quando a comunicação envolve grupos que possuem idiomas nativos diferentes, a probabilidade de erros de interpretação cresce. Ao tratar com requisitos, onde a clareza é fundamental, a diferença de idiomas entre os grupos introduz um risco maior de erros. Por outro lado, quando as equipes possuem o mesmo idioma nativo, a comunicação torna-se mais fácil e eficaz.

A diferença de fuso horário entre equipes também exerce efeito na comunicação, com impacto na engenharia de requisitos. Quando a diferença de fusos horários é muito grande, as equipes podem possuir pouco ou nenhum horário de trabalho em comum, dificultando ou impedindo o uso de ferramentas de comunicação síncrona como videoconferência. O uso de correio eletrônico nesses casos é comum. Entretanto, questões simples, como o esclarecimento de um requisito pode levar dias para se concretizar se conduzido de forma assíncrona com grupos trabalhando em horários distintos. Em casos com menor diferença de

fuso horário, a comunicação se torna mais livre, permitindo a utilização de mecanismos síncronos e, em geral, acelerando o processo.

O meio de comunicação influencia na engenharia de requisitos em ambientes distribuídos. Quando temos contato face-a-face, utilizamos diversos mecanismos para expressar a mensagem que desejamos. Expressões faciais, gestos, alteração no tom de voz, entre outros, auxiliam na comunicação. Dessa forma, o meio utilizado, dependendo do nível de interação que possibilita, pode afetar a qualidade da comunicação.

## 4.2. CULTURA

A cultura é apontada como uma categoria de grande influência na engenharia de requisitos em ambientes de DDS. Tanto a cultura nacional quanto a organizacional podem afetar a engenharia de requisitos. Com base nas abordagens estudadas, foram selecionados os fatores contexto, atitude e valores, conforme apresentado na Figura 11.

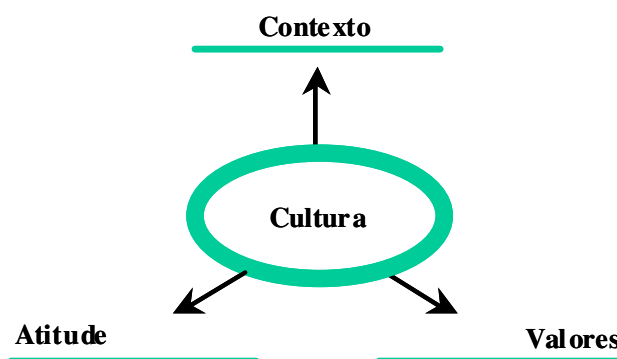


Figura 11 - Fatores associados à cultura

Contexto é um fator cultural com impacto na engenharia de requisitos. A compreensão do contexto onde o software em desenvolvimento será inserido é importante para o correto desenvolvimento do software. Ao abranger culturas diferentes, a dificuldade de entendimento tende a crescer. Em culturas similares, as dificuldades causadas por contexto, em geral, se reduzem.

As atitudes dos envolvidos no processo de requisitos podem variar de acordo com a cultura a que pertencem. Atitudes em relação à hierarquia, pontualidade, formalidade e contato físico, por exemplo, tendem a variar entre culturas. Com isso, pessoas de culturas diferentes podem se sentir incomodadas com as atitudes umas das outras, favorecendo situações de conflito.

Outro fator a ser considerado sob a ótica da cultura são os valores. Culturas diferentes têm valores diferentes, dificultando a priorização e negociação dos requisitos. Algumas culturas valorizam a estabilidade, tendendo a desejar que características existentes de um

sistema sejam mantidas. Outras culturas valorizam inovação, priorizando mais novas funcionalidades do que a manutenção de antigas, por exemplo. Por outro lado, pessoas com valores similares tendem a obter mais facilmente consenso na definição de requisitos.

A cultura organizacional, muitas vezes, auxilia a reduzir as diferenças de cultura nacional entre equipes, ao guiar, mesmo que indiretamente as atitudes e valores de seus colaboradores.

### 4.3. GESTÃO DE CONHECIMENTO

O processo de engenharia de requisitos envolve grande volume de informações. Captar, processar, armazenar e disponibilizar o conhecimento gerado pelo processo de requisitos, bem como unificar a visão organizacional são questões que devem ser endereçadas pela gestão do conhecimento. Os principais fatores identificados em relação a gestão do conhecimento são gerenciamento de papéis, *awareness*<sup>5</sup> e gerenciamento de informações, como apresentado na Figura 12.

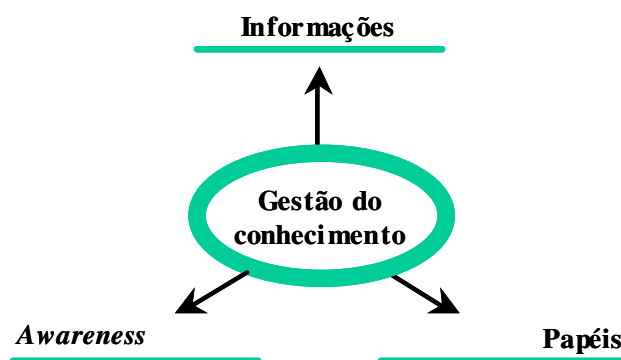


Figura 12 - Fatores associados à gestão de conhecimento

O gerenciamento de informações influencia a engenharia de requisitos em ambientes DDS. A gestão do conhecimento atua de forma transversal às categorias identificadas, auxiliando, por exemplo, nos processos utilizados e na gerência de informações culturais. O processo engenharia de requisitos, por exemplo, envolve documentos cujo conteúdo pode ter diversas origens. A obtenção, processamento e disponibilização dessas informações agilizam o processo de requisitos e auxiliam na obtenção de bons resultados. Além disso, a necessidade de informações culturais relacionadas com o local onde o software a ser desenvolvido será utilizado é importante para definir características desse software. O sistema métrico utilizado em cada país, por exemplo. Essas informações são fundamentais para as equipes envolvidas

<sup>5</sup> consciência, percepção, conhecimento geral sobre as atividades e sobre o grupo.

no processo de engenharia de requisitos poderem analisar as considerações necessárias para adaptação a cada cultura.

Outro fator importante no DDS é a gestão do conhecimento relativo aos papéis e contatos envolvidos no processo de requisitos. Durante o processo é importante o conhecimento dos responsáveis por cada atividade. No desenvolvimento co-localizado esse conhecimento circula de maneira informal. Entretanto, com a distribuição das equipes, há a necessidade de uma forma de disponibilizar esse conhecimento, facilitando a identificação dos papéis e localização dos responsáveis por cada tarefa.

O *awareness*, também relacionado com a gestão do conhecimento, é um importante no DDS. Equipes distribuídas necessitam ter conhecimento do que está acontecendo em cada uma das localidades envolvidas no processo. Mudanças na especificação e alterações nas regras de negócio, por exemplo, devem ser alertadas para as equipes.

#### 4.4. ASPECTOS TÉCNICOS

Diversos aspectos técnicos influenciam a engenharia de requisitos em ambientes distribuídos. O processo de requisitos é afetado por mecanismos de coordenação e controle, por exemplo, que podem auxiliar a reduzir o impacto causado pela distribuição das equipes no processo de requisitos. Os fatores identificados, em relação a aspectos técnicos, foram padrões, processos, *groupware* e gerência de configuração, conforme apresentados na Figura 13.

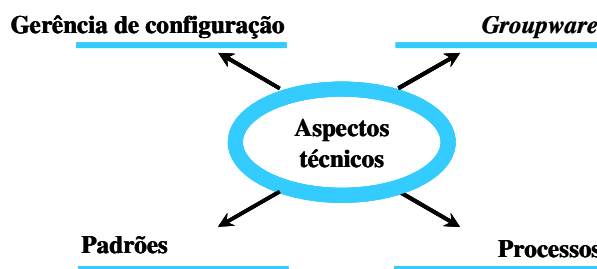


Figura 13 - Fatores associados à aspectos técnicos

Padrões são aspectos técnicos associados com o processo de engenharia de requisitos. Com a necessidade de maior formalização causada pelo desenvolvimento distribuído de software, evoluiu-se em busca de padrões. Modelos de maturidade do processo de desenvolvimento, como o SW-CMM tratam diversas dificuldades comuns no DDS. Ao mesmo tempo, há a necessidade de padronização dos documentos envolvidos no processo, bem como na estrutura dos requisitos e casos de uso utilizados.



A utilização de processos também é um fator técnico associado com a engenharia de requisitos em ambientes distribuídos. A estruturação das atividades das equipes envolvidas na engenharia de requisitos é fundamental para controle do que está sendo realizado. Processos equivalentes para toda organização é uma prática importante, pois evita entendimentos diferentes da ordem das atividades, papéis e responsabilidades de cada um.

O *groupware* é um aspecto técnico que influencia a engenharia de requisitos em ambientes de DDS. O uso de ferramentas de *groupware* pode melhorar a interação entre as equipes e ampliar sua consciência em relação ao andamento às atividades das localidades envolvidas. O *groupware* pode também ser utilizado na elicitação e negociação dos requisitos, por exemplo, como ferramenta auxiliar nessas atividades.

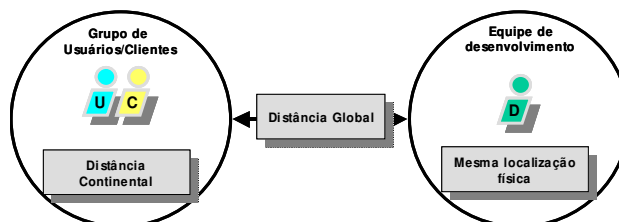
Os diversos documentos utilizados e a necessidade de rastreabilidade fazem da gerência de configuração um fator importante na engenharia de requisitos em DDS. O controle de versões dos documentos utilizados é fundamental para que a equipe trabalhe sobre uma versão consistente do documento de requisitos. Além disso, é necessário que documentos com alterações à especificação inicial estejam disponíveis rapidamente, e de forma consistente para todas as equipes envolvidas. A gerência dos requisitos em ambientes distribuídos adquire novas dimensões, pois, em alguns casos, requisitos de um software são divididos entre diversos locais de desenvolvimento. A manutenção da rastreabilidade é fundamental para evitar dificuldades na identificação da localidade responsável por cada requisito do software.

## 5. PROCESSO PRELIMINAR

Conforme previsto no desenho de pesquisa, foi definido um processo preliminar de engenharia de requisitos para ambientes de desenvolvimento distribuído de software, com base na proposta apresentada em [LOP03]. O objetivo deste processo é reduzir o impacto da dispersão das equipes na engenharia de requisitos. Para isso, são definidos os papéis envolvidos e a forma de evolução dos artefatos de requisitos, buscando o consenso entre os grupos participantes. Também está incluído no processo um modelo de especificação de requisitos em linguagem natural, visando principalmente a redução de ambigüidades e padronização do trabalho entre as equipes. Na seqüência, são descritos em detalhe o contexto onde o processo se aplica, o processo preliminar e o modelo de especificação de requisitos em linguagem natural.

### 5.1. CONTEXTO

O processo preliminar considera a existência de distância física entre membros da equipe de desenvolvimento e do grupo de usuários/clientes. A dispersão entre as equipes pode ser representada utilizando uma adaptação do formato proposto por Prikkladnicki [PRI03b], conforme o exemplo de cenário na Figura 14.



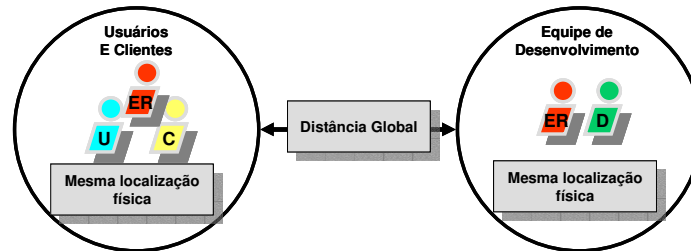
**Figura 14 - Exemplo de cenário de dispersão considerado no processo preliminar**

Os principais grupos envolvidos no processo preliminar são a equipe de engenheiros de requisitos, o grupo de usuários e clientes, e a equipe de desenvolvimento.

A **equipe de engenheiros de requisitos (ER)** é responsável pela elicitacão, análise, negociaçã, documentaçã, validaçã e gerência dos requisitos. Na proposta preliminar a equipe de engenheiro de requisitos possui membros próximos ao grupo de usuários e clientes, chamados de analistas de negócio, e membros próximos à equipe de desenvolvimento, chamados de analistas de aplicaçã, como apresentado na Figura 15.

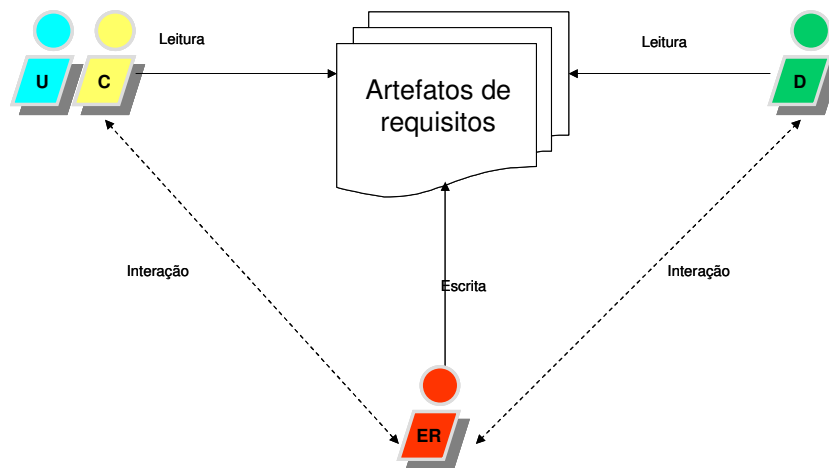
O **grupo de usuários (U) e clientes (C)** representa as pessoas físicas ou jurídicas que solicitaram e contrataram o projeto, bem como os responsáveis pela utilização do produto gerado. Esta equipe fornece informações para especificação do software.

A **equipe de desenvolvimento (D)** representa os envolvidos no desenvolvimento de um determinado projeto, utilizando como entrada os requisitos especificados pelo grupo de engenheiros de requisitos. Esta equipe pode envolver gerentes de projeto, codificadores, testadores, controladores de qualidade, equipe de suporte a ferramentas, entre outros.



**Figura 15 - Distribuição dos engenheiros de requisitos no exemplo de cenário**

Na proposta de processo é considerado um modelo de interação centrado no grupo de engenheiros de requisitos, responsável pelos artefatos de requisitos, conforme apresentado na Figura 16.



**Figura 16 - Interação entre engenheiros de requisitos, usuários, clientes e desenvolvedores**

A forma de interação utilizada no processo preliminar define o grupo de engenheiros de requisitos como intermediários entre o grupo de usuários/clientes e a equipe de desenvolvimento. O grupo de engenheiros de requisitos é responsável pela criação e manutenção dos artefatos de especificação do software, sendo o único grupo com permissão de alterar estes artefatos. Usuários, clientes e desenvolvedores podem avaliar os artefatos de especificação durante todo o processo de engenharia de requisitos, solicitando alterações quando necessário.

Representantes significativos de todos os grupos devem ser engajados desde o início do processo, permitindo que a especificação seja constantemente avaliada e adaptada conforme a necessidade dos participantes.

Todas as alterações nos artefatos de especificação devem passar pelo grupo de engenheiros de requisitos, permitindo que seja o controle do documento seja mantido centralizado neste grupo. Alterações podem ser solicitadas por membros do grupo de usuários/clientes ou membro da equipe de desenvolvimento, sendo avaliada pelo grupo de engenheiros de requisitos e aprovada por todos os representantes.

## 5.2. PROCESSO PRELIMINAR

Conforme apresentado anteriormente, o processo preliminar considera a existência de pelo menos um representante do grupo de engenheiros de requisitos junto ao grupo de usuários/clientes (papel chamado de analista de negócios), e um representante do grupo de engenheiros de requisitos junto à equipe de desenvolvimento (papel chamado de analista de aplicações). O processo consiste em cinco etapas, conforme representado na Figura 17 e detalhado na seqüência.

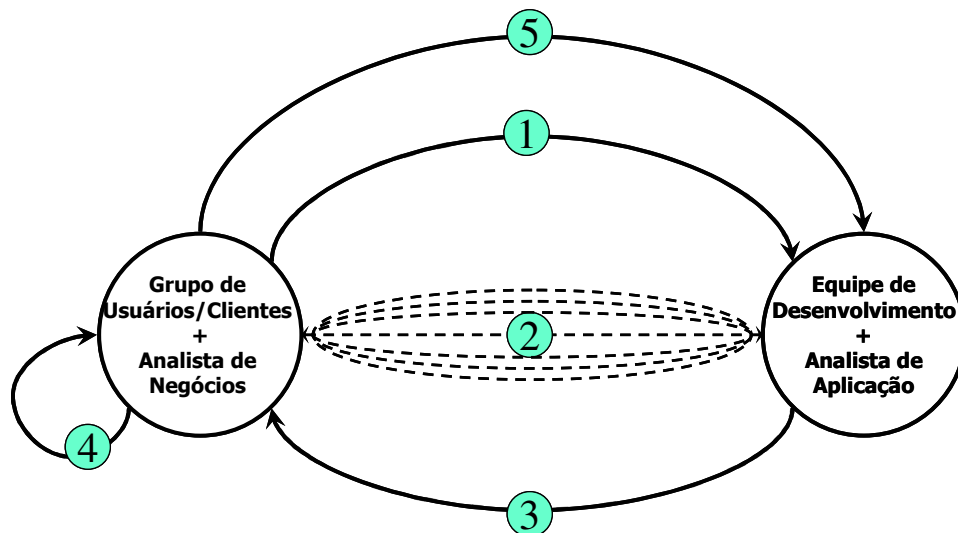


Figura 17 - Processo preliminar de engenharia de requisitos para ambientes de DDS

### **Etapa 1 - Envio dos artefatos de requisitos para a equipe de desenvolvimento.**

Após criar o conjunto inicial de artefatos de requisitos, o analista de negócios envia estes artefatos para o analista de aplicação. O conjunto inicial de artefatos pode ser constituído de documentos de alto nível, como um documento Visão/Escopo, ou mesmo incluir uma versão inicial de um documento de especificação de requisitos. O envio destes artefatos marca

o início da participação da equipe de desenvolvimento e do analista de aplicação no processo de engenharia de requisitos.

### **Etapa 2 - Análise e evolução dos artefatos de requisitos.**

Ao receber os artefatos iniciais de requisitos do analista de negócios, o analista de aplicação busca, em primeiro lugar o entendimento destes artefatos e seu contexto. Representantes da equipe de desenvolvimento também são colocados em contato com os artefatos para contextualização.

Durante esta etapa os artefatos de requisitos são complementados com informações e adaptados para reduzir potenciais fontes de problemas. Ambigüidade e falta de clareza são ainda mais prováveis quando diferenças culturais e de linguagem estão envolvidas. Dúvidas surgem e devem ser esclarecidas com o analista de negócios ou o grupo de usuários/clientes, caracterizando grande volume de comunicação entre as equipes.

A comunicação entre as equipes, além de esclarecimentos, visa obter consenso quanto à especificação em construção, alinhando as múltiplas visões em relação aos requisitos.

### **Etapa 3 - Envio dos artefatos de requisitos para aprovação pelo grupo de usuários/clientes.**

Após a evolução dos artefatos de requisitos, estes devem ser validados pelo grupo de usuários/clientes, assegurando que a especificação construída atende às necessidades e objetivos que motivaram o processo de desenvolvimento.

Nesta etapa o analista de aplicações envia os artefatos de requisitos para aprovação final pelo grupo de usuário e cliente e o analista de negócios.

### **Etapa 4 - Validação dos artefatos de requisitos pelos usuários/clientes.**

O grupo de usuários/clientes deve verificar os artefatos de requisitos, assegurando que eles refletem os objetivos do projeto. Como a participação de todos os grupos ocorreu durante a etapa 2, esta etapa é simplificada, pois o grupo de usuários e clientes já esteve em contato com os artefatos de requisitos durante toda a sua evolução. O mesmo vale para o analista de negócios, que deve aprovar os artefatos de requisitos, verificando se eles contém todas as informações exigidas para o software a ser desenvolvido.

### **Etapa 5 - Versão final dos artefatos de requisitos é definida.**

Após aprovação formal pelos usuários/clientes, os artefatos aprovados formam o conjunto de artefatos de entrada para equipe de desenvolvimento. Qualquer modificação nestes artefatos deve passar por aprovação de todas as equipes.

### **5.3. MODELO DE ESPECIFICAÇÃO DE REQUISITOS EM LINGUAGEM NATURAL**

O processo preliminar inclui um modelo para especificação de requisitos em linguagem natural, visando principalmente a redução de ambigüidades e padronização dos trabalhos entre as equipes. O modelo de especificação de requisitos foi desenvolvido com foco na elicitacão e documentacão de requisitos, visando capturar necessidades e objetivos do grupo de usuários e clientes.

Quando a equipe de desenvolvimento atende a diversas equipes de especificacão, é natural que os documentos de requisitos advindos das várias equipes tenham formatos e padronizacões diferentes. Com isso, questões como o nível de detalhe dos requisitos, formato do documento, glossário e formato de casos de uso e requisitos, por exemplo, afetam diretamente o processo de desenvolvimento e as métricas envolvidas. Com o uso de um modelo de especificacão de requisitos em linguagem natural busca-se padronizar a documentacão de entrada, reduzindo o efeito causado pela diversidade de padrões das equipes de especificacão.

São incluídos neste modelo um meta-modelo de requisitos e uma estrutura textual. O meta-modelo de requisitos é constituído de uma série de definições utilizadas para classificar as informacões obtidas durante a elicitacão, bem como o relacionamento entre elas. A estrutura de texto define estruturas de frase a serem utilizadas para cada classe de informacão, com o objetivo de simplificar a compreensão das informacões representadas.

Esta abordagem foi construída com base em um estudo das principais definições de requisitos [ARM01][GOG96][LEI98][SID96][THA00] e estruturas de frase da literatura [DAM02b][RAT03]. O modelo foi testado preliminarmente utilizando dados históricos de dois projetos de desenvolvimento de software.

## **6. RESULTADOS DOS ESTUDOS DE CASO**

Neste capítulo são apresentados os resultados dos estudos de caso. Nas seções 6.1 é detalhado o estudo de caso 1, incluindo uma descrição do processo preliminar de engenharia de requisitos, a apresentação do instrumento de coleta de dados utilizado, a análise dos dados e as lições aprendidas. A seção 6.2 aborda os resultados do estudo de caso 2, descrevendo o instrumento de coleta de dados, a análise dos dados para cada cenário e as lições aprendidas.

### **6.1. ESTUDO DE CASO 1**

O estudo de caso 1 visava avaliar a percepção de equipes envolvidas na aplicação do processo preliminar de engenharia de requisitos para ambientes de desenvolvimento distribuído, com relação ao processo preliminar e a qualidade do artefato gerado.

Com este objetivo, foi acompanhada a utilização do processo preliminar em dois projetos de desenvolvimento distribuído de software. Após o contato de toda equipe com o documento final de requisitos, foi aplicado um instrumento de pesquisa estruturado, composto principalmente de questões em escala Lickert.

Na seqüência é apresentado o instrumento de coleta de dados e a análise dos dados deste estudo de caso. Ao final, são apresentadas as lições aprendidas.

#### **6.1.1. Descrição do Instrumento de Coleta de Dados**

O instrumento de coleta de dados utilizado no estudo de caso 1 é um questionário composto de sete dimensões, visando avaliar o processo preliminar de engenharia de requisitos para ambientes de desenvolvimento distribuído de software (veja Apêndice 1 - Protocolo Para Estudo De Caso 2).

A primeira dimensão do questionário, relativa aos dados demográficos, tem como objetivo identificar os respondentes, quanto à formação acadêmica e profissional, bem como em relação à função desempenhada na organização em estudo. Na segunda dimensão, é solicitado aos respondentes que descrevessem o processo de engenharia de requisitos utilizado anteriormente na organização.

A terceira dimensão constituí-se de um grupo de questões em escala Lickert de cinco níveis, variando de “Discordo Totalmente” (com peso 1) a “Concordo Totalmente” (com peso 5). Estas questões buscam avaliar o processo preliminar de forma não comparativa. Ou seja, nesta dimensão é solicitado aos respondentes que avaliassem afirmações sem comparar o processo preliminar com os processos que utilizavam anteriormente.

Da mesma forma que a terceira dimensão, a quarta dimensão é composta de questões em escala Lickert. Entretanto, este grupo de questões visa avaliar o processo de forma comparativa. Neste caso, é solicitado aos respondentes que avaliassem as afirmações comparando com sua experiência anterior.

A quinta dimensão, também em escala Lickert, visa capturar a percepção dos respondentes com relação à qualidade da especificação de requisitos (SRS) produzida pelo processo preliminar. Os construtos utilizados foram obtidos no Padrão 830 de 1998 da IEEE (*Institute of Electrical and Electronic Engineers*), definidos como características de um bom SRS.

A dimensão 6 é composta de questões abertas visando identificar as vantagens e desvantagens do uso do processo preliminar de engenharia de requisitos em ambientes de desenvolvimento distribuído de software. A sétima e última dimensão é composta de questões abertas com o objetivo de obter informações sobre os projetos em estudo.

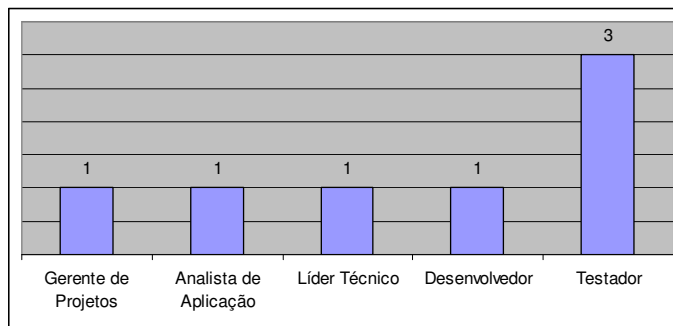
### **6.1.2. Análise de Dados**

A análise dos resultados do estudo de caso foi realizada utilizando o módulo estatístico do Microsoft Excel. Os dados do questionário foram triangulados com documentação do projeto de forma a obter maior confiabilidade nos resultados. Foram também entrevistados alguns membros dos projetos para sanar dúvidas com relação às respostas obtidas no questionário.

#### ***A) Projeto 1***

No projeto 1 foram entrevistados sete participantes, incluindo o gerente de projetos, o analista de aplicação, o líder técnico, três testadores e um desenvolvedor (Figura 18). Destes, todos possuíam o ensino superior completo, sendo que dois deles já haviam concluído mestrado. O tempo de empresa dos respondentes é bastante variado. Quatro dos respondentes estavam há menos de um ano na organização. Os demais respondentes haviam trabalhado entre um e cinco anos na organização em questão. A maioria dos respondentes (cinco) possuía experiência de dois a cinco anos na função que estavam desempenhando. Dois dos respondentes eram vinculados a uma empresa terceira, que presta serviço para organização onde o estudo de caso foi conduzido.





**Figura 18 - EC1 – Número de respondentes em cada função (Projeto 1)**

Com relação ao processo de engenharia de requisitos utilizado anteriormente na organização (dimensão 2 da entrevista), segundo o analista de aplicação, baseava-se no MSF. Os projetos eram entendidos através da leitura da documentação disponível. Em seguida, era realizado o entendimento do escopo, modelagem do sistema e criação do SRS quando este não estava disponível. O gerente de projetos destacou que era utilizada a modelagem de casos de uso. O líder técnico apontou que não havia participado no levantamento de requisitos em projetos anteriores.

O primeiro conjunto de questões em escala Lickert referia-se ao processo preliminar, utilizado no projeto em questão, de forma não comparativa. Ou seja, os respondentes foram instruídos a escolher a opção que melhor representasse a opinião deles, sem comparar com o processo utilizado em projetos anteriores. A consolidação dos resultados para o projeto 1 está representado na Tabela 5.

Questão	Média	Desvio Padrão
1. A escrita dos requisitos é simples utilizando o novo processo.	3,33	1,15
2. Os requisitos escritos utilizando o novo processo são simples de serem entendidos pelos usuários	4,33	0,58
3. A escrita de requisitos com o novo processo permite a captura de todos os tipos de informação necessárias à especificação.	3,00	1,73
4. A escrita de requisitos segundo o novo processo simplifica o entendimento dos requisitos por pessoas cuja língua nativa não é o inglês.	3,67	1,15
5. Ao escrever requisitos utilizando o novo processo, sinto confiança que os requisitos serão entendidos pelos demais envolvidos no processo de desenvolvimento de software (equipe de desenvolvimento, usuários finais, cliente, equipe de teste, etc.)	4,00	0,00

**Tabela 5 - EC1 - Resultados da dimensão 3 no projeto 1.**

Nesta dimensão da entrevista, três das cinco questões tiveram um desvio padrão alto. Ao abordar a simplicidade do processo, abrangência e facilidade de entendimento dos requisitos observamos que não existe consenso entre os respondentes.

Ao questionar se o processo preliminar permite a captura de todos os tipos de informação necessários à especificação, o gerente de projetos avaliou como “Discordo totalmente”, enquanto que as respostas do analista de aplicações e do líder técnico foram “Concordo”. Ao ser questionado sobre a motivação desta resposta, o gerente de projetos justificou a importância de se utilizar diversas formas de representação de requisitos. De acordo com ele, nem todo requisito deve ser representado textualmente, muitas vezes sendo necessário utilizar diagramas e representações formais.

Com a média mais alta da dimensão (4,33), foi avaliado pelos respondentes que o processo utilizado permite a escrita de requisitos mais simples de serem entendidos pelos usuários. Assim, a validação dos requisitos junto aos clientes, foi simplificada. Alinhando-se com esta resposta, todos os respondentes concordaram (4 na escala Lickert) que, com o uso do processo preliminar, a equipe estava confiante que os envolvidos no processo de ER entenderiam os requisitos especificados. A dificuldade de entendimento, comum em ambientes de DDS onde há diferença de linguagem entre as equipes, pode causar a redução da confiança, redução do espírito de equipe, desalinhamento de visões, entre outros. Dessa forma, com a confiança no entendimento da especificação pelos demais envolvidos, reduzimos a probabilidade destas dificuldades ocorrerem.

O próximo conjunto de questões (dimensão 4) era comparativo. Nesta dimensão, foi solicitado aos respondentes que avaliassem as questões comparando com sua experiência anterior em projetos de DDS. Os resultados consolidados para dimensão quatro no projeto 1 são apresentados na Tabela 6.

Questão	Média	Desvio Padrão
1 - As múltiplas interações contribuem para um melhor <b>entendimento das necessidades</b> envolvidas no desenvolvimento do sistema	5,00	0,00
2 - A utilização do novo processo promove uma melhor comunicação, fornecendo às equipes uma <b>linguagem comum</b> para comunicação (com relação à estrutura de frase).	4,67	0,58
3 - A comunicação entre as equipes torna-se mais <b>eficiente</b> utilizando-se o novo processo.	4,00	1,00
4 - Dificuldades de interpretação e ambigüidades originadas de <b>diferenças culturais</b> , ou de <b>contexto</b> , são reduzidas com a utilização do novo processo.	3,67	0,58
5 - O novo processo de engenharia de requisitos simplifica o entendimento de características específicas do <b>contexto</b> onde o software será utilizado.	3,00	1,00
6 - A especificação de requisitos resultante do novo processo é mais detalhada.	4,67	0,58
7 - O novo processo de engenharia de requisitos permite um <b>conhecimento</b> mais claro sobre o sistema a ser desenvolvido.	4,00	1,00
8 - Com a utilização do novo processo a confiança no entendimento dos requisitos entre as equipes aumenta.	4,33	0,58
9 - A especificação dos requisitos tornou-se <b>mais rápida</b> com a utilização do novo processo.	2,67	0,58
10 - Com a utilização do novo processo os requisitos tornaram-se mais <b>claros</b> .	4,00	1,00
11 - O novo processo simplifica a descoberta de requisitos ocultos.	4,00	1,00
12 - Com a utilização do processo é possível se realizar uma melhor <b>estimativa</b> do esforço necessário para os próximos estágios de desenvolvimento.	4,00	1,00
13 - O novo processo traz benefícios para as atividades de <b>teste</b> .	4,33	0,58
14 - O novo processo promove melhorias na <b>documentação</b> do software desenvolvido.	4,33	0,58
15 - O processo de engenharia de requisitos proposto representa um benefício para as seguintes etapas do processo de desenvolvimento:		
- <i>Planning</i>	4,67	0,58
- <i>Developing</i>	4,33	0,58
- <i>Stabilizing</i>	3,67	1,15

**Tabela 6 - EC1 - Resultados da dimensão 4 no projeto 1**

Todos os respondentes concordaram que as interações entre as equipes foram benéficas, permitindo um melhor entendimento das necessidades envolvidas no desenvolvimento do sistema. Ainda em relação à comunicação, os respondentes apontaram que a estruturação das frases na especificação auxiliou a fornecer uma linguagem comum na comunicação, tornando-a mais eficiente.

Os resultados em relação a entendimento de características específicas de contexto, bem como redução das ambigüidades e dificuldades de interpretação causadas por diferenças culturais ou de contexto, foram duas das questões com média mais baixa desta dimensão. Assim, percebe-se que ainda são necessários esforços para melhoria no processo preliminar nesse sentido. A compreensão do ambiente onde os requisitos em especificação serão utilizados, bem como as características culturais que podem afetar o sistema são fundamentais para que o esforço de desenvolvimento não seja desperdiçado, como destacado por [DAM02].

Os respondentes também apontaram que a especificação produzida tende a ser mais detalhada e agregar um maior volume de conhecimento quando realizada utilizando o processo preliminar de ER em DDS. Quando em ambientes distribuídos, a obtenção de informações e esclarecimentos é dificultada. Dessa forma é ainda mais importante a presença de todas as informações relevantes na especificação.

Alinhando-se com o apresentado na dimensão 3, os respondentes tenderam a considerar que a confiança no entendimento dos requisitos entre as equipes aumenta com a utilização do processo preliminar. Uma visão comum dos objetivos do sistema é de difícil obtenção em ambientes de DDS, como apontado por Damian [DAM02]. Ao melhorar a compreensão dos requisitos entre todos os envolvidos, são dados os primeiros passos na direção de reduzir esta dificuldade.

Conforme esperado, os respondentes em média apontaram que o processo utilizado foi mais lento para especificação de requisitos. Ao inserir um maior volume de interações, bem como detalhamento e formato na especificação de requisitos, a tendência à redução na velocidade de especificação era esperada. Entretanto, como apresentado na questão 3 a comunicação foi considerada mais eficiente. De acordo com o analista de aplicação envolvido, a maior demora ocorrida na especificação de requisitos foi, em parte, motivada pela inexperiência da equipe com o processo utilizado. Esta demora pode ser reduzida, de acordo com a curva de aprendizado da equipe.

Com relação à descoberta de requisitos ocultos e a clareza nos requisitos a média das respostas foi quatro, com um desvio padrão um pouco alto (1). Dessa forma, houve uma tendência a avaliar positivamente os benefícios do processo utilizado.

Considerando-se os benefícios obtidos em etapas posteriores do processo de desenvolvimento, como estimativa, teste e documentação do software, as respostas tenderam a ficar acima de quatro, demonstrando que a percepção dos envolvidos foi de uma melhora nesse sentido ao utilizar o processo preliminar. Nas fases do MSF de *planning* e *developing*, as respostas também apontaram contribuições com a utilização do processo. Entretanto, para a fase de *stabilizing*, o desvio padrão foi muito alto, apontando a falta de consenso na resposta.

A quinta dimensão da entrevista consiste em uma avaliação da especificação de requisitos produzida, com relação às características de um bom SRS, de acordo com [IEE98]. Nesta dimensão foram utilizadas afirmações em escala Lickert visando obter a percepção dos envolvidos no projeto em questão com relação à qualidade do SRS. Esta dimensão também é comparativa, sendo solicitado aos respondentes que avaliassem as afirmações com base em sua experiência passada. A consolidação das respostas para esta dimensão está na Tabela 7.

Questão	Média	Desvio Padrão
1 - Com a utilização do novo processo o SRS torna-se mais <b>correto</b> .	4,43	0,79
2 - Com a utilização do novo processo as <b>ambigüidades</b> se reduzem.	4,00	1,15
3 - Com a utilização do novo processo o SRS torna-se mais <b>completo</b> .	3,71	0,95
4 - Com a utilização do novo processo o SRS torna-se mais <b>consistente</b> .	3,33	1,21
5 - Com a utilização do novo processo o SRS é mais facilmente <b>priorizado</b> .	3,43	0,98
6 - Com a utilização do novo processo o SRS torna-se mais <b>verificável</b> .	4,43	0,53
7 - Com a utilização do novo processo o SRS torna-se mais <b>modificável</b> .	3,86	0,90
8 - O SRS é mais facilmente <b>rastreável</b> com a utilização do novo processo.	4,14	0,90

**Tabela 7 - EC1 - Resultados da dimensão 5 no projeto 1**

De forma geral as respostas foram positivas, com média superior a três, indicando que, de acordo com a percepção dos participantes, o SRS foi melhorado em consideração às características definidas pelo IEEE. Quando considerando o atributo “correto”, a média de respostas foi próxima a cinco, com um desvio padrão baixo, indicando que a percepção dos participantes foi de uma melhora em relação a este item.

Considerando a redução de ambigüidades, a média de respostas foi quatro, com o segundo maior desvio padrão da entrevista. Em uma análise aprofundada, os testadores indicaram os menores valores, com uma média de 3, enquanto que a média dos demais foi 4,5. Isto foi clarificado por um testador pela seguinte afirmação: “é importante levar em consideração que devido às características do projeto (reengenharia e recodificação de um sistema existente), o sistema existente foi utilizado como entrada para o desenvolvimento, o que reduziu o nível de detalhe”.

Com relação à afirmação que o SRS se tornou mais completo as respostas foram, em média, 3,71, com desvio padrão de 0,95. Isto indica uma tendência dos respondentes considerarem o SRS produzido mais completo.

Considerando consistência, as respostas tiveram um desvio padrão alto, com uma leve tendência positiva. Uma vez que consistência é relacionada a documentação de mais alto nível [IEE98] e alguns respondentes não tiveram acesso a ela, este resultado não é significativo.

Os melhores resultados foram obtidos na afirmação de que o SRS se tornou mais verificável. A média de respostas foi próxima a cinco (4.43), com o menor desvio padrão na dimensão. Isto indica que, na percepção dos entrevistados, o modelo produziu um SRS mais verificável.

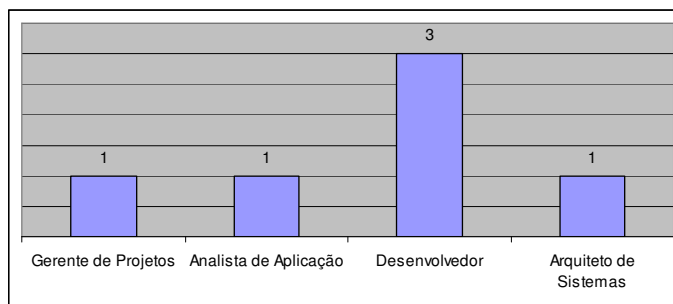
Na afirmação que o SRS se tornou mais modificável as respostas foram, em média, positivas (3,86). O desvio padrão foi 0,9, demonstrando uma tendência a considerar que o SRS tornou-se mais modificável.

Considerando-se rastreabilidade, a média de respostas foi acima de 4, indicando uma melhora. Este resultado deriva principalmente do processo de numeração utilizado neste projeto, baseado no documento de nível superior de forma a deixar explícita a rastreabilidade.

### **B) Projeto 2**

No projeto 2 foram entrevistados seis participantes, incluindo o gerente de projetos, o analista de aplicação, três desenvolvedores e o arquiteto de sistemas (Figura 18). Neste projeto o analista de aplicação também exerceu o papel de líder técnico. A equipe de teste estava distante da equipe de desenvolvimento. Embora a entrevista tenha sido enviada, não foi respondida. O arquiteto do sistema fazia parte da equipe de usuários e clientes.

Apenas um dos seis respondentes ainda não havia finalizado o curso superior. Os demais possuíam o ensino superior completo (2) ou especialização (3). A maior parte da equipe (cinco pessoas) estava há mais de um ano trabalhando na organização. Dos seis respondentes, cinco eram funcionários da organização. Um deles trabalhava para uma empresa sub-contratada pela organização em pesquisa. Quatro dos respondentes possuíam mais de cinco anos de experiência na função que estavam realizando. Apenas um estava há menos de um ano na função realizada no projeto em pesquisa.



**Figura 19 - EC1 - Número de respondentes em cada função (Projeto 2)**

Quando questionado a respeito do processo de engenharia de requisitos utilizado anteriormente pela equipe (dimensão 2 da entrevista), o gerente do projeto explicou que o levantamento de requisitos era realizado através de uma documentação textual dos requisitos. O analista de aplicação ressaltou que, em alguns casos, o SRS era finalizado e enviado apenas para desenvolvimento *offshore*. Quando o desenvolvimento do SRS fazia parte do escopo do projeto da localidade em questão, este era aprovado pelos clientes.

A terceira dimensão da pesquisa visava obter dados sobre o processo preliminar de engenharia de requisitos para ambientes de DDS utilizado pela equipe entrevistada. A síntese dos resultados para esta dimensão está apresentada na Tabela 8.

Questão	Média	Desvio Padrão
1 - A escrita dos requisitos é simples utilizando o novo processo.	4,50	0,71
2 - Os requisitos escritos utilizando o novo processo são simples de serem entendidos pelos usuários	4,50	0,71
3 - A escrita de requisitos com o novo processo permite a captura de todos os tipos de informação necessárias à especificação.	4,00	0,00
4 - A escrita de requisitos segundo o novo processo simplifica o entendimento dos requisitos por pessoas cuja língua nativa não é o inglês.	5,00	0,00
5 - Ao escrever requisitos utilizando o novo processo, sinto confiança que os requisitos serão entendidos pelos demais envolvidos no processo de desenvolvimento de software (equipe de desenvolvimento, usuários finais, cliente, equipe de teste, etc.)	4,50	0,71

**Tabela 8 - EC1 - Resultados da dimensão 3 no projeto 2**

Nesta dimensão a média das respostas foi alta, com questões onde o desvio padrão foi zero. Todos os respondentes concordaram fortemente que o processo utilizado simplifica o entendimento dos requisitos por pessoas cuja língua nativa não é o inglês. Este fator é importante para redução de ambigüidades e alinhamento das visões entre as equipes, conforme [DAM02].

Com relação à simplicidade de entendimento dos requisitos pelos usuários e confiança no entendimento dos requisitos pelos demais envolvidos no processo de desenvolvimento (questões 2 e 5), a média das respostas foi de 4,5, com um desvio padrão de 0,71, apontando que os entrevistados concordam com estas características no processo preliminar em uso no projeto.

A média mais baixa entre as respostas, embora um consenso entre os respondentes, foi de 4, onde todos concordaram que o processo utilizado permite a captura de todas as informações necessárias à especificação.

Ao avaliar se os requisitos são simples de serem escritos com o processo preliminar utilizado no projeto (questão 1) a média das respostas foi de 4,5 com desvio padrão de 0,71, indicando que os entrevistados concordam com a afirmação. Entretanto foi apontado como um problema a curva de aprendizagem do processo, durante a qual é despendido um tempo muitas vezes importante para o andamento do projeto.

No segundo conjunto de questões em escala Lickert (dimensão 4), foi solicitado que os entrevistados avaliassem o processo preliminar de forma comparativa com os projetos anteriores que participaram. A síntese das respostas para dimensão 4 está apresentada na Tabela 9.

Questão	Média	Desvio Padrão
1 - As múltiplas interações contribuem para um melhor <b>entendimento das necessidades</b> envolvidas no desenvolvimento do sistema	5,00	0,00
2 - A utilização do novo processo promove uma melhor comunicação, fornecendo às equipes uma <b>linguagem comum</b> para comunicação (com relação à estrutura de frase).	5,00	0,00
3 - A comunicação entre as equipes torna-se mais <b>eficiente</b> utilizando-se o novo processo.	4,50	0,71
4 - Dificuldades de interpretação e ambigüidades originadas de <b>diferenças culturais</b> , ou de <b>contexto</b> , são reduzidas com a utilização do novo processo.	4,50	0,71
5 - O novo processo de engenharia de requisitos simplifica o entendimento de características específicas do <b>contexto</b> onde o software será utilizado.	3,50	0,71
6 - A especificação de requisitos resultante do novo processo é mais detalhada.	5,00	0,00
7 - O novo processo de engenharia de requisitos permite um <b>conhecimento</b> mais claro sobre o sistema a ser desenvolvido.	4,00	0,00
8 - Com a utilização do novo processo a confiança no entendimento dos requisitos entre as equipes aumenta.	4,50	0,71
9 - A especificação dos requisitos tornou-se <b>mais rápida</b> com a utilização do novo processo.	3,50	0,71
10 - Com a utilização do novo processo os requisitos tornaram-se mais <b>claros</b> .	5,00	0,00
11 - O novo processo simplifica a descoberta de requisitos ocultos.	2,50	0,71
12 - Com a utilização do processo é possível se realizar uma melhor <b>estimativa</b> do esforço necessário para os próximos estágios de desenvolvimento.	3,00	1,41
13 - O novo processo traz benefícios para as atividades de <b>teste</b> .	4,50	0,71
14 - O novo processo promove melhorias na <b>documentação</b> do software desenvolvido.	4,50	0,71
15 - O processo de engenharia de requisitos proposto representa um benefício para as seguintes etapas do processo de desenvolvimento:		
- <i>Planning</i>	5,00	0,00
- <i>Developing</i>	4,50	0,71
- <i>Stabilizing</i>	3,50	0,71

**Tabela 9 - EC1 - Resultados da dimensão 4 no projeto 2**

Todos os respondentes concordaram que as múltiplas interações promovidas pelo processo contribuem fortemente para um melhor entendimento das necessidades envolvidas no desenvolvimento do sistema. Dessa forma, é possível um maior alinhamento da visão das diversas equipes envolvidas no processo.

Com relação à comunicação, da mesma forma que a questão anterior, todos os respondentes concordaram fortemente que ela é melhorada com o novo processo, permitindo, com a estruturação das frases, uma linguagem comum de comunicação entre as equipes. Em conjunto, nota-se que, na percepção dos respondentes, a comunicação tornou-se mais eficiente com o novo processo, questão onde a média foi 4,5, com desvio padrão 0,71.

Os respondentes concordaram que o processo utilizado reduz as dificuldades de interpretação e ambigüidades devido às diferenças culturais e de contexto (média de 4,5, com desvio padrão de 0,71). Entretanto, ao avaliar se o processo utilizado simplifica o entendimento de características de contexto do local onde o software será utilizado, a média



de respostas foi de 3,5, indicando que os benefícios não foram muito significativos nesse sentido.

Um consenso entre os respondentes foi de que o processo utilizado permitiu que se produzisse uma especificação mais detalhada, com uma média 5. Em conjunto, com média 4 e desvio padrão 0, os respondentes concordaram que o processo utilizado permite um conhecimento mais claro sobre o sistema a ser desenvolvido.

Os respondentes concordaram totalmente (média 5) que os requisitos escritos com o processo proposto tornaram-se mais claros. Assim, alinhando-se com o respondido na dimensão 3, a média das respostas da questão 8 foi de 4,5, apontando que, na percepção dos respondentes, a confiança no entendimento dos requisitos entre as equipes é ampliada com a utilização do processo preliminar.

Com relação à velocidade de especificação, a média das respostas foi 3,5, indicando que não houve grande diferença no tempo de especificação com o uso do processo. Este resultado é esperado, considerada a curva de aprendizado necessária, já que esta foi a primeira vez que a equipe utilizava o processo. Além disso, a especificação mais detalhada, conforme apontado na questão 6, tende a demandar mais tempo.

A única questão com média abaixo de 3 foi com relação à descoberta de requisitos ocultos, com média 2,5. Isso indica um provável ponto de melhoria no processo, de forma a auxiliar a descoberta de requisitos ocultos em ambientes de DDS, onde sua obtenção é dificultada.

Com relação a melhorias na estimativa do processo de desenvolvimento a média das respostas foi 3, com desvio padrão de 1,71, indicando falta de consenso no assunto. Assim, são necessários mais dados para melhor identificar a tendência.

Conforme apontado pela opinião dos entrevistados o processo trouxe benefícios para as atividades de teste e documentação (média de 4,5, com desvio padrão 0,71). Ao avaliar as atividades do MSF beneficiadas pelo processo, os respondentes concordaram totalmente (média 5) que o processo trouxe melhorias para a fase de *Planning*. A fase de *Developing* obteve média 4,5, indicando ser também beneficiada pelo processo. Com a média mais baixa entre as fases do MSF, a fase de *Stabilizing* obteve média 3,5, com desvio padrão 0,71, indicando que o processo introduziu pouca ou nenhuma melhoria nesta fase.

Na quinta dimensão, foi solicitado aos respondentes que avaliassem a especificação produzida no projeto com relação às características de um bom SRS, de acordo com a IEEE [IEE98]. A Tabela 10 apresenta a síntese dos resultados para esta dimensão.

Questão	Média	Desvio Padrão
1 - Com a utilização do novo processo o SRS torna-se mais <b>correto</b> .	4,17	0,75
2 - Com a utilização do novo processo as <b>ambigüidades</b> se reduzem.	4,00	0,63
3 - Com a utilização do novo processo o SRS torna-se mais <b>completo</b> .	4,00	0,63
4 - Com a utilização do novo processo o SRS torna-se mais <b>consistente</b> .	3,83	1,47
5 - Com a utilização do novo processo o SRS é mais facilmente <b>priorizado</b> .	3,50	1,38
6 - Com a utilização do novo processo o SRS torna-se mais <b>verificável</b> .	4,17	0,75
7 - Com a utilização do novo processo o SRS torna-se mais <b>modificável</b> .	4,33	0,52
8 - O SRS é mais facilmente <b>rastreável</b> com a utilização do novo processo.	4,00	1,26

**Tabela 10 - EC1 - Resultados da dimensão 5 no projeto 2**

Neste conjunto de questões, todas as médias foram acima de 3, indicando uma melhoria geral na qualidade do SRS no que tange as características definidas pela IEEE.

Com relação ao atributo “correto”, a média das respostas foi de 4,17, com desvio padrão de 0,75. Isso demonstra que os respondentes avaliaram como positiva a contribuição do processo para tornar o SRS mais correto.

Na percepção dos entrevistados, as ambigüidades do SRS são reduzidas com o novo processo. A média de respostas foi 4, com um desvio padrão baixo (0,63), indicando alinhamento entre os respondentes. Com a mesma avaliação, o atributo correto também foi melhorado no SRS. Ambigüidade e informações incorretas são uma fonte de retrabalho, e sua redução contribui para diminuição de custo e tempo no processo de desenvolvimento.

Com um desvio padrão alto (1,47, 1,38 e 1,26 respectivamente) os atributos “consistência”, “priorização” e “rastreadibilidade” tiveram respostas bastante divergentes dentro da equipe, impossibilitando uma melhor avaliação de suas tendências. Se desconsiderarmos a avaliação do gerente de projetos, a média de respostas indicando que o SRS tornava-se mais consistente com o novo processo foi de 4,4, com desvio padrão de 0,55. De forma similar, se eliminarmos a resposta do gerente de projetos na questão sobre rastreadibilidade, a média das respostas torna-se 4,4, com desvio padrão 0,89, indicando melhoria no SRS em relação a este atributo.

Ao avaliar se o SRS produzido pelo processo proposto era mais verificável, a média de respostas foi de 4,17, com desvio padrão de 0,75, indicando uma melhoria. O mesmo ocorreu com o atributo “modificável”, com média 4,33 e o menor desvio padrão da dimensão (0,52), indicando um maior consenso quanto ao benefício do processo em relação a este item.

### 6.1.3. Consolidação dos resultados

Ao consolidar os resultados dos dois projetos podemos identificar os pontos fortes e fracos do processo preliminar de engenharia de requisitos para ambientes de DDS em estudo. De forma geral na avaliação, todas as questões tiveram média acima de três indicando que, no

pior caso, não houve diferença com uso do processo e, na maioria das questões, houve melhoria com o uso do processo preliminar.

Diversas melhorias foram apontadas com o uso do processo preliminar. Os pontos onde a média de respostas foi mais alta (acima de 4,3) estão apresentados na Tabela 11.

Questão	Média	Desvio Padrão
<i>Dimensão 3 - Avaliação não comparativa do processo</i>		
2 - Os requisitos escritos utilizando o novo processo são <b>simples</b> de serem entendidos pelos usuários.	4,40	0,55
<i>Dimensão 4 - Avaliação comparativa do processo</i>		
1 - As múltiplas interações contribuem para um melhor <b>entendimento das necessidades</b> envolvidas no desenvolvimento do sistema.	5,00	0,00
2 - A utilização do novo processo promove uma melhor comunicação, fornecendo às equipes uma <b>linguagem comum</b> para comunicação (com relação à estrutura de frase).	4,80	0,45
6 - A especificação de requisitos resultante do novo processo é mais detalhada.	4,80	0,45
8 - Com a utilização do novo processo a confiança no entendimento dos requisitos entre as equipes aumenta.	4,40	0,55
10 - Com a utilização do novo processo os requisitos tornaram-se mais <b>claros</b> .	4,40	0,89
13 - O novo processo traz benefícios para as atividades de <b>teste</b> .	4,40	0,55
14 - O novo processo promove melhorias na <b>documentação</b> do software desenvolvido.	4,40	0,55
15 - O processo de engenharia de requisitos proposto representa um benefício para as seguintes etapas do processo de desenvolvimento:		
- <i>Planning</i>	4,80	0,45
- <i>Developing</i>	4,40	0,55
<i>Dimensão 5 - Avaliação da qualidade do SRS</i>		
1 - Com a utilização do novo processo o SRS torna-se mais <b>correto</b> .	4,31	0,75
6 - Com a utilização do novo processo o SRS torna-se mais <b>verificável</b> .	4,31	0,63

**Tabela 11 - EC1 - Pontos fortes do processo preliminar**

Os pontos de destaque do processo preliminar estão relacionados à comunicação, nível de detalhe, confiança, clareza e benefícios para demais fases e atividades do desenvolvimento de software. Ao obter requisitos de simples compreensão pelos usuários, a atividade de validação é simplificada. O *feedback* torna-se mais eficiente, permitindo uma melhor evolução dos artefatos de requisitos. Em conjunto, a confiança no entendimento dos requisitos entre as equipes é ampliado, permitindo um melhor relacionamento entre os envolvidos das diversas localidades. Este resultado reflete-se nas respostas apontando que a especificação tornou-se mais correta com a utilização do processo preliminar.

A estruturação dos requisitos em um formato padrão (neste caso, em relação à estrutura de frase) permitiu uma melhor comunicação entre as equipes, pois estas passaram a ter uma forma comum de expressar suas necessidades e objetivos. A melhora na comunicação

promovida pelas diversas interações entre as equipes foi consenso entre os entrevistados. Com isto, os requisitos se tornaram mais claros e a especificação mais rica em detalhes.

Com a maior clareza e profundidade dos requisitos, as atividades de documentação e teste, bem como as fases de *planning* e *developing* são beneficiadas. O resultado em relação ao teste foi corroborado pelos respondentes que apontaram que a especificação tornou-se mais verificável com o uso do processo preliminar.

Através do estudo de caso foram também identificados pontos de melhoria no processo preliminar. A síntese dos pontos avaliados com médias mais baixas (abaixo de 3,60) na pesquisa está na Tabela 12.

Questão	Média	Desvio Padrão
<i>Dimensão 3 - Avaliação não comparativa do processo</i>		
3 - A escrita de requisitos com o novo processo permite a captura de todos os tipos de informação necessárias à especificação.	3,40	1,34
<i>Dimensão 4 - Avaliação comparativa do processo</i>		
5 - O novo processo de engenharia de requisitos simplifica o entendimento de características específicas do <b>contexto</b> onde o software será utilizado.	3,20	0,84
9 - A especificação dos requisitos tornou-se <b>mais rápida</b> com a utilização do novo processo.	3,00	0,71
11 - O novo processo simplifica a descoberta de requisitos ocultos.	3,40	1,14
12 - Com a utilização do processo é possível se realizar uma melhor <b>estimativa</b> do esforço necessário para os próximos estágios de desenvolvimento.	3,60	1,14
15 - O processo de engenharia de requisitos proposto representa um benefício para as seguintes etapas do processo de desenvolvimento: - <i>Stabilizing</i>	3,60	0,89
<i>Dimensão 5 - Avaliação da qualidade do SRS</i>		
4 - Com a utilização do novo processo o SRS torna-se mais <b>consistente</b> .	3,58	1,31
5 - Com a utilização do novo processo o SRS é mais facilmente <b>priorizado</b> .	3,46	1,13

**Tabela 12 - EC1 - Pontos fracos do processo preliminar**

Ao avaliar se o processo preliminar permitiu capturar todos os tipos de informação necessários à especificação, a média aponta um provável ponto de melhoria. Entretanto, todos os respondentes concordaram com esta afirmação, exceto o gerente de projetos do projeto 1. Ao ser consultado a respeito, ele ressaltou que não existe forma única de representar requisitos, formas complementares de representação de requisitos podem ser utilizadas para simplificar o entendimento e tornar a especificação mais completa.

O entendimento de características de contexto também apresentou uma avaliação média baixa no estudo. Embora fosse esperado que com a melhora na comunicação fosse facilitada a descoberta de características de contexto, este resultado não se refletiu na

pesquisa. Uma possível forma de melhorar esta característica é apresentada por Mahemoff [MAH98]. Assim, a descoberta de requisitos ocultos poderia ser melhorada, principalmente quando relacionados à questões específicas de uma localidade.

O tempo despendido na especificação de requisitos obteve a avaliação mais baixa na pesquisa. Como esta foi a primeira vez que as equipes utilizaram o processo preliminar, é esperado que o tempo de treinamento e aprendizado exerça impacto. Entretanto, ainda são necessários mais dados para poder avaliar com maior precisão o efeito do uso do processo no esforço total da equipe e no tempo de desenvolvimento.

Contribuições para estimativa eram esperadas, uma vez que foi apontado que os requisitos estavam mais claros, corretos e detalhados com o processo preliminar, entretanto isto não ocorreu. Como o processo de estimativa da organização é baseado em dados históricos, talvez fosse necessário um maior número de projetos utilizando o processo preliminar para adequar a estimativa.

A contribuição do processo preliminar para fase de *stabilizing* do MSF também apresentou uma média baixa quando comparado aos demais resultados da pesquisa. A influência de uma melhoria nos requisitos nesta fase do MSF pode não ser clara, por isso recebendo uma menor avaliação em relação às demais fases.

Com relação às características de um bom SRS, as avaliações mais baixas foram com relação à consistência e priorização. A priorização não foi tratada como foco no processo preliminar, sendo esperado este resultado. Com relação à consistência a média foi baixa, entretanto teve um alto desvio padrão, apontado para uma falta de consenso entre os respondentes. Um possível motivo para isso foi o uso dos requisitos em linguagem natural na especificação nos dois projetos. Dessa forma, a identificação de inconsistências é reduzida quando comparada a representações formais de requisitos.

#### **6.1.4. Lições aprendidas**

Com base nos resultados consolidados e nas observações realizadas durante o estudo, foram identificadas as lições aprendidas do estudo de caso. Estas lições servirão como base para melhorias no processo preliminar de engenharia de requisitos para ambientes de DDS. Na seqüência, são apresentadas as lições aprendidas do estudo de caso 1:

**Lição 1: As interações entre as equipes visando evoluir os artefatos de requisitos permitem um melhor entendimento das necessidades envolvidas no desenvolvimento do software em questão.**

As interações entre as equipes de desenvolvimento e grupo de usuários e clientes com o objetivo de evoluir os artefatos promovem uma melhor compreensão dos motivos que guiaram o processo de desenvolvimento. Sem estas interações, o projeto e a codificação de software podem ser iniciados com a equipe de desenvolvimento desalinhada em relação aos objetivos que devem ser atingidos.

Dessa forma, evita-se o modelo de desenvolvimento apontado por Al-Rawas [ALR96] como “sobre o muro” onde, em cada estágio do projeto, a especificação é lançada sobre o muro para a próxima equipe, que está aguardando para conduzir a próxima etapa do processo.

**Lição 2: O uso de estruturas de frase para requisitos em linguagem natural contribui para melhorar a comunicação e o entendimento entre as equipes.**

Na avaliação dos resultados percebeu-se que um ponto importante para a melhoria na comunicação entre as equipes, bem como na clareza e entendimento dos requisitos foi o uso de estruturas de frases para especificação de requisitos em linguagem natural. Como havia diferença de idioma nativo entre as equipes, a utilização de estruturas simples de frases permitiu um melhor entendimento dos requisitos escritos. A comunicação também foi influenciada pela estrutura, uma vez que as equipes passaram a utilizá-la quando interagindo. A clareza obtida com a estruturação das frases promove uma melhor comunicação.

**Lição 3: São necessárias melhorias no processo visando entender questões relacionadas ao contexto dos grupos envolvidos.**

A captura de informações referentes ao contexto não obteve benefícios claros com o processo preliminar. Embora as interações entre as equipes visando evoluir os artefatos de requisitos pudessem auxiliar no entendimento do contexto, não existem atividades específicas com este objetivo no processo preliminar. Este é um ponto de melhoria para a proposta de um modelo de processo de engenharia de requisitos para ambientes de desenvolvimento distribuído de software. Em especial, o estudo de caso evidenciou a necessidade da captura de informações de contexto dos usuários e clientes, visando melhor compreender o ambiente onde o software em desenvolvimento será inserido.

**Lição 4: O processo preliminar necessita de novas atividades de forma a reduzir as inconsistências e melhorar a priorização dos requisitos.**

No estudo de caso, as principais características de qualidade da especificação de requisitos apontadas como passíveis de melhoria foram a consistência e a priorização dos requisitos. Nesse sentido são necessárias novas atividades no processo de engenharia de requisitos de forma a reduzir inconsistências e melhorar a priorização.

**Lição 5: Melhorias no processo preliminar devem ser realizadas de forma a permitir a captura de uma maior diversidade de informações, bem como a descoberta de informações ocultas.**

Outro ponto de melhoria no processo é o estabelecimento de formas que permitam a captura de uma maior variedade de informações, bem como a descoberta de informações ocultas. O uso da linguagem natural para captura de requisitos foi insuficiente em termos de representação. É necessário ampliar a forma de representação de acordo com o tipo de projeto. Nesse sentido, são necessárias novas atividades ao processo de engenharia de requisitos em DDS.

## **6.2. ESTUDO DE CASO 2**

O estudo de caso 2 tinha como objetivo identificar os problemas na especificação de requisitos ampliados por fatores relacionados à engenharia de requisitos em ambientes de desenvolvimento distribuídos de software na especificação de requisitos. Visando atingir este objetivo, foram entrevistados engenheiros de requisitos com experiência em desenvolvimento distribuído de software. A entrevista foi conduzida com base em um instrumento de pesquisa que permitia relacionar os principais fatores do desenvolvimento distribuído de software com problemas na especificação de requisitos, em cenários previamente estabelecidos.

Uma descrição do instrumento de coleta de dados é apresentada na seqüência. Em seguida, é descrita a análise dos dados obtidos. Ao final, são apresentadas as lições aprendidas.

### **6.2.1. Instrumento de Coleta de Dados**

O instrumento de coleta de dados utilizado no estudo de caso 1 era uma entrevista composta de três dimensões, visando capturar a percepção dos respondentes quanto aos problemas na especificação de requisitos ampliados por fatores do desenvolvimento distribuído de software (veja Apêndice 2 - Protocolo Para Estudo De Caso 2). A primeira dimensão do questionário, relativa aos dados demográficos, tinha como objetivo identificar os respondentes, quanto à formação acadêmica e profissional.

Na segunda dimensão foram apresentados dois cenários do desenvolvimento distribuído de software. O primeiro cenário apresentava um grupo co-localizado de engenheiros de requisitos distantes globalmente de um grupo, também co-localizado, de usuários e clientes. Neste cenário, foi solicitado aos entrevistados que estabelecessem relações em uma matriz composta de problemas na especificação de requisitos (colunas) e fatores do desenvolvimento distribuído de software (linhas), identificando quais problemas eram

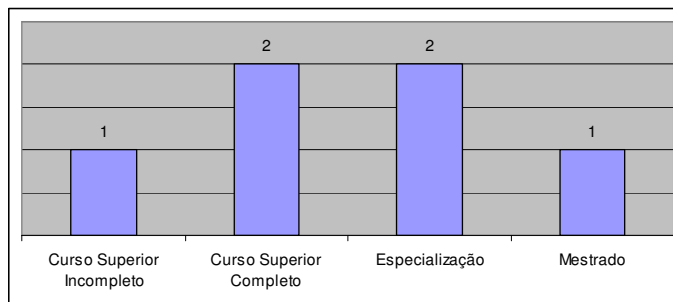
ampliados por cada fator. O segundo cenário apresentava uma distribuição onde um grupo co-localizado de engenheiros de requisitos está distante de um grupo de usuários e clientes disperso globalmente. Neste cenário foi questionado aos entrevistados se os problemas na especificação causados pelos fatores apontados no cenário 1 eram ampliados quando no cenário 2. Em seguida, foi questionado aos respondentes apontar se existiam novas relações entre fatores do DDS e problemas na especificação no cenário 2 e, se existirem, quais são elas.

Na terceira dimensão da entrevista eram apresentados dois cenários de desenvolvimento distribuído de software. O primeiro cenário apresentava um grupo co-localizado de engenheiros de requisitos distantes globalmente de uma equipe de desenvolvimento co-localizada. Neste cenário, era solicitado ao entrevistado que estabelecesse relações em uma matriz composta de problemas na especificação de requisitos (colunas) e fatores do desenvolvimento distribuído de software (linhas), identificando quais problemas eram ampliados por cada fator. O quarto cenário apresentava uma distribuição onde um grupo co-localizado de engenheiros de requisitos está distante de uma equipe de desenvolvimento dispersa globalmente. Neste cenário era questionado aos entrevistados se os problemas na especificação causados pelos fatores apontados no cenário 3 eram ampliados quando no cenário 4. Em seguida, era questionado aos respondentes apontar se existiam novas relações entre fatores do DDS e problemas na especificação no cenário 4 e, se existissem, quais seriam elas.

### **6.2.2. Análise de resultados**

No estudo de caso 2 foram entrevistados seis engenheiros de requisitos com experiência em projetos de desenvolvimento global. Destes cinco possuíam curso ensino superior completo, sendo que um deles já havia concluído mestrado e dois possuíam curso de especialização, conforme apresentado na Figura 20. Quatro dos entrevistados possuíam mais de cinco anos de experiência profissional. Todos os respondentes trabalhavam na unidade brasileira de uma organização que realiza desenvolvimento global de software.





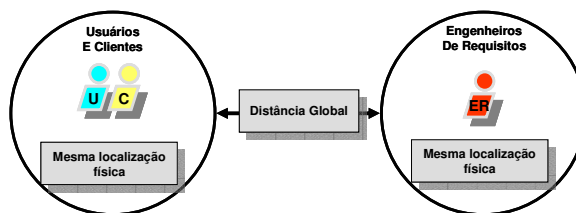
**Figura 20 - EC2 - Formação dos respondentes**

As respostas obtidas para cada cenário foram analisadas utilizando o módulo estatístico de uma planilha eletrônica. Os resultados foram comparados com a base teórica existente em engenharia de requisitos para ambientes de DDS. Na seqüência é apresentada a análise dos resultados do estudo de caso 2 divididos segundo os cenários que guiaram a entrevista. As tabelas completas com os dados consolidados deste estudo de caso estão no Apêndice 3 - Dados Consolidados Do Estudo De Caso 2.

#### **A) Cenário 1**

O cenário 1 apresenta um ambiente de desenvolvimento de software global, onde os engenheiros de requisitos estão distantes dos usuários e clientes (Figura 21). Entretanto, usuários e clientes estão na mesma localidade, bem como engenheiros de requisitos estão co-localizados. Ou seja:

- Engenheiros de requisitos em uma mesma localização física;
- Usuários e clientes em uma mesma localização física, distante globalmente dos engenheiros de requisitos.



**Figura 21 - EC2 - Equipe de engenheiros de requisitos distribuída em relação a usuários e clientes**

Alguns exemplos deste cenário são:

- Engenheiros de requisitos em São Paulo e usuários e clientes em um prédio em Londres.
- Engenheiros de requisitos em Bangalore e usuários e clientes em Tóquio.

Neste cenário foi solicitado aos respondentes que apontassem os principais problemas na especificação de requisitos causados pelo fatores relacionados ao desenvolvimento

distribuído de software. Na seqüência são apresentados os resultados consolidados para este cenário, agrupados por tipo de problema na especificação. A Tabela 13 apresenta a síntese das respostas para afirmação ambígua no cenário 1. Os resultados são comentados na seqüência, após a Tabela.

<b>Fator relacionado ao Desenvolvimento Distribuído de Software</b>	<b>Informação Ambígua</b>	<b>% de respondentes</b>
<b>Idioma nativo dos <i>stakeholders</i></b>	6	100%
<b>Comunicação entre as localidades via teleconferência</b>	5	83%
<b>Diferenças de contexto entre as localidades</b>	1	17%

**Tabela 13 - EC2 - Síntese das respostas para Informação Ambígua no cenário 1**

Os seis respondentes relacionaram a diferença de idioma dos *stakeholders* como uma fonte de ambigüidades para a especificação de requisitos quando o engenheiro de requisitos está distante de usuários e clientes. Durante o processo de engenharia de requisitos, as interações realizadas entre as equipes, bem como a escrita e leitura do documento de especificação em um idioma diferente do nativo tende a ampliar a possibilidade de mal-entendidos e ambigüidades.

Alinhando-se com o encontrado na literatura, onde Al-Rawas [ALR96] aponta o risco de interpretações incorretas devido a opiniões pré-concebidas, a comunicação por teleconferência foi indicada pela maioria dos respondentes como fonte de ambigüidade nos requisitos, quando utilizada como meio para interação entre engenheiros de requisitos e usuários e clientes. Layzell [LAY00] aponta para problema similar quando a comunicação em um projeto ocorre por e-mail, uma vez que os participantes tendem a ser menos comprometidos. Entretanto, apenas metade dos respondentes apontou a comunicação por e-mail como fator ampliador de ambigüidades na especificação. Este resultado foi justificado por alguns dos respondentes devido à vantagem do e-mail sobre a teleconferência ao permitir um histórico de mensagens, reduzindo, desta forma, a possibilidade de ambigüidades além das comuns ao uso da linguagem.

Embora a relação entre diferença de contexto e ambigüidades fosse esperada, apenas um dos entrevistados estabeleceu esta ligação. Conforme apresentado em [MAH98], as diferenças de contexto e cultura podem inserir ambigüidades no entendimento da especificação de requisitos. Uma hipótese para este desvio é obtida pela relação entre diferença de contexto e informação faltante estabelecida pela maioria dos respondentes. Ao realizar esta relação, os respondentes possivelmente consideraram que as informações relacionadas a contexto que poderiam originar ambigüidades ocorreriam devido à falta de um maior detalhamento que impedisse múltiplas interpretações.

A síntese dos resultados para informação faltante é apresentada no Tabela 14 e detalhada a seguir.

Fator relacionado ao Desenvolvimento Distribuído de Software	Informação Faltante	% de respondentes
Valores pessoais dos <i>stakeholders</i> (influenciada pela cultura)	5	83%
Atitudes e forma de agir (influenciada pela cultura)	5	83%
Gerenciamento de informações dispersas em múltiplas localidades	5	83%
Consciência das mudanças (no negócio, equipe, etc.) ou progressos nas localidades envolvidas ( <i>awareness</i> )	6	100%
Dificuldade de identificação dos <i>stakeholders</i>	5	83%
Diferenças de contexto entre as localidades	5	83%

**Tabela 14 - EC2 - Síntese das respostas para Informação Faltante no cenário 1**

O *awareness* foi considerado como um dos fatores causadores de falta de informações na especificação por todos os respondentes. A consciência do ambiente de negócio e suas alterações é importante para que não sejam deixadas de lado informações críticas para o desenvolvimento do software. Este resultado alinha-se com o estudo de Damian [DAM03].

Os valores pessoais dos *stakeholders*, bem como atitudes e forma de agir (ambos influenciados pela cultura) foram relacionados com a falta de informações por 5 dos 6 respondentes. Isto pode ser justificado pela resistência de alguns grupos de fornecer informações importantes para a engenharia de requisitos. Esta resistência tende a ser ampliada pelo desenvolvimento *offshore*, devido a questões como a rivalidade entre as equipes [LAY00] e o receio da perda de empregos para outros países [DRE04].

Também de acordo com a maioria dos respondentes (83%), informações dispersas em diversas localidades tendem a causar a falta de informações na especificação de requisitos. Este resultado alinha-se com os estudos de Zowghi [ZOW02]. O acesso a documentos e informações referentes ao negócio é necessário durante o processo de elicitação e especificação de requisitos. Quando em ambientes distribuídos o acesso a estes documentos é dificultado, podendo ocasionar a falta de requisitos importantes para o sistema.

Conforme esperado, de acordo com outros estudos [SHA99][ALR96], a dificuldade de identificação dos *stakeholders* também foi apontada como fonte para a falta de informações na especificação (5 dos 6 respondentes). Uma vez que não sejam identificados corretamente os *stakeholders*, informações importantes podem ser deixadas de lado, gerando uma especificação incompleta.

Além destas, foi observada uma relação da diferença de contexto entre as localidades e a falta de informações (5 dos 6 respondentes). A diferença de contexto pode fazer com que os *stakeholders* deixem de fornecer informações importantes por considerarem óbvias. Este resultado se alinha com o estudo de Damian [DAM02].

Os resultados consolidados para informações inconsistentes no cenário 1 são apresentadas na seqüência (Tabela 15).

Fator relacionado ao Desenvolvimento Distribuído de Software	Informação Inconsistente	% de respondentes
Idioma nativo dos <i>stakeholders</i>	5	83%
Dificuldade de identificação dos <i>stakeholders</i>	5	83%

**Tabela 15 - EC2 - Síntese das respostas para Informações Inconsistentes no cenário 1**

A dificuldade de identificação dos *stakeholders* foi apontada por 5 dos 6 respondentes como fonte para inconsistência das informações na especificação. Se identificados incorretamente, os *stakeholders* selecionados podem não possuir domínio sobre o universo de informações em questão e fornecer informações contraditórias ao engenheiro de requisitos, causando inconsistências na especificação.

O idioma nativo dos *stakeholders* foi relacionado com a presença de informações inconsistentes nos requisitos por 83% dos respondentes. Esta informação não foi encontrada em outros estudos da área. Uma possível explicação para este fato é o incorreto entendimento do termo inconsistência, muitas vezes tratado como equivalente a ambigüidade. Outra hipótese provável é o surgimento de inconsistências devido à incorreta interpretação ou mau uso da linguagem no documento de requisitos.

O Tabela 16 apresenta a síntese dos resultados para informações incorretas no cenário 1. Estes dados são detalhados na seqüência.

Fator relacionado ao Desenvolvimento Distribuído de Software	Informação Incorreta	% de respondentes
Consciência das mudanças (no negócio, equipe, etc.) ou progressos nas localidades envolvidas ( <i>Awareness</i> )	6	100%
Diferenças de contexto entre as localidades	5	83%

**Tabela 16 - EC2 - Síntese das respostas para Informação Incorreta no cenário 1**

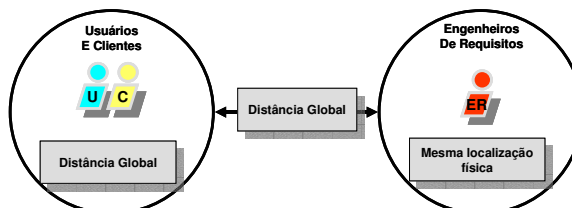
O principal fator apontado como ampliador das informações incorretas na especificação foi o *awareness* (todos os respondentes). Este fator tem efeito principalmente sobre o gerenciamento de requisitos. Se uma modificação no negócio não for refletida na especificação de requisitos, serão utilizadas informações desatualizadas, ou incorretas, no desenvolvimento. Este resultado era esperado, estando de acordo com o estudo [DAM03].

Além disso, foi observada uma relação entre a diferença de contexto entre as localidades e informações incorretas (5 dos 6 respondentes). A diferença de contexto pode fazer com que os *stakeholders* deixem de fornecer informações importantes por considerarem óbvias [DAM02][MAH98]. Em conjunto, o responsável pela especificação pode assumir determinadas questões com base em seu contexto (como legislação, por exemplo) e não consultar usuários e clientes, definindo requisitos incorretos.

### B) Cenário 2

O cenário 2 aborda a distribuição de um grupo de engenheiros de requisitos co-localizados em relação a usuários e clientes distribuídos (Figura 22). Ou seja:

- Engenheiros de requisitos em uma mesma localização física;
- Usuários e clientes distribuídos globalmente, distantes dos engenheiros de requisitos.



**Figura 22 - EC2 - Usuários e clientes distribuídos entre si e em relação aos engenheiros de requisitos**

Alguns exemplos deste cenário são:

- Engenheiros de requisitos em Porto Alegre, usuários no Canadá, Inglaterra e Japão e clientes nos Estados Unidos.
- Engenheiros de requisitos em Bangalore, usuários dispersos na Ásia e Oceania e clientes na Austrália.

No cenário 2 foi questionado aos respondentes se os problemas na especificação ampliados pelos fatores relacionados ao DDS eram aprofundados em relação ao cenário 1. Em seguida, foi solicitado aos respondentes para identificar se existem novos relacionamentos entre problemas na especificação e fatores relacionados ao DDS emergem no cenário 2 e, se existirem, quais são.

Na percepção de todos os respondentes os problemas nos requisitos causados pelos fatores selecionados no cenário 1 são aprofundados quando existem diversos grupos de usuários e clientes dispersos (cenário 2). Assim, de acordo com a experiência dos entrevistados, a tendência a erros na especificação é ampliada quando existem múltiplos grupos dispersos de usuários e clientes.

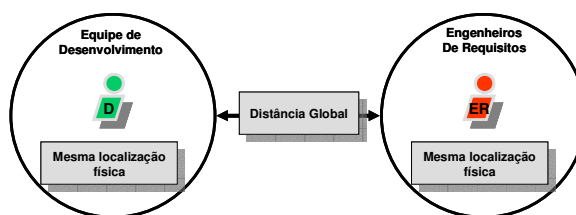
Da mesma forma, todos os respondentes concordaram que emergem novos problemas na especificação de requisitos ampliados pelos fatores relacionados ao DDS, além dos apontados no cenário 1. Entretanto, analisando os dados consolidados dos novos problemas apontados pelos entrevistados, não foi possível obter nenhuma tendência significativa. Uma possível explicação para este fato é de que nem todos os respondentes tiveram experiência com usuários e clientes em múltiplas localidades. Outra possibilidade é de que os problemas

na especificação de requisitos ampliados pelo DDS sejam os mesmos nos cenários 1 e 2, sendo apenas ampliados neste último.

### C) Cenário 3

O cenário 3 apresenta um ambiente de desenvolvimento de software global, onde os engenheiros de requisitos estão distantes da equipe de desenvolvimento. Entretanto, toda a equipe de desenvolvimento está na mesma localidade, bem como engenheiros de requisitos estão co-localizados (Figura 23). Ou seja:

- Engenheiros de requisitos em uma mesma localização física;
- A equipe de desenvolvimento está em uma mesma localização física, distante globalmente dos engenheiros de requisitos.



**Figura 23 - EC2 - Engenheiros de requisitos distribuídos globalmente em relação à equipe de desenvolvimento**

Alguns exemplos deste cenário são:

- Engenheiros de requisitos em Porto Alegre e equipe de desenvolvimento em Bangalore.
- Engenheiros de requisitos em Londres e equipe de desenvolvimento em Porto Alegre.

Neste cenário foi solicitado aos respondentes que apontassem os principais problemas na especificação de requisitos causados pelos fatores relacionados ao desenvolvimento distribuído de software. Na seqüência são apresentados os resultados consolidados para este cenário, agrupados por tipo de problema na especificação. O Tabela 17 apresenta a síntese das respostas para afirmação ambígua no cenário 3. Os resultados são comentados na seqüência, após o quadro.

Fator relacionado ao Desenvolvimento Distribuído de Software	Informação Ambígua	% de respondentes
Idioma nativo dos <i>stakeholders</i>	5	83%
Valores pessoais dos <i>stakeholders</i> (influenciada pela cultura)	5	83%
Atitudes e forma de agir (influenciada pela cultura)	5	83%

**Tabela 17 - EC2 - Síntese das respostas para Informação Ambígua no cenário 3**

A maioria dos respondentes apontou o idioma nativo dos *stakeholders* como uma fonte para ambigüidades. Quando a especificação de requisitos é escrita por um grupo distante da

equipe de desenvolvimento, a diferença de idioma nativo tende a ampliar as ambigüidades. Este resultado alinha-se com o estudo de Damian [DAM02] onde a diferença de idioma é apontada como uma fonte de ambigüidades.

Os valores pessoais dos *stakeholders*, bem como atitudes e forma de agir dos envolvidos no processo de desenvolvimento foram considerados origem de ambigüidades para 5 dos 6 respondentes. Estes fatores foram apontados, de acordo com um dos respondentes, porque muitas vezes por inibição, ou receio de demonstrar o não conhecimento de uma determinada característica do sistema, os envolvidos deixam de solicitar clarificações. Dessa forma, especificações são realizadas com ambigüidade propositadamente, onde seus responsáveis deixam para buscar maiores informações no decorrer do processo de desenvolvimento. O mesmo ocorre com os responsáveis pelo projeto da aplicação e os desenvolvedores. O DDS amplia este problema, pois existe receio de que a confiança seja abalada se forem realizados muitos questionamentos. Este resultado não foi encontrado em outros estudos da literatura.

A síntese dos resultados para informação faltante no cenário 13 é apresentada no Tabela 18 e detalhada na seqüência.

<b>Fator relacionado ao Desenvolvimento Distribuído de Software</b>	<b>Informação Faltante</b>	<b>% de respondentes</b>
<b>Idioma nativo dos <i>stakeholders</i></b>	5	83%
<b>Valores pessoais dos <i>stakeholders</i> (influenciada pela cultura)</b>	5	83%
<b>Comunicação entre as localidades via e-mail</b>	5	83%
<b>Dificuldade de identificação dos <i>stakeholders</i></b>	5	83%
<b>Diferenças de contexto entre as localidades</b>	6	100%

**Tabela 18 - EC2 - Síntese das respostas para Informação Faltante no cenário 3**

O fator mais fortemente relacionado pelos respondentes com a falta de informações foi a diferença de idioma entre os *stakeholders*. Com a diferença de idioma, a comunicação é prejudicada, fazendo com que requisitos importantes possam ser deixados de lado.

Os valores pessoais dos *stakeholders* também foram apontados como uma possível fonte de falta de informações. O motivo para isso é a provável resistência dos *stakeholders* a compartilharem informações, com receio da perda de empregos ou *status* para outra localidade. Esta dificuldade tem aparecido em diversas empresas que realizam desenvolvimento *offshore*, como apontado em [DRE04].

A dificuldade de identificação dos *stakeholders* também foi apontada como fonte de falta de informações. Neste cenário isto acontece principalmente pela dificuldade da equipe de desenvolvimento de identificar os responsáveis pela clarificação das informações na

especificação. A redução da comunicação informal é um fator envolvido com essa dificuldade, como apresentado em [DAM02] e [BRA03].

A diferença de contexto entre as localidades também foi apontada como causa para a falta de informações neste cenário. Se o responsável pela especificação dos requisitos não considera o contexto existente nas localidades onde o desenvolvimento será realizado e omite informações necessárias por considerá-las óbvias, poderão faltar informações importantes na especificação de requisitos.

O Tabela 19 apresenta a síntese das respostas para informações inconsistentes no cenário 3. Estas informações são detalhadas na seqüência.

Fator relacionado ao Desenvolvimento Distribuído de Software	Informação Inconsistente	% de respondentes
Idioma nativo dos <i>stakeholders</i>	5	83%
Consciência das mudanças (no negócio, equipe, etc.) ou progressos nas localidades envolvidas ( <i>Awareness</i> )	5	83%
Dificuldade de identificação dos <i>stakeholders</i>	5	83%

**Tabela 19 - EC2 - Síntese das respostas para Informações Inconsistentes no cenário 3**

No cenário de distribuição da equipe de engenheiros de requisitos em relação à equipe de desenvolvimento, as informações inconsistentes foram relacionadas principalmente à diferença de idioma, *awareness* e dificuldade de identificação dos *stakeholders*.

A relação entre a diferença de idioma nativo dos *stakeholders* e informações inconsistentes na especificação foi apontada por 83% dos respondentes. As possíveis justificativas para esta resposta são similares a do cenário 1. Os respondentes podem ter confundido os termos inconsistência e ambigüidade. Outra possibilidade é que a inconsistência seja decorrente de erros de interpretação devido à diferença de linguagem.

A consciência das mudanças ou progresso nas localidades envolvidas foi identificado por 83% dos respondentes como ampliador de inconsistências nos requisitos. Quando em ambientes distribuídos, alterações em uma das localidades, se não forem corretamente identificadas e alertadas para todas as localidades envolvidas, podem levar a inconsistências.

A dificuldade de identificação dos *stakeholders* foi apontada por 5 dos 6 respondentes como fator que amplia informações inconsistentes nos requisitos. Ao contatar os *stakeholders* em busca de informações ou clarificação de requisitos, se não forem identificados corretamente os responsáveis, podem ser obtidas informações conflitantes, originando inconsistências.

Os resultados consolidados para informação incorreta no cenário 3 são apresentadas no Tabela 20 e detalhadas a seguir.



Fator relacionado ao Desenvolvimento Distribuído de Software	Informação Incorreta	% de respondentes
Gerenciamento de informações dispersas em múltiplas localidades	5	83%
Consciência das mudanças (no negócio, equipe, etc.) ou progressos nas localidades envolvidas ( <i>Awareness</i> )	5	83%
Dificuldade de identificação dos <i>stakeholders</i>	6	100%

Tabela 20 - EC2 - Síntese das respostas para Informação Incorreta no cenário 3

Os principais fatores que foram apontados como causa para informações incorretas no cenário 3 foram o gerenciamento de informações dispersas em múltiplas localidades, *awareness* e a dificuldade de identificação dos *stakeholders*.

Informações dispersas em múltiplas localidades, em conjunto com a dificuldade de identificar mudanças nas informações (relacionada ao *awareness*) tendem a fazer com que informações incorretas façam parte da especificação. Se a equipe de desenvolvimento não é alertada de alteração nos requisitos, o processo de desenvolvimento ocorre sobre uma especificação desatualizada, com requisitos incorretos. Isto é ampliado no DDS, onde mecanismos informais de troca de informação, como conversas em encontros casuais nos corredores da empresa [DAM02][DAM03], são reduzidos.

A dificuldade de identificação dos *stakeholders* também foi apontada como causa para a presença de informações incorretas na especificação. Ao buscar clarificação sobre a especificação, se não forem contatadas as pessoas corretas, informações incorretas poderão ser utilizadas.

#### D) Cenário 4

O cenário 4 aborda a distribuição de um grupo de engenheiros de requisitos co-localizados em relação a equipes de desenvolvimento distribuídas (Figura 24). Ou seja:

- Engenheiros de requisitos em uma mesma localização física;
- Equipes de desenvolvimento distribuídas globalmente e distantes globalmente dos engenheiros de requisitos.

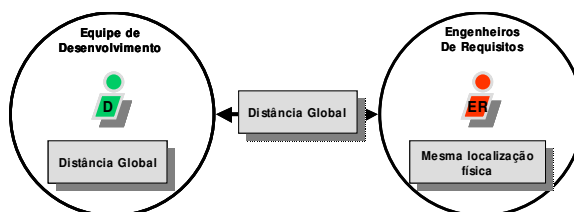


Figura 24 - EC2 - Equipe de desenvolvimento distribuída entre si e em relação ao engenheiro de requisitos

Alguns exemplos deste cenário são:

- Engenheiros de requisitos em Toronto, equipe de desenvolvimento em Toronto, Porto Alegre e Pequim.

- Engenheiros de requisitos em Porto Alegre, desenvolvimento (codificação) em Porto Alegre e Bangalore, testes em Lisboa.

No cenário 4 foi questionado aos respondentes se os problemas na especificação ampliados pelos fatores relacionados ao DDS eram aprofundados em relação ao cenário 3. Em seguida, foi solicitado aos respondentes para identificar se existem novos relacionamentos entre problemas na especificação e fatores relacionados ao DDS emergem no cenário 4 e, se existirem, quais são.

Todos os respondentes concordaram que, no cenário 4, todos os fatores relacionados ao desenvolvimento distribuído de software apontados no cenário 3 que ampliam a incidência de problemas nos requisitos são aprofundados. Dessa forma observamos que, na percepção dos respondentes, a existência de diversas equipes de desenvolvimento distribuídas entre si e em relação aos engenheiros de requisitos tende a ampliar os problemas na especificação apontados no cenário 3.

Na questão 2 todos os respondentes concordaram que existem outros fatores relacionados ao DDS que ampliam os problemas em requisitos, além dos apontados no cenário 3. Entretanto, não foram encontrados dados significativos dentre os novos fatores apontados para este cenário. A principal hipótese que emerge é que não existem novas relações entre fatores e problemas nos requisitos no cenário 4, somente uma ampliação dos apontados no cenário 3.

### **6.2.3. Consolidação dos resultados**

Ao consolidar os resultados dos cenários, podemos identificar tendências que ocorreram de forma geral no estudo, conforme identificado na seqüência.

A diferença de fuso-horário não apresentou relações significativas com nenhum dos problemas nos requisitos, embora fosse esperada sua relação com a falta de informações, por exemplo. Ao considerar equipes com grandes diferenças de fuso-horário, os horários de trabalho podem ser pouco ou nunca sincronizados. Com isso, espera-se que problemas na especificação surjam, como a falta de informações devido à dificuldade de comunicação síncrona. Entretanto, esta relação não foi encontrada.

O número de respondentes que consideraram que os fatores listados ampliavam o número de informações desnecessárias na especificação de requisitos foi, de maneira geral, baixo. De acordo com dois dos entrevistados, a dificuldade de se obter informações com o DDS é comumente grande, fazendo com que toda a informação trocada seja importante.

Dentre os fatores analisados, destacaram-se como causadores de problemas na especificação de requisitos a diferença de idioma, a dificuldade de identificação dos *stakeholders* e o *awareness*, como apresentado na Tabela 21. A diferença de idioma apresenta-se como um fator ampliador de problemas na especificação, em especial ambigüidade, inconsistência e falta de informações. A identificação incorreta dos *stakeholders* tende a ampliar a ocorrência de informações incorretas ou inconsistentes, bem como causar a falta de requisitos na especificação. Dificuldades com o *awareness*, como as localidades envolvidas não estarem a par de todas as modificações nos negócios ou requisitos do software em desenvolvimento, tendem a ampliar as informações incorretas na especificação, bem como reduzir a completude e gerar inconsistências.

<b>Principais fatores que ampliam problemas na especificação</b>
Diferença de Idioma
Dificuldade de identificação dos <i>stakeholders</i>
<i>Awareness</i>

**Tabela 21 - EC2 - Principais fatores que ampliam problemas na especificação**

Dentre os problemas na especificação ampliados pelos fatores do desenvolvimento distribuído de software destaca-se a falta de informações, com o maior número de respostas. Diversos fatores podem causar a falta de informações. Dificuldades de comunicação causadas pela diferença de idioma nativo dos *stakeholders*, pelo meio de comunicação, ou mesmo, pela diferença de fuso horário, tende a reduzir o volume de informações transmitido para equipes distantes, causando assim a falta de informações na especificação. Além disso, a dispersão de documentos, dificuldade de identificação dos *stakeholders*, entre outros, também contribuem para que informações importantes não sejam descobertas pelos engenheiros de requisitos, ou ainda, não estejam disponíveis para a equipe de desenvolvimento.

#### **6.2.4. LIÇÕES APRENDIDAS**

Com base nos resultados do estudo de caso, foram obtidas lições aprendidas em relação aos problemas na especificação causados pelos fatores do desenvolvimento distribuído de software. Estas lições são apresentadas na seqüência.

**Lição 1: Os problemas na especificação de requisitos são ampliados quando existe diferença de idioma nativo entre os envolvidos.**

A diferença de idioma é um dos principais fatores que ampliam os problemas na especificação de requisitos. A diferença de idioma apresenta-se como um ampliador de ambigüidades, inconsistências e falta de informações. As ambigüidades são ampliadas, por exemplo, pela falta de conhecimento do idioma utilizado na especificação, levando à má interpretação, ou mesmo à escrita de requisitos incorretos. As inconsistências podem ser

ampliadas, por exemplo, devido à incorreta compreensão das necessidades dos *stakeholders* durante a elicitação, fazendo com que apareçam conflitos entre os requisitos especificados. A falta de informações pode ocorrer pela dificuldade de expressão em uma língua diferente da nativa, por parte dos envolvidos, que deixam de informar ou questionar informações importantes para especificação.

**Lição 2: A identificação incorreta dos *stakeholders* tende a ampliar os problemas na especificação de requisitos quando em ambientes de DDS.**

A identificação incorreta dos *stakeholders* é um fator que amplia o volume de informações incorretas, de inconsistências e a falta de informações na especificação. Se não forem consultadas pessoas com conhecimento profundo do ambiente durante a elicitação de requisitos, informações incorretas podem passar a fazer parte da especificação, sendo descobertas tardiamente no processo de desenvolvimento. O mesmo pode causar inconsistências, se opiniões conflitantes fizerem parte da especificação, por exemplo. Além disso, se os todos os *stakeholders* significativos não forem entrevistados, pode ocorrer a falta de requisitos na especificação.

**Lição 3. Problemas na especificação são ampliados devidos à falta de consciência das modificações ou progressos nas localidades envolvidas.**

O *awareness* tende a aprofundar as informações incorretas na especificação, ampliar o volume de inconsistências e a falta de requisitos. Se mudanças no negócio durante a especificação, por exemplo, ocorrerem sem que a equipe de engenheiros de requisitos seja alertada, informações desatualizadas, e portanto incorretas, podem fazer parte da especificação. Isso também pode provocar inconsistências quando as modificações não forem alertadas completamente a todos os grupos. Da mesma forma, a falta de informações pode ocorrer se todas as equipes não estiverem conscientes do estado do negócio.

**Lição 4: O principal problema na especificação ampliado pelos fatores do DDS é a falta de informações.**

O principal problema na especificação ampliado por fatores relacionados ao desenvolvimento distribuído de software é a falta de informações. A falta de informações é causada principalmente pela diferença de idioma nativo dos *stakeholders*, pelo meio de comunicação utilizado, pela dispersão dos documentos e pela dificuldade de identificação dos *stakeholders*.

**Lição 5: Quanto maior o número de grupos de *stakeholders* envolvidos, mais ampliados tornam-se os problemas na especificação.**

Conforme apresentado nos cenário 2, os problemas na especificação identificados no cenário 1 são ampliados quando existem diversos grupos de usuários e clientes dispersos. Da mesma forma, no cenário 4, apresentou-se que, quando o número de equipes de desenvolvimento dispersas aumenta, os problemas na especificação tendem a ser ampliados.

## 7. MODELO DE PROCESSO PROPOSTO

Neste capítulo é apresentada a proposta de um modelo de processo de engenharia de requisitos para ambientes de desenvolvimento distribuído de software, com base na teoria existente no assunto e nos resultados dos estudos de caso (sintetizado nas lições aprendidas). O modelo de processo visa reduzir o impacto da dispersão das equipes na engenharia de requisitos.

Zowghi [ZOW02] apresenta a necessidade de definição de um processo de engenharia de requisitos para o desenvolvimento global de software. De acordo com essa abordagem, foi construída a proposta de um modelo de processo, identificando as principais atividades que devem ser realizadas para atenuar as dificuldades promovidas pelo desenvolvimento distribuído de software. De acordo com Sommerville [SOM96], modelos de processo são dependentes de interpretação para serem instanciados de acordo com conjuntos particulares de circunstâncias. As atividades e suas implementações não são definidas em detalhe nestes modelos. A quebra do processo em atividades de baixa granularidade é uma tarefa específica de cada projeto. Modelos de processo identificam os produtos do processo de software e as interações entre os desenvolvedores.

Dessa forma, o modelo proposto não se restringe a apresentar técnicas específicas durante o processo, definindo atividades de mais alto nível, com os artefatos de entrada e saída e sugestões de técnicas para sua realização. Dessa forma, de acordo com a estrutura organizacional, forma de distribuição das equipes, idiomas e culturas envolvidas, por exemplo, podem ser definidas diferentes técnicas para cada atividade do processo.

A proposta de modelo de processo de engenharia de requisitos para ambientes de DDS é composta de quatro fases (Figura 25): definições iniciais (Fase 1), mapeamento do contexto (Fase 2), criação da especificação de requisitos (Fase 3) e gerência de requisitos (Fase 4).

Estas fases foram identificadas com base nos resultados do estudo de caso 1, que evidenciou a necessidade de atividades anteriores (como planejar o processo e estabelecer o contexto) e posteriores ao processo preliminar (como a gerência de requisitos).

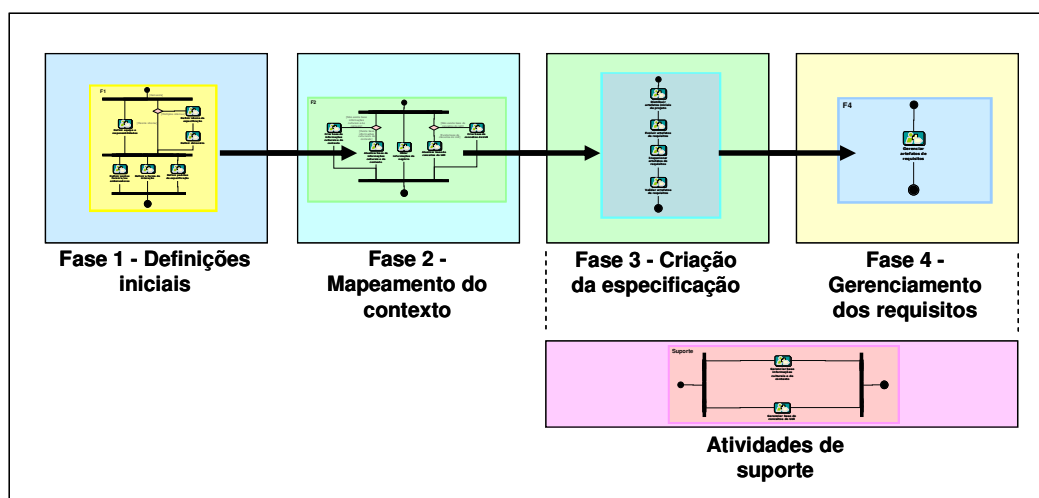
Na Fase 1 é estabelecida a infra-estrutura a ser utilizada durante o processo de engenharia de requisitos. Neste momento são definidos a linguagem de especificação, os embaixadores (pontos focais), o dicionário, os meios de comunicação a serem utilizados, os padrões de especificação, a equipe e as responsabilidades.

Durante a Fase 2 é realizado o mapeamento do contexto onde o software será inserido, incluindo a criação ou atualização da base de conceitos do universo de informações (UDI), criação ou atualização da base de informações culturais e obtenção das informações gerais de negócio.

Na Fase 3 é criada a especificação de requisitos. Neste momento os artefatos iniciais de requisitos são obtidos, evoluídos, inspecionados e validados. Em paralelo com esta fase, são mantidas atualizadas as bases de informações culturais e de conceitos do UDI.

Na Fase 4 é realizado o gerenciamento de requisitos, das bases de conceitos do UDI e da base de informações culturais, de forma a assegurar que todas as informações que sofrerem alterações durante o processo de desenvolvimento são alertadas para todas as equipes envolvidas.

Existem ainda atividades de suporte, para manutenção das bases de conceitos e de informações culturais ao longo das fases 3 e 4.



**Figura 25 - Modelo de processo de ER em DDS proposto**

Na seqüência são descritos os papéis e responsabilidades envolvidos no processo. Em seguida, são apresentadas em detalhe as fases do modelo de processo proposto, incluindo sugestões de técnicas a serem utilizadas para cada uma das atividades.

## 7.1. PAPÉIS E RESPONSABILIDADES

Os principais papéis envolvidos no modelo de processo são engenheiros de requisitos, usuários, clientes, gerentes de projeto e a equipe de desenvolvimento.

**O engenheiro de requisitos (ER)** é responsável pela elicitação, análise, negociação, documentação, validação e gerência dos requisitos. O engenheiro de requisitos próximo aos

usuários e clientes é chamado de **analista de negócios**. O engenheiro requisitos próximo à equipe de desenvolvimento é chamado de **analista de aplicações**.

**Usuários (U)** são os responsáveis pela utilização do produto gerado. **Clientes (C)** são as pessoas físicas ou jurídicas que solicitaram e contrataram o projeto. Tanto usuários quanto clientes são importantes fontes de informações para a especificação do software.

O **gerente de projetos (GP)** é o responsável pelo planejamento e condução do projeto de desenvolvimento de software contratado. O gerente de projetos pode fazer parte da equipe de desenvolvimento, do grupo de usuários e clientes, ou mesmo existir um gerente de projetos junto a cada um desses grupos.

A **equipe de desenvolvimento (D)** representa os envolvidos no desenvolvimento de um determinado projeto, utilizando como entrada os requisitos especificados pelo grupo de engenheiros de requisitos. Esta equipe pode envolver gerentes de projeto, codificadores, testadores, controladores de qualidade, equipe de suporte a ferramentas, entre outros.

Um exemplo da distribuição das equipes envolvidas é apresentado na Figura 26.

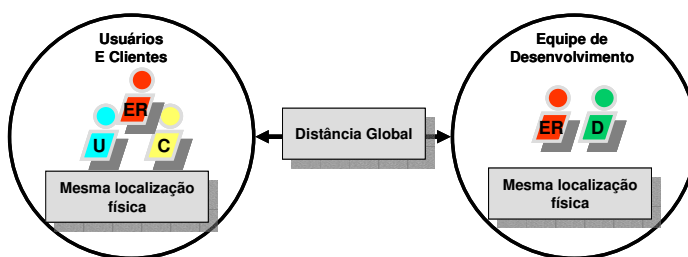


Figura 26 - Distribuição dos engenheiros de requisitos no exemplo de cenário

## 7.2. FASE 1 - DEFINIÇÕES INICIAIS

A primeira fase do modelo de processo visa estabelecer a infra-estrutura necessária à condução do processo de engenharia de requisitos em ambientes distribuídos. Neste ambiente é necessário escolher o idioma da especificação, o dicionário a ser utilizado, os meios de comunicação, os embaixadores e pontos focais, a equipe e responsabilidades, bem como os padrões de especificação. A Figura 27 apresenta o diagrama destas atividades.

Os **artefatos de entrada** para a fase de definições iniciais são: documentos iniciais do projeto (como charter ou documento de visão/escopo); o plano de projeto; e o plano de comunicação (pode ser uma seção do plano de projeto). Os **artefatos de saída** para esta fase são: o plano de projeto e o plano de comunicação atualizado.



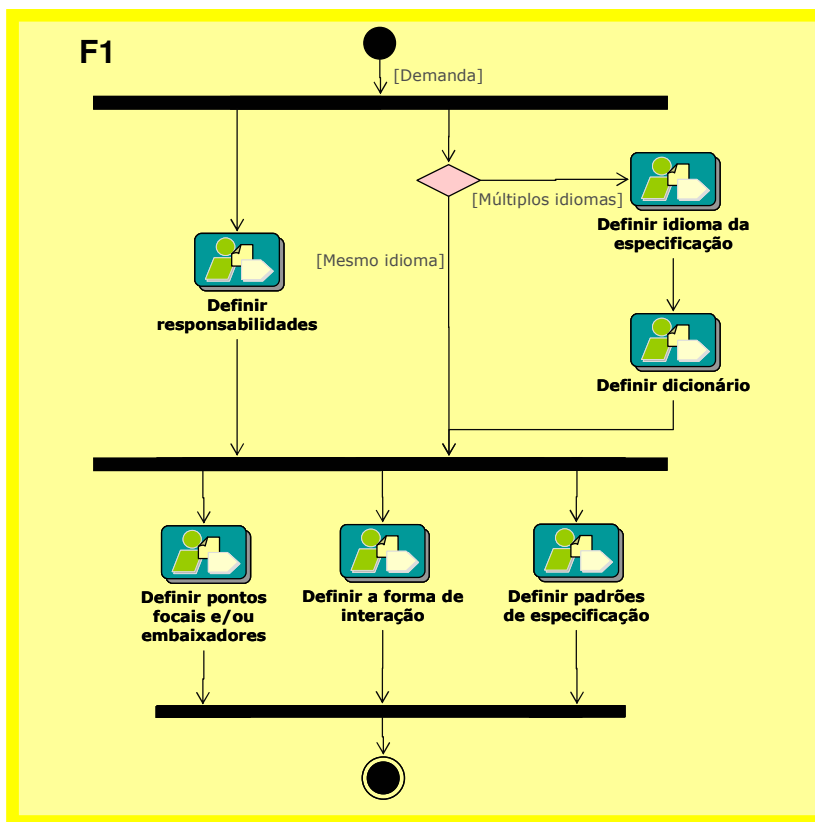


Figura 27 - Fase 1 - Definições iniciais

Estas atividades foram definidas com base na literatura e nos resultados dos estudos de caso. A Tabela 22 apresenta a origem de cada uma das atividades propostas. Em seguida, estas atividades são descritas em detalhe.

Atividade	Fonte
Definir idioma da especificação	EC2 (#1)
Definir dicionário	EC2 (#1)
Definir responsabilidades	EC2 (#3), teoria
Definir pontos focais/embaixadores	EC2 (#2,3), teoria
Definir formas de interação	EC1 (#1,4), teoria
Definir padrões de especificação	EC1 (#2,5)

# - Lição aprendida

Tabela 22 - Atividades relacionadas ao estabelecimento da infra-estrutura

### 7.2.1. Definir idioma da especificação

No início do processo de desenvolvimento de software, ao estabelecer a demanda, a primeira definição necessária refere-se ao idioma de especificação de requisitos. Quando o idioma nativo é o mesmo entre usuários, clientes, engenheiros de requisitos e equipe de desenvolvimento, esta etapa é desnecessária. Entretanto, quando existem múltiplos idiomas nativos entre as equipes, deve ser escolhido um para ser utilizado na condução do processo. A escolha do idioma a ser utilizado na especificação deve considerar diversos fatores, como:

- O uso da especificação após final do processo de desenvolvimento, em caso de manutenção do sistema, por exemplo;
- Problemas de interpretação que podem ser causados quando utilizando um idioma diferente do nativo da equipe de desenvolvimento;
- A necessidade de validação dos requisitos pelos usuários e clientes;

Por ser uma atividade que envolve interesses diretamente relacionadas ao negócio a definição do idioma da especificação deve envolver os clientes, o gerente de projetos e os engenheiros de requisitos. A Tabela 23 apresenta os artefatos de entrada e saída desta atividade.

<b>Artefatos de entrada:</b>	- Documentos iniciais do projeto (ex.: charter ou documento Visão/Esopo); - Plano de projeto; - Plano de comunicação; - Diretrizes da organização quanto ao idioma a ser utilizado em especificações de requisitos (se houver).
<b>Artefatos de saída:</b>	- Plano de projeto atualizado com informação de idioma da especificação;

**Tabela 23 - Artefatos de entrada e saída da atividade de definição do idioma da especificação**

### 7.2.2. Definir dicionário

Quando em ambientes com múltiplos idiomas nativos entre as equipes envolvidas, após a seleção do idioma a ser utilizado na especificação, é necessário definir-se um dicionário padrão a ser utilizado pelas equipes. Este dicionário serve como base na resolução de dúvidas de linguagem.

O uso de um dicionário reduzido como, por exemplo, as duas mil palavras mais utilizadas do idioma selecionado, permite que as equipes não nativas no idioma tenham maior facilidade de compreensão das informações. Neste caso, pode ser decidido que palavras que não constam no dicionário reduzido serão aceitas, desde que definidas explicitamente utilizando apenas palavras já existentes no dicionário.

A definição do dicionário a ser utilizado como base no processo de engenharia de requisitos deve ser realizada por consenso entre os engenheiros de requisitos envolvidos, que passam então a assegurar que os artefatos de requisitos gerados baseiam-se nas definições do dicionário escolhido. Em casos onde diversos projetos são desenvolvidos pelas mesmas organizações envolvidas, ou em *offshore insourcing*, por exemplo, um dicionário reduzido comum a todos os projetos desenvolvidos pode ser definido, evitando que termos tenham múltiplas interpretações dependendo do momento ou localidade onde a especificação foi desenvolvida.

A Tabela 24 apresenta os artefatos de entrada e saída desta atividade.

<b>Artefatos de entrada:</b>	- Documentos iniciais do projeto (ex.: charter ou documento Visão/Esopo); - Plano de projeto; - Plano de comunicação; - Diretrizes da organização quanto ao idioma a ser utilizado em especificações de requisitos (se houver).
<b>Artefatos de saída:</b>	- Plano de projeto atualizado com informação de dicionário;

**Tabela 24 - Artefatos de entrada e saída da atividade de definição de dicionário**

### 7.2.3. Definir responsabilidades

Nesta etapa do processo é definida a responsabilidade do analista de negócios e do analista de aplicações, com relação às atividades da engenharia de requisitos. A responsabilidade pela elicitação de requisitos, por exemplo, pode ser definida para o analista de negócios de acordo com suas habilidades e de sua disponibilidade. Em outros cenários, onde o analista de aplicação está preparado para realizar esta atividade, a elicitação pode ser conduzida por ele, com o analista de negócios agindo como facilitador junto aos usuários e clientes.

Da mesma forma, a responsabilidade pela especificação de requisitos pode ser compartilhada, com o analista de negócios sendo responsável pela especificação de requisitos de alto nível, e o analista de aplicação responsável por completar a especificação. Este tipo de decisão deve ser tomada pelo gerente de projetos, em conjunto com o cliente e os engenheiros de requisitos. A Tabela 25 apresenta os artefatos de entrada e saída da atividade de definição de responsabilidades.

<b>Artefatos de entrada:</b>	- Documentos iniciais do projeto (ex.: charter ou documento Visão/Esopo); - Plano de projeto;
<b>Artefatos de saída:</b>	- Plano de projeto atualizado com informação de responsabilidades.

**Tabela 25 - Artefatos de entrada e saída da atividade de definição de responsabilidades.**

### 7.2.4. Definir pontos focais e/ou embaixadores

A identificação dos responsáveis por cada informação é dificultada pela distribuição das equipes. Assim, a definição de pontos focais em todas as localidades, responsáveis por cada equipe ou tipo de informação, por exemplo, tende a reduzir os problemas existentes na busca de informações entre equipes distantes. A lista de pontos focais deve ser compartilhada e de fácil acesso, para que todos os envolvidos no processo de engenharia de requisitos possam consultar, quando necessário, para solicitar esclarecimentos e informações adicionais, por exemplo. Em caso de mudanças no negócio que afete o produto em desenvolvimento, os pontos focais são responsáveis por alertar as equipes envolvidas, gerando requisições de mudanças nos requisitos (veja subseção 7.5.1 - Gerenciar artefatos de requisitos).

O envio de membros da equipe próxima ao negócio para junto da equipe de desenvolvimento (chamados de embaixadores [FOW04]) em também pode ser benéfico para

o processo de engenharia de requisitos em DDS. Como a comunicação face a face tende a ser mais efetiva, as equipes se aproximam e o contexto de negócio é mais facilmente transmitido para equipe de desenvolvimento. O inverso também é válido, o envio de membros da equipe de desenvolvimento para junto aos usuários e clientes auxilia na compreensão das necessidades de negócio, bem como permite que se estabeleça um melhor relacionamento entre as equipes.

Outra importante atividade dos embaixadores é servir como ponte para comunicação informal entre as localidades envolvidas. Em projetos de software existe um grande volume de comunicação informal e, embora grande parte não seja útil, em alguns casos importantes informações deixam de ser comunicadas pelos canais formais para as demais localidades. Nestes casos, os embaixadores são responsáveis pela transferência dessas informações.

A definição dos pontos focais e embaixadores é conduzida pelo gerente de projetos em conjunto com usuários, clientes e engenheiros de requisitos, buscando obter representantes de todos os grupos e áreas de negócio envolvidas. Os artefatos de entrada e saída relacionados a esta atividade são apresentados na Tabela 26.

<b>Artefatos de entrada:</b>	- Documentos iniciais do projeto (ex.: charter ou documento Visão/Escopo); - Plano de projeto; - Plano de comunicação.
<b>Artefatos de saída:</b>	- Plano de projeto atualizado com a definição das pessoas que serão embaixadores e/ou pontos focais de cada área.

**Tabela 26 - Artefatos de entrada e saída da atividade de definir pontos focais e/ou embaixadores**

### 7.2.5. Definir a forma de interação

A definição da forma de interação entre os *stakeholders* é importante para evitar mal entendidos e problemas de comunicação. A forma de interação é dependente de restrições tecnológicas (infra-estrutura) e relacionadas à dispersão (como a diferença de fuso horário).

Em conjunto com a definição dos pontos focais e embaixadores, devem ser consideradas as formas de comunicação disponíveis a cada um deles e como utilizá-las. O uso de listas de distribuição de e-mail ou fóruns, por exemplo, podem ser uma alternativa para ampliar a consciência das equipes sobre o andamento ou mudanças nas localidades envolvidas. Neste caso, todos os envolvidos receberiam mensagens de alerta quando informações de interesse fossem enviadas por um dos membros da equipe.

A forma de interação também varia quanto ao nível de contexto. Quando temos contato face-a-face, utilizamos diversos mecanismos para expressar a mensagem que desejamos. Expressões faciais, gestos, alteração no tom de voz, entre outros, auxiliam na comunicação.

Conforme apontado no estudo de caso 2, embora mensagens eletrônicas sejam um meio de baixo contexto, ele possui a vantagem de manter registro histórico das comunicações, permitindo consultas posteriores.

A definição da forma de interação a ser utilizada durante o processo de requisitos pode ser realizada junto com o plano de comunicação do projeto, ou através de consenso entre o gerente de projetos, usuários, clientes e engenheiros de requisitos envolvidos. A Tabela 27 apresenta os artefatos de entrada e saída desta atividade.

<b>Artefatos de entrada:</b>	- Documentos iniciais do projeto (ex.: charter ou documento Visão/Esopo); - Plano de projeto; - Plano de comunicação;
<b>Artefatos de saída:</b>	- Plano de comunicação atualizado com a definição das formas de interação a serem utilizadas para engenharia de requisitos.

**Tabela 27 - Artefatos de entrada e saída da atividade de definição da forma de interação.**

### 7.2.6. Definir padrões de especificação

O objetivo da atividade de definição de padrões de especificação é negociar entre os envolvidos a forma de documentação a ser utilizada no processo de engenharia de requisitos. Neste momento, são definidas as estruturas dos documentos bem como seu conteúdo previsto.

Dentre os padrões de especificação, podem ser negociados o uso de estruturas de frase, cenários, casos de uso, diagramas, entre outros. Para isso, é importante considerar o idioma que será utilizado para especificação, e as alternativas existentes para melhorar a clareza e simplificar a leitura neste idioma.

Em muitos casos, como no *offshore insourcing*, estes padrões podem ser organizacionais, sendo selecionados de um conjunto pré-definido pela organização.

Os padrões devem ser escolhidos por consenso entre os engenheiros de requisitos, gerentes de projetos e clientes, de forma que todos os grupos contribuam na definição de padrões que melhorem a clareza da especificação para os leitores.

Os artefatos de entrada e saída relacionados a atividade de definição dos padrões de especificação são apresentados na Tabela 28.

<b>Artefatos de entrada:</b>	- Documentos iniciais do projeto (ex.: charter ou documento Visão/Esopo); - Plano de projeto; - Documentos de padrões de especificação da organização (se houver).
<b>Artefatos de saída:</b>	- Plano de projeto atualizado com a definição de padrões de especificação a serem utilizados no projeto, com referências para obtenção de maior detalhamento.

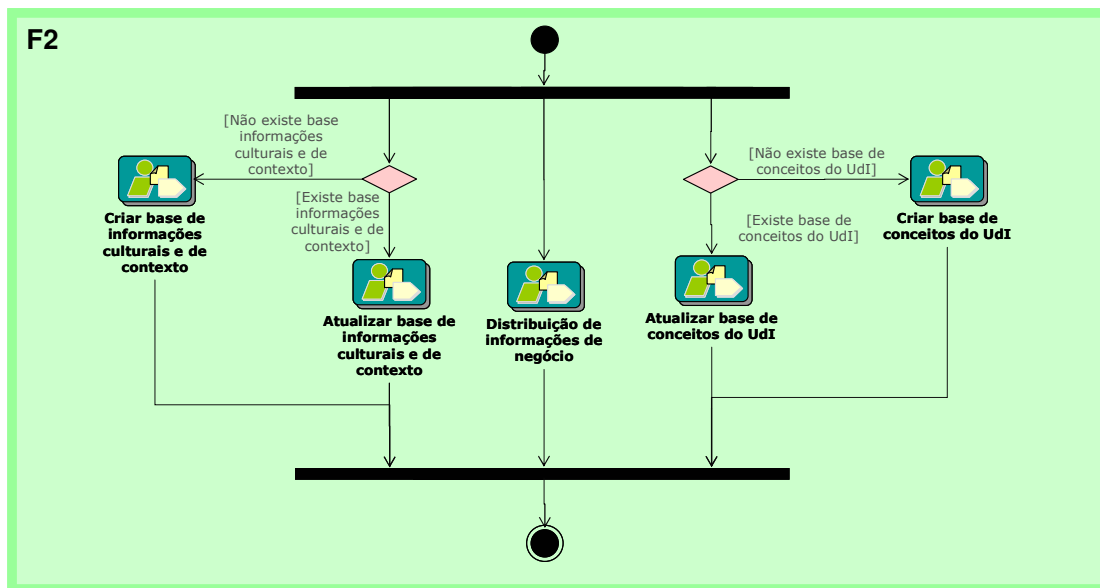
**Tabela 28 - Artefatos de entrada e saída da atividade de definição dos padrões de especificação.**

## 7.3. FASE 2 - MAPEAMENTO DO CONTEXTO

A segunda fase do modelo de processo visa mapear o contexto da localidade e do negócio onde o software em desenvolvimento será inserido, de forma a simplificar o

entendimento pelas equipes distantes. Nesta fase são conduzidas a criação ou atualização da base de informações culturais e de contexto, a criação ou atualização da base de conceitos do UdI, e a obtenção de informações gerais sobre o negócio. A Figura 28 apresenta o diagrama com as atividades desta fase.

Os **artefatos de entrada** para a fase de mapeamento do contexto são: os documentos iniciais do projeto; o plano de projeto; e o plano de comunicação. Os **artefatos de saída** para esta fase são: a base de informações culturais e de contexto atualizada; e a base de conceitos do UdI atualizada.



**Figura 28 - Fase 2 - Mapeamento do contexto**

Estas atividades foram derivadas dos resultados dos estudos de caso e da literatura existente no tema, conforme apresentado na Tabela 29, e descritas em detalhe na seqüência.

<b>Atividade</b>	<b>Fonte</b>
Criar/atualizar base de informações culturais e de contexto	EC1 (#3,5), EC2 (#4), teoria
Criar/atualizar base de conceitos do UdI	EC1 (#5), EC2 (#4)
Distribuir informações de negócio	EC1 (#4), EC2 (#3)

# - Lição aprendida

**Tabela 29 - Atividades relacionadas ao mapeamento do contexto**

### 7.3.1. Criar/atualizar base de informações culturais e de contexto

No desenvolvimento distribuído de software, as diferenças culturais e de contexto podem fazer com que as equipes envolvidas tenham uma compreensão diferenciada do sistema a ser desenvolvido. Conforme apontado por Mahemoff [MAH98], em culturas distintas podem existir medidas e regras de interação social variadas, que se não forem

explicitamente definidas podem causar o desenvolvimento incorreto do software. Um exemplo deste tipo de erro ocorreu em uma sonda espacial da Nasa enviada para Marte [NAS99].

Neste momento do processo é construída (ou atualizada, se existente) uma base de informações culturais e de contexto, para servir como base para as equipes de especificação e desenvolvimento no conhecimento das características do local onde o software será inserido. Uma forma de organizar esta base de informações é proposta por Mahemoff [MAH98]. É importante ressaltar que todas as informações inseridas na base de dados devem ser validadas por pessoas pertencentes à cultura ou contexto em questão, evitando a entrada de dados incorretos.

A partir do momento de sua criação esta base deve ser mantida atualizada até o final do processo de desenvolvimento, através de uma atividade de gerenciamento. Assim assegura-se que todas as novas informações descobertas durante o desenvolvimento poderão ser utilizadas em projetos posteriores.

A base de informações culturais deve ser alimentada por todos os envolvidos no processo de desenvolvimento. Muitas vezes, pode ser difícil para pessoas nativas em uma determinada cultura compreender as próprias características. Nesse momento, os embaixadores podem ser fontes importantes de informações culturais para a base.

Os artefatos de entrada e saída relacionados a esta atividade são apresentados na Tabela 30.

<b>Artefatos de entrada:</b>	- Documentos iniciais do projeto; - Plano de projeto; - Plano de comunicação.
<b>Artefatos de saída:</b>	- Base de informações culturais e de contexto atualizada.

**Tabela 30 - Artefatos de entrada e saída da atividade de criar/atualizar base de informações culturais e de contexto.**

### **7.3.2. Criar/atualizar base de conceitos do universo de informações (UdI)**

Outra atividade na fase de mapeamento de contexto é a criação (ou atualização, se existente) de uma base de conceitos do universo de informações em questão. Dessa forma, o vocabulário específico da aplicação é mapeado, permitindo uma melhor compreensão do ambiente de negócios.

Os conceitos podem ser capturados em um glossário, um thesaurus, ou mesmo através de modelos conceituais [BLE04], do LEL [LEI93] ou ontologias [FEL03]. É importante que sejam de simples consulta e associação com os requisitos do software. Caso tenha sido adotado um dicionário reduzido, a descrição dos conceitos do UdI deve ser realizada preferencialmente utilizando palavras deste dicionário ou conceitos já descritos.

Uma atividade de gerência da base de conceitos do UdI assegura que as informações existentes na base são atualizadas até a finalização do processo de desenvolvimento de software.

A criação da base de conceitos, em geral, deve ser conduzida pelo analista de negócio, dada sua proximidade dos usuários e clientes, e de seu conhecimento do negócio. O analista de aplicação, ao ter contato com os documento do projeto pode gerar demandas para serem incluídas na base de conceitos, bem como verificar se o conteúdo é de simples compreensão. A Tabela 31 apresenta os artefatos de entrada e saída da atividade de criar/atualizar base de conceitos do universo de informações (UdI).

<b>Artefatos de entrada:</b>	- Documentos iniciais do projeto; - Plano de projeto; - Plano de comunicação.
<b>Artefatos de saída:</b>	- Base de conceitos do UdI atualizada.

**Tabela 31 - Artefatos de entrada e saída da atividade de criar/atualizar base de conceitos do universo de informações (UdI).**

### 7.3.3. Distribuição de informações de negócio

Durante a condução das atividades desta fase, diversas informações relacionadas ao negócio em questão emergem. Estas informações, como escopo, necessidades, entre outros devem ser identificadas e transmitidas às equipes envolvidas, de forma a permitir uma melhor compreensão das razões que levaram ao desenvolvimento do software em questão.

O principal responsável pela distribuição das informações de negócio é o analista de negócios, devido à sua proximidade dos usuários e clientes e maior conhecimento das motivações que geraram a demanda pelo projeto de software.

Esta atividade auxilia na identificação de informações para serem incluídas nas bases de conceitos e de informações culturais e de contexto. Da mesma forma, as informações tendem a ser melhor compreendidas com a consulta a estas bases para obter clarificações.

Os artefatos de entrada e saída relacionados a esta atividade são apresentados na Tabela 32.

<b>Artefatos de entrada:</b>	- Documentos iniciais do projeto; - Plano de projeto; - Plano de comunicação.
<b>Artefatos de saída:</b>	- Comunicação incluindo a documentação inicial e informações de negócio.

**Tabela 32 - Artefatos de entrada e saída da atividade de distribuição de informações de negócio.**

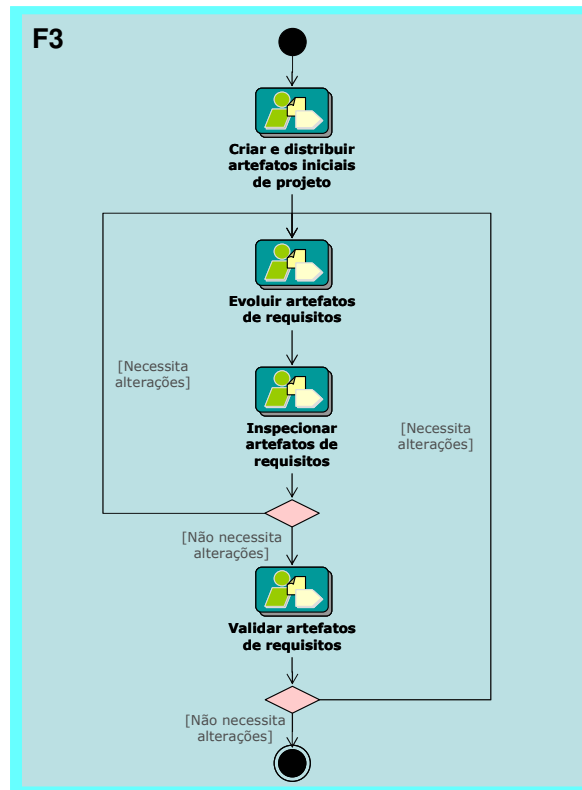
## 7.4. FASE 3 - CRIAÇÃO DA ESPECIFICAÇÃO DE REQUISITOS

As fases um e dois permitem formar as bases para o início da especificação de requisitos. Na terceira fase do modelo de processo proposto, é criada a especificação de



requisitos. Esta fase é composta das atividades de obtenção, evolução, inspeção e validação dos artefatos de requisitos. A Figura 29 apresenta as atividades desta fase.

Os **artefatos de entrada** para a fase de criação da especificação de requisitos são: os documentos iniciais do projeto; o plano de projeto; o plano de comunicação; a base de conceitos do UDI; e a base de informações culturais e de contexto. Os **artefatos de saída** para esta fase são: os artefatos de especificação de requisitos do software; a documentação resultante da inspeção realizada; e a aprovação formal das localidades envolvidas.



**Figura 29 - Fase 3 - Criação da especificação de requisitos**

Estas atividades foram selecionadas com base na literatura e resultados dos estudos de caso, conforme apresentado na Tabela 33.

<b>Atividade</b>	<b>Fonte</b>
Distribuir artefatos iniciais de requisitos	EC1
Evoluir artefatos de requisitos	EC1 (#1)
Inspeccionar artefatos de requisitos	EC1 (#4), EC2 (#1, 4), teoria
Validar artefatos de requisitos	EC1, teoria

# - Lição aprendida

**Tabela 33 - Atividades relacionadas à criação da especificação de requisitos**

Na seqüência são descritas em detalhe cada uma das atividades desta fase do processo.

#### 7.4.1. Criar e distribuir artefatos iniciais de requisitos

A primeira atividade da fase 3 é a criação e distribuição dos artefatos iniciais de requisitos para as equipes de desenvolvimento (ou a equipe de desenvolvimento, se existir apenas uma). Neste momento o analista de negócios é responsável por reunir as informações necessárias para criar os artefatos iniciais de requisitos (conforme definidos para sua responsabilidade no plano de projeto) e enviar estes artefatos para os analistas de aplicação. O conjunto inicial de artefatos pode ser constituído de documentos de alto nível, como um documento Visão/Escopo, ou mesmo incluir uma versão inicial de um documento de especificação de requisitos.

Caso uma versão inicial de um documento de especificação de requisitos seja criada nesta fase, sob responsabilidade do analista de negócios, este deve conduzir a elicitação e negociação dos requisitos junto aos usuários e clientes, bem como sua documentação. A rastreabilidade dos requisitos deve ser mantida, de forma a permitir a identificação de sua origem.

A Tabela 34 apresenta os artefatos de entrada e saída da atividade de criar e distribuir artefatos iniciais de requisitos.

<b>Artefatos de entrada:</b>	<ul style="list-style-type: none"> <li>- Documentos iniciais do projeto;</li> <li>- Plano de projeto;</li> <li>- Plano de comunicação.</li> <li>- Base de conceitos do UdI;</li> <li>- Base de informações culturais e de contexto.</li> </ul>
<b>Artefatos de saída:</b>	<ul style="list-style-type: none"> <li>- Artefatos iniciais de especificação de requisitos do software, conforme definido pelas responsabilidades no plano de projeto. Caso o analista de negócios seja o responsável pelas atividades de elicitação e documentação dos requisitos, é criada e enviada a especificação de requisitos. Caso contrário, a saída desta fase pode ser composta somente de documentos de alto nível;</li> </ul>

**Tabela 34 - Artefatos de entrada e saída da atividade de criar e distribuir artefatos iniciais de requisitos.**

#### 7.4.2. Evoluir artefatos de requisitos

Ao receber os artefatos iniciais de requisitos o analista de aplicação busca, em primeiro lugar, o entendimento destes artefatos e seu contexto. Representantes da equipe de desenvolvimento também são colocados em contato com os artefatos para contextualização. A compreensão dos artefatos iniciais de requisitos é simplificada pela base de conceitos do UdI e, em caso de diferença de idioma entre as equipes, pelo uso de um dicionário reduzido.

Durante esta etapa os artefatos de requisitos são complementados com informações e adaptados para reduzir potenciais fontes de problemas. Dúvidas surgem e devem ser esclarecidas com o analista de negócios ou o grupo de usuários/clientes, caracterizando grande volume de comunicação entre as equipes. Caso a especificação de requisitos deva ser

criada nesta fase, o analista de aplicação deve estar preparado para conduzir a elicitação, análise e documentação dos requisitos à distância dos usuários e clientes. A rastreabilidade dos requisitos deve ser mantida, de forma a permitir a identificação de sua origem.

A comunicação entre as equipes, além de esclarecimentos, visa obter consenso quanto à especificação em construção, alinhando as múltiplas visões em relação aos requisitos.

Os artefatos finais de requisitos devem estar de acordo com os padrões planejados na fase de definições iniciais e registrados no plano de projeto. Os artefatos de entrada e saída relacionados a esta atividade são apresentados na Tabela 35.

<b>Artefatos de entrada:</b>	<ul style="list-style-type: none"> <li>- Documentos iniciais do projeto;</li> <li>- Plano de projeto;</li> <li>- Plano de comunicação.</li> <li>- Base de conceitos do UdI;</li> <li>- Base de informações culturais e de contexto;</li> <li>- Artefatos iniciais de especificação de requisitos do software.</li> </ul>
<b>Artefatos de saída:</b>	- Artefatos de especificação de requisitos do software conforme definido nas responsabilidades no plano de projeto.

**Tabela 35 - Artefatos de entrada e saída da atividade de evoluir artefatos de requisitos.**

#### **7.4.3. Inspeccionar artefatos de requisitos**

Ao finalizar a evolução dos artefatos de requisitos, todas as localidades envolvidas devem realizar uma inspeção nestes artefatos, de forma a assegurar que todos compreendem e são capazes de realizar o processo de desenvolvimento com base neles. Como resultado das inspeções, diversas alterações podem ser necessárias à especificação, de forma a clarificar e melhorar sua qualidade.

A equipe de inspeção de requisitos deve ser composta por uma grande variedade de papéis, como o líder técnico, o gerente de projetos, o engenheiro de testes, entre outros, como destacado pro Firesmith [OPE04]. Existem várias técnicas de inspeção que podem ser utilizadas nesta atividade, como Ad-Hoc, baseada em *Checklist*, baseada em cenários [FUS97] e baseada em perspectiva [SHU00].

A Tabela 36 apresenta os artefatos de entrada e saída desta atividade.

<b>Artefatos de entrada:</b>	<ul style="list-style-type: none"> <li>- Documentos iniciais do projeto;</li> <li>- Base de conceitos do UdI;</li> <li>- Base de informações culturais e de contexto;</li> <li>- Artefatos de especificação de requisitos do software.</li> </ul>
<b>Artefatos de saída:</b>	- Documentação resultante da inspeção realizada.

**Tabela 36 - Artefatos de entrada e saída da atividade de inspeccionar artefatos de requisitos.**

#### **7.4.4. Validar artefatos de requisitos**

Após a evolução dos artefatos de requisitos, estes devem ser validados pelo grupo de usuários/clientes, assegurando que a especificação construída atende às necessidades e objetivos que motivaram o processo de desenvolvimento.

O grupo de usuários/clientes deve verificar os artefatos de requisitos, assegurando que eles refletem os objetivos do projeto. Como a participação de todos os grupos ocorreu durante a atividade de evolução dos artefatos de requisitos, esta etapa é simplificada. Caso a especificação seja validada, esta é definida com a base inicial para o processo de desenvolvimento.

Os artefatos de entrada e saída relacionados a atividade de validar artefatos de requisitos são apresentados na Tabela 37.

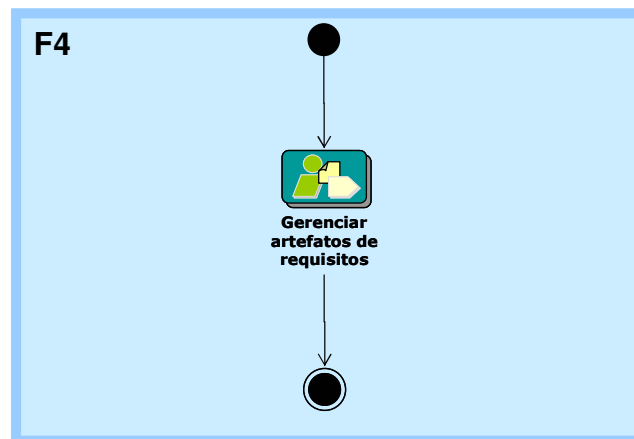
<b>Artefatos de entrada:</b>	- Base de conceitos do UdI; - Base de informações culturais e de contexto; - Artefatos de especificação de requisitos do software.
<b>Artefatos de saída:</b>	- Aprovação formal das localidades e grupos envolvidos.

**Tabela 37 - Artefatos de entrada e saída da atividade de validar artefatos de requisitos.**

## 7.5. FASE 4 - GERENCIAMENTO DE REQUISITOS

A quarta e última fase do modelo de processo é responsável pelo gerenciamento dos requisitos. Nesta fase, é conduzida a manutenção dos artefatos de requisitos, assegurando que estão constantemente alinhados com os objetivos de negócios e atualizados de acordo com as modificações do ambiente (Figura 30).

Os **artefatos de entrada** para a fase de gerenciamento de requisitos são: os artefatos de especificação de requisitos do software inspecionado; a solicitação de alteração nos requisitos do software; a base de conceitos do UdI; e a base de informações culturais e de contexto. Os **artefatos de saída** para esta fase são: a análise de impacto das modificações nos requisitos; a aprovação ou rejeição das alterações; e os artefatos de requisitos atualizados (caso a modificação tenha sido alterada).



**Figura 30 - Fase 4 - Gerenciamento de requisitos**

Esta atividade foi proposta com base na literatura e nos resultados dos estudos de caso, conforme apresentado na Tabela 38 e descrito em detalhe na seqüência.

<b>Atividade</b>	<b>Fonte</b>
Gerenciamento dos artefatos de requisitos	EC2(#3), teoria
# - Lição aprendida	

**Tabela 38 - Atividades relacionadas ao gerenciamento de requisitos**

### 7.5.1. Gerenciar artefatos de requisitos

A atividade de gerenciar os artefatos de requisitos é realizada durante todo o processo de desenvolvimento de software, com o objetivo de gerenciar as mudanças nos requisitos. Nesta atividade é realizado o controle a análise de impacto das modificações. Mudanças nos requisitos podem ocorrer por alterações nas necessidades dos *stakeholders*, no ambiente onde o sistema será utilizado ou na legislação, por exemplo.

No desenvolvimento distribuído de software a gerência de requisitos adquire caráter ainda mais crítico, devendo assegurar que todas as equipes e localidades sejam alertadas de modificações nos requisitos.

Requisições de mudanças iniciadas pelos usuários e clientes devem ser analisadas pelo analista de negócios. Este deve encaminhar a requisição para todos os analistas de aplicação e gerentes de projetos, para análise de impacto e aprovação. Durante a análise de impacto, analistas de negócios e aplicação trabalham em conjunto, para assegurar que as alterações estão refletidas em todas as localidades envolvidas. A aprovação de todas as partes é necessária para assegurar que todos estão cientes da mudança e aptos a executá-la. A rastreabilidade dos requisitos deve ser mantida durante esta fase, bem como usada como entrada para a avaliação de impacto das modificações.

A Tabela 39 apresenta os artefatos de entrada e saída desta atividade.

<b>Artefatos de entrada:</b>	<ul style="list-style-type: none"> <li>- Base de conceitos do UdI;</li> <li>- Base de informações culturais e de contexto;</li> <li>- Artefatos de especificação de requisitos do software inspecionados e aprovados;</li> <li>- Solicitação de alteração nos requisitos do software;</li> </ul>
<b>Artefatos de saída:</b>	<ul style="list-style-type: none"> <li>- Análise de impacto das modificações nos requisitos;</li> <li>- Aprovação ou rejeição das alterações;</li> <li>- Artefatos de requisitos atualizados (caso a modificação tenha sido aceita).</li> </ul>

**Tabela 39 - Artefatos de entrada e saída da atividade de gerenciar artefatos de requisitos.**

## 7.6. ATIVIDADES DE SUPORTE

Após a finalização da fase 2 surgem duas atividades de suporte, de forma a assegurar que as bases de informações culturais e conceitos do UdI. Estas atividades estão apresentadas na Figura 31 e descritas na seqüência.

Os **artefatos de entrada** para as atividades de suporte são: a base de conceitos do UdI; a base de informações culturais e de contexto; e uma solicitação de alterações na base de

conceitos do UdI ou solicitação de alterações na base de informações culturais e de contexto. Os **artefatos de saída** para estas atividades são: a base de conceitos do UdI atualizada com as modificações solicitadas; a base de informações culturais e de contexto atualizada com as modificações solicitadas.

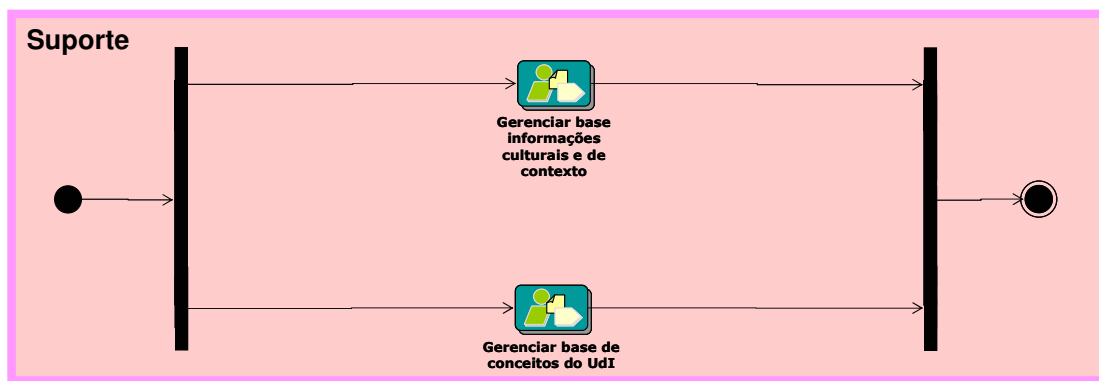


Figura 31 - Atividades de suporte

#### 7.6.1. Gerenciar base de informações culturais e de contexto

Esta atividade é conduzida durante as fases 3 e 4 do modelo de processo, com o objetivo de manter atualizada a base de informações culturais. Para isso, devem ser definidos os responsáveis por gerenciar as alterações nesta base, bem como os mecanismos de alteração.

De forma geral, todos os envolvidos podem solicitar a inclusão, remoção ou atualização de informações culturais ou de contexto. Assim, deve ser definida uma forma para que estes possam encaminhar as mudanças para avaliação.

Os artefatos de entrada e saída relacionados a atividade de gerenciar base de informações culturais e de contexto são apresentados na Tabela 40.

<b>Artefatos de entrada:</b>	- Base de informações culturais e de contexto; - Solicitação de alterações na base de informações culturais e de contexto;
<b>Artefatos de saída:</b>	- Base de informações culturais e de contexto atualizada com as modificações solicitadas.

Tabela 40 - Artefatos de entrada e saída da atividade de gerenciar base de informações culturais e de contexto.

#### 7.6.2. Gerenciar base de conceitos do UdI

Da mesma forma que o gerenciamento de informações culturais, esta atividade é realizada durante as fases 3 e 4 do modelo de processo visando manter atualizada a base de conceitos do UdI. Durante o processo de desenvolvimento, pode ser necessária a inclusão de novos conceito, por exemplo. Assim, devem ser definidos os responsáveis pela manutenção da base de conceitos do UdI.

Como a base de conceitos é ligada a informações de negócio, específicas do ambiente da aplicação em andamento, alterações nesta base devem ser alertadas para todos os analistas

de negócios e de aplicação envolvidos. A Tabela 41 apresenta os artefatos de entrada e saída desta atividade.

<b>Artefatos de entrada:</b>	- Base de conceitos do UdI; - Solicitação de alterações na base de conceitos do UdI.
<b>Artefatos de saída:</b>	- Base de conceitos do UdI atualizada com as modificações solicitadas;

**Tabela 41 - Artefatos de entrada e saída da atividade de gerenciar base de conceitos do UdI.**

## **8. CONSIDERAÇÕES FINAIS**

A engenharia de software vem realizando progressos nos últimos anos. Modelos de processo como o RUP e modelos de maturidade como o SW-CMM têm sido adotados largamente e com sucesso no meio empresarial. Entretanto, novos desafios estão surgindo, exigindo diferentes abordagens para processos existentes. O desenvolvimento distribuído de software é um destes desafios.

A engenharia de requisitos sempre foi reconhecida como uma área crítica no desenvolvimento de software. O seu estudo em ambientes de desenvolvimento distribuído de software oferece grandes oportunidades de pesquisa por ser uma área nova, com crescimento acelerado e poucos trabalhos desenvolvidos no Brasil. Novas técnicas, processos e ferramentas são claramente necessários, aumentando a relevância dos estudos.

Conforme apresentado no capítulo 7, o objetivo geral desta dissertação foi atingido, com a proposta de um modelo de processo de engenharia de requisitos para ambientes de desenvolvimento distribuído de software.

O objetivo específico de aprofundar o conhecimento em engenharia de requisitos, DDS e tópicos relacionados foi atingido, conforme demonstrado pela base teórica apresentada no capítulo 2. As principais categorias e fatores relacionados à engenharia de requisitos em ambientes de desenvolvimento distribuído de software foram identificados de acordo com a literatura, conforme apresentados no capítulo 4.

A elaboração de uma proposta preliminar de processo de engenharia de requisitos para ambientes de desenvolvimento distribuído de software foi realizada conforme descrita no capítulo 5, tendo sido utilizado como base para o estudo de caso 1.

### **8.1. CONTRIBUIÇÕES**

A proposta de um modelo de processo de engenharia de requisitos para ambientes de desenvolvimento distribuído de software visa contribuir na área de engenharia de software, ao preencher uma lacuna existente na área de desenvolvimento distribuído de software, especificamente em engenharia de requisitos, conforme apontado por Zowghi [ZOW02]. Além disso, o estudo apresenta novos dados empíricos relativos à área de engenharia de requisitos em ambientes de desenvolvimento distribuído de software.



Finalmente, este estudo visa contribuir com a prática ao atender uma demanda organizacional crescente por melhorias nos processos de engenharia de requisitos, bem como, por tratar das dificuldades causadas pela distribuição das equipes de desenvolvimento.

## **8.2. LIMITAÇÕES DO ESTUDO**

Uma das principais limitações da pesquisa refere-se ao número de empresas estudadas na parte empírica do estudo, restringindo a generalização dos resultados obtidos. Deve-se, entretanto, destacar que os resultados do estudo de caso foram sustentados na base teórica estudada, o que permite um bom grau de segurança nas conclusões obtidas. Isto também é típico do tipo de pesquisa desenvolvida, exploratória e de base qualitativa, permitindo o uso de inferência nas conclusões obtidas.

Outra limitação do estudo foi a definição de um modelo de processo de alto nível, sem explorar técnicas específicas a serem utilizadas em cada atividade. A opção por esta abordagem foi feita devido a restrições de escopo e tempo impostas ao trabalho.

## **8.3. ESTUDOS FUTUROS**

Identifica-se grande potencial de crescimento nesta linha de pesquisa, centrada no tema de engenharia de requisitos em ambientes de desenvolvimento distribuído de software. Como pesquisas futuras, sugere-se:

- A realização de experimentos visando avaliar quantitativamente o modelo de processo proposto em relação ao uso de modelos tradicionais de engenharia de requisitos;
- Abordar detalhadamente as atividades do modelo de processo, identificando técnicas para utilização de acordo com cada cenário envolvido;
- Um aprofundamento do estudo na área de manutenção, onde problemas como acesso à documentação e *stakeholders*, por exemplo, é dificultada em ambientes de DDS;
- Estudos visando identificar critérios para distribuição de requisitos entre diversas localidades.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [ALR96] AL-RAWAS, Amer; EASTERBROOK, Steve. Communication problem in requirements engineering: A field study. Westminster Conference on Professional Awareness in Software Engineering, 1., 1996, London. **Proceedings...** Fev. 1996. 12 p.
- [ARM01] ARMOUR, Frank; MILLER, Granville. **Advanced Use Case Modeling**. EUA: Addison Wesley, 2001. 425 p.
- [BER98] BERRY, Daniel; LAWRENCE, Brian. Guest Editors' Introduction - Requirements Engineering. **IEEE Software**, California, v. 15, n. 2, p. 26-29, mar. 1998.
- [BLE04] BLEEKER, Araminte; PROPER, Erik; HOPPENBROWERS, Stijn. The Role of Concept Management in System Development - A practical and a theoretical perspective. In. International Conference on Advanced Information Systems Engineering Forum (CAiSE Forum 2004), 16., 2004. Riga, Letônia. **Proceedings...** 2004.
- [BRA03] BRAUN, Andreas; DUTOIT, Allen; BRÜGGE, Bernd. A Software Architecture for Knowledge Acquisition and Retrieval for Global Distributed Teams. In: International Workshop on Global Software Development at ICSE, 2003, Oregon. **Proceedings...** EUA, 2003.
- [CAR99] CARMEL, Erran. **Global Software Teams - Collaborating Across Borders and Time Zones**. EUA: Prentice Hall, 1999. 269 p.
- [CAR01] CARMEL, Erran; AGARWAL, Ritu. Tactical Approaches for Alleviating Distance in Global Software Development. **IEEE Software**, California, v. 18, n. 2, p. 22-29, mar. 2001.
- [DAM00] DAMIAN, Daniela; EBERLEIN, Armin; SHAW, Mildred; GAINES, Brian. Using Different Communication Media in Requirements Negotiation. **IEEE Software**. California, v. 17, n. 3, p. 28-36, may. 2000.
- [DAM02] DAMIAN, Daniela; ZOWGHI, Didar. The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization. In: IEEE Joint International Conference on Requirements Engineering (RE'02), 2002, Essen, Germany. **Proceedings...** IEEE Computer Society, 2002, p. 319-328.

- [DAM02b] DAMIAN, Daniela et al. An industrial experience in process improvement: An early assessment at the Australian Center for Unisys Software. In: International Symposium on Empirical Software Engineering (ISESE'02), 2002, Nara, Japão. **Proceedings...** IEEE Computer Society, 2002, p. 111-123.
- [DAM03] DAMIAN, Daniela et al. Awareness meets requirements management: awareness needs in global software development. In: International Workshop on Global Software Development at ICSE, 2003, Oregon. **Proceedings...** EUA, p. 6-10, 2003.
- [DAV95] DAVIS, Alan. Tracing: A simple necessity neglected. **IEEE Software**. California, v. 12, n. 5, p. 6-7, sep. 1995.
- [DRE04] DREZNER, Daniel. The Outsourcing Bogeyman. Foreign Affairs. Disponível em: [www.foreignaffairs.org](http://www.foreignaffairs.org). Acessado em: 23 de outubro de 2004. Mai. 2004. 7 p.
- [ESP03] ESPINOSA, J. Alberto; CARMEL, Erran. Modeling coordination costs due to time separation in Global Software Teams. In: International Workshop on Global Software Development at ICSE, 2003, Oregon. **Proceedings...** EUA, p. 64-69, 2003.
- [FAV01] FAVELA, Jesús; PEÑA-MORA, Feniosky. An experience in collaborative software engineering education. **IEEE Software**, California, v. 18, n. 2, p. 47-53, mar. 2001.
- [FEL03] FELICISSIMO, Carolina; et all. Geração de ontologias subsidiada pela engenharia de requisitos. In: Workshop on Requirements Engineering. 6., 2003, Piracicaba, Brasil. **Proceedings...** Nov. 2003. p. 255-269.
- [FOW04] FOWLER, Martin. Using an Agile Software Process with Offshore Development. Disponível em: <http://www.martinfowler.com/articles/agileOffshore.html>. Acessado em: 14 de novembro de 2004. Abr. 2004.
- [FRA98] FRANCH, Xavier; BOTELLA, Pere. Putting Non-Functional Requirements into Software Architecture. In: International Workshop on Software Specification and Design (IWSSD'98). 9., 1998, Isobe, Japão. **Proceedings...** IEEE Computer Society, abr. 1998, p. 60.
- [FRE84] FREEMAN, R. Edward. **Strategic Management: A Stakeholder Approach**. Boston: Pitman. 1984.
- [FUS97] FUSARO, Pierfrancesco; LANUBILE, Filippo; VISAGGIO, Giuseppe. A Replicated Experiment To Assess Requirements Inspection Techniques. **Journal of Empirical Software Engineering**, Boston, v. 2, n. 1, Kluwer Academics Publisher, p. 39-57, 1997.

- [GOG96] GOGUEN, Joseph. Formality and Informality in Requirements Engineering. In: International Conference on Requirements Engineering (ICRE '96), 2., 1996, Colorado Springs, EUA. **Proceedings...** IEEE Computer Society. 1996. p. 102-109
- [GOT95] GOTEL, Orlena; FINKELSTEIN, Anthony. Contribution Structures. In: IEEE International Symposium on Requirements Engineering (RE'95), 2., 1995, York, Inglaterra. **Proceedings...** IEEE Computer Society. Mar. 1995, p. 100.
- [GOT97] GOTEL, Orlena; FINKELSTEIN, Anthony. Extended Requirements Traceability: Results of an Industrial Case Study. In: IEEE International Symposium on Requirements Engineering (RE'97), 3., 1997, Annapolis, EUA. **Proceedings...** IEEE Computer Society. Jan. 1997, p. 169-178.
- [HEU01] HEUMANN, Jim. What does "No time for Requirements" mean? **The Rational Edge**. Nov. 2001. 7p.
- [HER01] HERBSLEB, James; MOITRA, Deependra. Guest Editors' Introduction: Global Software Development. . **IEEE Software**, California, v. 18, n. 2, p. 16-20, mar. 2001.
- [HER03] HERBSLEB, James; MOCKUS, Audris. An Empirical Study of Speed and Communication in Globally Distributed Software Development. **IEEE Transactions on Software Engineering**. EUA, v. 29, n. 6. p. 481-494, 2003.
- [IEE98] IEEE, Institute. **IEEE Std. 830-1998 - Recommended Practice for Software Requirements Specifications**. Institute of Electrical and Electronic Engineers. Inc. 1998.
- [KAR98] KAROLAK, Dale Walter. **Global Software Development - Managing Virtual Teams and Environments**. Los Alamitos, EUA: IEEE Computer Society, 1998. 159 p.
- [KOB03] KOBLYNSKI, Rafael; CREIGHTON, Oliver; DUTOIT, Allen; BRUEGGE, Bernd. Building awareness in global software engineering: Using issues as context. In: International Workshop on Global Software Development at ICSE, 2003, Oregon. **Proceedings...** EU, p. 29-33, 2003.
- [KOG00] KOGUT, Bruce; MEITU, Anca. **The emergence of e-Innovation: Insights from Open Source Software Development**. Working Paper. Wharton School. University of Pennsylvania. 2000.
- [KOT98] KOTONYA, Gerald; SOMMERVILLE, Ian. **Requirements Engineering: process and techniques**. EUA: John Wiley. 1998. 294 p.

- [KRU00] KRUCHTEN, Philippe. **The Rational Unified Process: An Introduction**. EUA: Addison-Wesley, 2000. 298p
- [LAY00] LAYZELL, Paul; BRERENTON, O. Pearl; FRENCH, Andrew. Supporting Collaboration in Distributed Software Engineering Teams. In: Asia-Pacific Software Engineering Conference (APSEC'00), 7., 2000, Singapore. **Proceedings...** IEEE Computer Society. Dez. 2000. p. 38-45.
- [LEF00] LEFFINGWELL, Dean; Widrig, Don. **Managing Software Requirements - A Unified Approach**. EUA: Addison-Wesley. 2000. 492 p.
- [LEI93] LEITE, Julio; FRANCO, Ana. A Strategy For Conceptual Model Acquisition. In: IEEE International Symposium on Requirements Engineering, 1., 1993, São Diego. **Proceedings...** IEEE Computer Society. 1993. p. 243-246.
- [LEI98] LEITE, Julio; LEONARDI, Maria. Business Rules as Organizational Policies. In: International Workshop on Software Specification and Design (IWSSD'98). 9., 1998, Isobe, Japão. **Proceedings...** IEEE Computer Society. Abr. 1998. p. 68.
- [LLO02] LLOYD, James W.; ROSSON, Mary B.; ARTHUR, James D. Effectiveness of Elicitation Techniques in Distributed Requirements Engineering. In: IEEE Joint International Conference on Requirements Engineering (RE'02). 2002. Essen, Alemanha. **Proceedings...** IEEE Computer Society. Set. 2002. p. 311-318.
- [LOP03] LOPES, Leandro; MAJDENBAUM, Azriel; AUDY, Jorge. Uma proposta para processo de requisitos em ambientes de desenvolvimento distribuído de software. In: International Workshop on Requirements Engineering. 2003. 6., Piracicaba, Brasil. **Proceedings...** Nov. 2003. p. 329-342.
- [MAH98] MAHEMOFF, Michael J.; JOHNSTON, Lorraine. Software Internationalisation: Implications for Requirements Engineering. In: Australian Workshop On Requirements Engineering, 3., 1998, Geelong, Australia. **Proceedings...** Geelong: Deaking University. 1998. p. 83-90.
- [MAR01] MARQUARDT, Michael J; HORVATH, Lisa. **Global Teams: how top multinationals span boundaries and cultures with high-speed teamwork**. Davies-Black Publishing. Palo Alto, EUA. 2001. 246p.
- [MCC96] McCONNELL, Steve. **Rapid Development**. EUA, Redmond: Microsoft Press, 1996. 647p.

- [NAS99] NASA Jet Propulsion Laboratory. Mars Climate Orbiter Failure Board Releases Report, Numerous NASA Actions Underway In Response. Disponível em: <http://mars.jpl.nasa.gov/msp98/news/mco991110.html>. Acessado em: 14 de novembro de 2004. Nov. 1999.
- [NUS00] NUSEIBEH, Bashar; EASTERBROOK, Steve. Requirements Engineering: A Roadmap. In: Conference on The Future of Software Engineering (ICSE'00), 2000, Limerick, Irlanda, **Proceeding...** ACM-SIGSOFT. Jun 2000. p. 37-45
- [OBE00] OBERG, Roger; PROBASCO, Leslee; ERICSSON, Maria. **Applying Requirements Management with Use Cases**. Rational Software Corporation. 2000. 24 p.
- [OPE04] OPEN Process Framework Repository Organization. **OPEN Process Framework**. Disponível em: <http://www.donald-firesmith.com/>. Acessado em: 14 de novembro de 2004.
- [OSB96] OSBOURNE, Miles; MACNISH, C. K.. Processing Natural Language Software Requirements Specifications. In: International Conference on Requirements Engineering (ICRE '96), 2., 1996, Colorado Springs, EUA. **Proceedings...** IEEE Computer Society. 1996. 229-233.
- [PAU93] PAULK, Mark et al. **Capability Maturity Model for Software, Version 1.1**. Software Engineering Institute. Technical Report. 1993. 82 p.
- [PET97] PETERS, Tom. **Em busca do UAU!**. São Paulo: Harbra. 1997. 302 p.
- [POU99] POULOU DI, Athanasia. Aspects of the Stakeholder Concept and their Implications for Information Systems Development. Hawaii International Conference on System Sciences. 32., Big Island, EUA. **Proceedings...** IEEE Computer Society. January 1999. p. 7030.
- [PRE95] PRESSMAN, Roger S. **Engenharia de Software**. Rio de Janeiro: Makron Books. 1995. 1056 p.
- [PRE01] PRESSMAN, Roger S. **Software Engineering: a practitioner's approach**. EUA: McGraw Hill, 2001. 860 p.
- [PRI02] PRIKLADNICKI, Rafael. **Problemas, desafios e abordagens do processo de desenvolvimento de software**. 2002. Trabalho Individual I, FACIN - PPGCC, PUCRS, Porto Alegre, Jun. 2002. 69 p.
- [PRI02b] PRIKLADNICKI, Rafael. **Desenvolvimento Distribuído de Software e Processos de Desenvolvimento de Software**. 2002. Trabalho Individual II, FACIN - PPGCC, PUCRS, Porto Alegre, Ago. 2002. 66 p.

- [PRI03] PRICKLADNICKI, Rafael; AUDY, Jorge; EVARISTO, Roberto. Requirements Management in Global Software Development: Preliminary Findings from a Case Study in a SW-CMM context. In: International Workshop on Global Software Development at ICSE, 2003, Oregon. **Proceedings...** EUA. mai. 2003.
- [PRI03b] PRIKLADNICKI, Rafael. **MuNDDoS - Um Modelo De Referência Para Desenvolvimento Distribuído De Software**. 2003. Dissertação de Mestrado, FACIN - PPGCC, PUCRS, Porto Alegre, Out. 2003. 143 p.
- [RAM95] RAMESH, Bala; STUBBS, Curtis; POWERS, Tomothy; EDWARDS, Michael. **Implementing Requirements Traceability: A Case Study** In: IEEE International Symposium on Requirements Engineering. 2., 1995, York. **Proceedings...** Inglaterra. Mar. 1995. p. 89-101.
- [RAT03] Rational Software. **The principles behind good requirements - Webinar**. Disponível em: <http://www.rational.com/events/webinararchives/details.jsp?EVENTID=2525&SMSESSION=NO>. Acessado em: 15 de setembro de 2003.
- [SAW99] SAWYER, Pete; SOMMERVILLE, Ian; VILLER, Stephen. Capturing the Benefits of Requirements Engineering. **IEEE Software**, California, v. 16, n. 9, p. 78-85, mar. 1999.
- [SHA99] SHARP, Helen; FILKENSTEIN, Anthony; GALAL, Galal. Stakeholder Identification in the Requirements Engineering Process. In: International Workshop on Database & Expert Systems Applications, 10., 1999, Florença. **Proceedings...** Italia. Set. 1999. p. 387-391.
- [SHU00] SHULL, Forrest; RUS, Ioana; BASILI, Victor. How Perspective-Based Reading Can Improve Requirements Inspections. **IEEE Software**, California, v. 33, n. 7, p. 73-79, jul. 2000.
- [SID96] SIDDIQI, Jawed; SHEKARAN, M. Chandra. Requirements Engineering: the emerging wisdom. **IEEE Software**, California, v. 13, n. 2, p. 15-19, mar. 1996.
- [SOM96] SOMMERVILLE, Ian. **Software Engineering - Fifth Edition**. Harlow: Addison-Weasley. 1996. 745 p.
- [SOM97] SOMMERVILLE, Ian; SAWYER, Peter. **Requirements Engineering - a good practice guide**. EUA: Wiley, 1997
- [STA95] The Standish Group International. **Chaos Report**. Disponível em: [http://www.standishgroup.com/sample\\_research/index.php](http://www.standishgroup.com/sample_research/index.php). Visualizado em: 27 de julho de 2004. Standish Group. 1995. 9p.
- [THA00] THAYER, Richard; DORFMAN, Merlin. **System and Software Requirements Engineering - Second Edition**. Los Alamitos: IEEE Computer Society Press Tutorial, 2000. 528p

- [THO76] THOMPSON, James D. **Dinâmica organizacional - Fundamentos sociológicos da teoria administrativa**. São Paulo: McGraw-Hill do Brasil.1976. p. 79-80.
- [YIN01] YIN, Robert. **Estudo de Caso: planejamento e métodos**. São Paulo: Bookman, 2001, 205 p.
- [ZAN99] ZANLORENCI, Edna. Descrição E Qualificação De Requisitos: Um Modelo Aplicável À Análise E Validação Da Informação. Dissertação de Mestrado, Curso de Pós-Graduação em Informática Aplicada. PUCPR. 1999.
- [ZAV97] ZAVE, Pamela. Classification of Research Efforts in Requirements Engineering. **ACM Computing Surveys**. v. 29, n. 4. Dev. 1997. p. 315-321.
- [ZOW02] ZOWGHI, Didar. Does Global Software Development Need a Different Requirements Engineering Process? In: International Workshop on Global Software Development at ICSE, 2002, Florida. **Proceedings...** EUA, p. 56-58, 2002.



## APÊNDICE 1 - PROTOCOLO PARA ESTUDO DE CASO 1

### Protocolo para Estudo de Caso: Engenharia de Requisitos em ambientes de Desenvolvimento Distribuído de Software (DDS)

---

#### Objetivo

Avaliar a percepção dos envolvidos na aplicação do processo de engenharia de requisitos (ER) para ambientes de desenvolvimento distribuído e da qualidade do artefato gerado.

#### Questão de pesquisa

Como desenvolver a engenharia de requisitos em um ambiente de desenvolvimento distribuído de software visando melhorar a qualidade das especificações geradas e simplificar as tarefas de gerenciamento?

#### Unidade de estudo

Projetos de desenvolvimento de software que utilizaram o processo de engenharia de requisitos para ambientes de desenvolvimento distribuído de software.

#### Característica-chave do método de estudo de caso

Este é um roteiro para o desenvolvimento e aplicação de um instrumento de pesquisa estruturada com questões em escala Lickert que se caracteriza como uma pesquisa do tipo transeccional. O objetivo é avaliar o processo proposto, de acordo com a visão dos envolvidos, bem como a percepção da qualidade do artefato gerado.

#### Organização desse Protocolo

O protocolo será organizado com o segue:

#### 1. Procedimentos

A. Levantamento das questões e estruturação do guia para a entrevista	
Participantes:	Leandro Lopes
Local:	Curso de Pós-Graduação em Ciência da Computação - PUCRS
Datas:	De 03/03/2004 a 14/03/2004

B. Reuniões para revisão do guia para a entrevista	
Participantes:	Jorge Luis Nicolas Audy. Doutor em Administração - UFRGS - 2001 (Pesquisador Sênior)
Local:	Curso de Pós-Graduação em Ciência da Computação - PUCRS
Datas:	03/03/2004, 09/03/2004 e 14/03/2004

C. Autorização da empresa participante	
Participantes:	Diretor de TI da organização estudada
Local:	Porto Alegre
Data:	16/03/2004

D. Validação de Face e Conteúdo	
Participantes:	Marcelo Blois Ribeiro. Doutor em Ciência da Computação - PUC-RJ - 2002 Ricardo Melo Bastos. Doutor em Ciência da Computação - UFRGS - 1998 Gerente de Projetos da organização estudada.
Local:	Porto Alegre
Data:	04/03/2004 e 11/03/2004

E. Pré-teste	
Participantes:	Líder Técnico 1 da organização estudada Líder Técnico 2 da organização estudada
Local:	Porto Alegre
Data:	12/03/2004

F. Envio do instrumento - Questões gerais sobre o projeto (Dimensão 7)	
Participantes:	Projeto 1: <i>Ok</i> - Gerente de Projeto: <i>Ok</i>
Participantes:	Projeto 2: <i>Ok</i> - Gerente de Projeto: <i>Ok</i>

F. Envio do instrumento - Questões referentes à qualidade do artefato gerado (Dimensões 1 e 5)	
Participantes:	Projeto 1: <i>Ok</i> - Analista de Negócios: <i>Ok</i> - Desenvolvedores: <i>Ok</i> - Testadores: <i>Ok</i>
Participantes:	Projeto 2: <i>Ok</i> - Analista de Negócios: <i>Ok</i> - Desenvolvedores: <i>Ok</i> - Testadores: <i>Ok</i>

G. Envio do Instrumento - Questões Referentes ao processo e qualidade do artefato gerado (Dimensões 1, 2, 3, 4, 5, 6)	
Participantes:	Projeto 1: <i>Ok</i> - Gerente de Projeto: <i>Ok</i> - Líder Técnico: <i>Ok</i> - Analista de Aplicação: <i>Ok</i>
Participantes:	Projeto 2: <i>Ok</i> - Gerente de Projeto: <i>Ok</i> - Líder Técnico/Analista de Aplicação: <i>Ok</i>

Relação respondentes x dimensões

Respondentes	Dimensões						
	1	2	3	4	5	6	7
Gerente de Projeto	x	x	x	X	x	x	x
Arquiteto de sistemas	x	x	x	X	x	x	
Líder Técnico	x	x	x	X	x	x	
Desenvolvedores	x				x		
Testadores	x				x		
Analista de Negócios	x				x		
SQA	x	x	x	X	x	x	

## 2. Outros recursos utilizados

### A. Recursos materiais

→ Sistema de gerenciamento de e-mails para envio e recebimento das entrevistas

## 3. Modelo do estudo e Dimensões da Pesquisa

O esquema a seguir representa graficamente os principais aspectos enfocados no desenvolvimento deste trabalho.

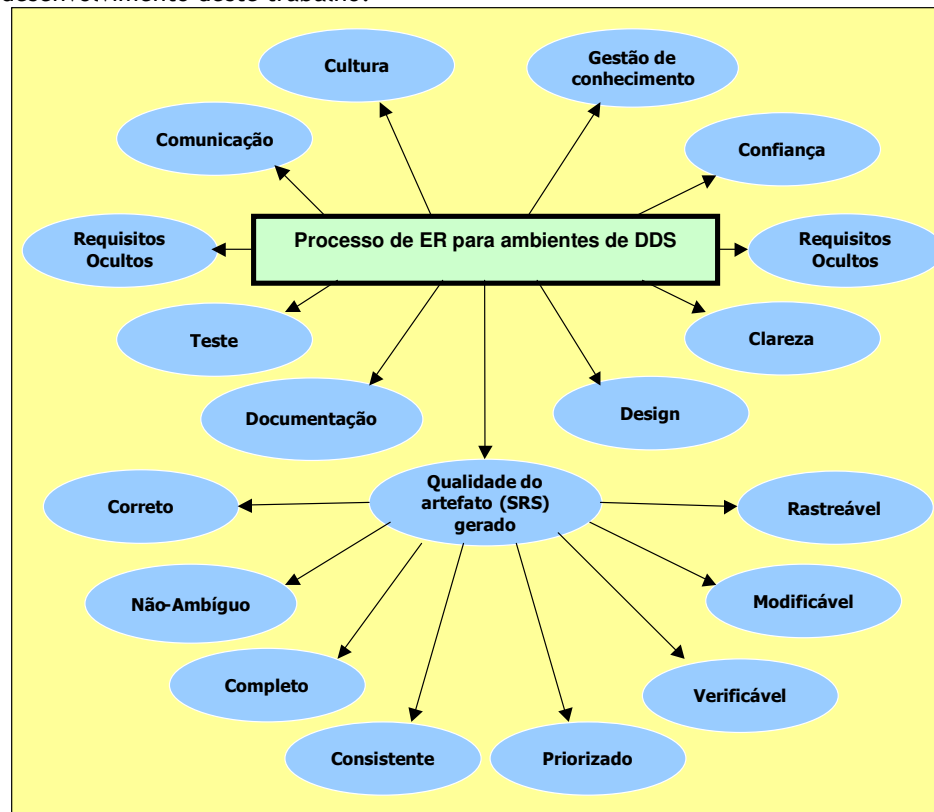


Figura 32 - Aspectos enfocados no trabalho

## 4. Análise de dados

A análise de dados utilizará a técnica de análise de conteúdo [YIN01] e para tabulação dos dados coletados pretende-se utilizar o módulo estatístico do Excel. A coleta de dados

envolve fontes primárias (resultado da aplicação do instrumento) e fontes secundárias (documentação e registros de arquivos). A triangulação dos dados coletados permitirá maior confiabilidade nos resultados obtidos. Esta entrevista insere-se em uma pesquisa de base qualitativa, exploratória, sendo o estudo de caso o principal método de pesquisa, aplicado conforme proposto por [YIN01].

### 5. Dimensões e questões do guia para entrevista semi-estruturada

<b>1 - Dados Demográficos</b>	
<b>Nome</b>	Qual o seu nome?
<b>Idade</b>	Qual a sua idade?
<b>Formação Acadêmica</b>	Qual a sua maior formação acadêmica?
<b>Função</b>	Qual a função que você desempenha na organização?
<b>Experiência na função</b>	Há quantos anos você desempenha a sua função na carreira profissional? E na organização?
<b>Projeto</b>	Em qual o projeto que você trabalhou em que foi aplicado o novo processo de requisitos?
<b>Papel no projeto</b>	Qual o papel que você desempenhou (a) no projeto?

<b>2 - Processo anterior</b>	
<b>Processo anterior</b>	Qual o processo de engenharia de requisitos que você costuma utilizar na organização? Descreva sucintamente as atividades envolvidas.

<b>3 - Processo de engenharia de requisitos para ambientes de desenvolvimento distribuído de software Não-comparativo</b>	
<i>O novo processo de engenharia de requisitos se caracteriza pelas interações realizadas entre as equipes (equipe de especificação, clientes, usuários e equipe de desenvolvimento) durante a especificação dos requisitos, seguindo os guias para escrita de requisitos textuais.</i>	
<b>Comunicação</b> [ZOW02] [DAM02]	<p>A escrita dos requisitos é simples utilizando o novo processo.</p> <p>Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente</p> <p>Os requisitos escritos utilizando o novo processo são simples de serem entendidos pelos usuários.</p> <p>Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente</p>
<b>Gestão de conhecimento</b> [DAM02]	<p>A escrita de requisitos com o novo processo permite a captura de todos os tipos de informação necessários à especificação.</p> <p>Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente</p>
<b>Cultura</b> [ZOW02] [MAH98]	<p>A escrita de requisitos segundo o novo processo simplifica o entendimento dos requisitos por pessoas cuja língua nativa não é o inglês.</p> <p>Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente</p>
<b>Confiança</b> [DAM02]	<p>Ao escrever requisitos utilizando o novo processo, sinto confiança que os requisitos serão entendidos pelos demais envolvidos no processo de desenvolvimento de software (equipe de desenvolvimento, usuários finais, cliente, equipe de teste, etc.).</p> <p>Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente</p>

<b>4 - Processo de engenharia de requisitos para ambientes de desenvolvimento distribuído de software Comparativo</b>	
<i>O novo processo de engenharia de requisitos se caracteriza pelas interações realizadas entre as equipes (equipe de especificação, clientes, usuários e equipe de desenvolvimento) durante a especificação dos requisitos, seguindo os guias para escrita de requisitos textuais.</i>	
<i>As questões desta seção devem ser respondidas levando em consideração o processo de requisitos utilizado anteriormente em sua experiência.</i>	
<b>Comunicação</b> [ZOW02] [DAM02b]	As múltiplas interações contribuem para um melhor <b>entendimento das necessidades</b> envolvidas no desenvolvimento do sistema.  Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente
	A utilização do novo processo promove uma melhor comunicação, fornecendo às equipes uma <b>linguagem comum</b> para comunicação (com relação à estrutura de frase).  Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente
	A comunicação entre as equipes torna-se mais eficiente utilizando-se o novo processo.  Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente
<b>Cultura</b> [ZOW02] [MAH98]	Dificuldades de interpretação e ambigüidades originadas de <b>diferenças culturais</b> , ou de <b>contexto</b> , são reduzidas com a utilização do novo processo.  Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente
	O novo processo de engenharia de requisitos simplifica o entendimento de características específicas do <b>contexto</b> onde o software será utilizado.  Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente
<b>Gestão de conhecimento</b> [DAM02]	A especificação de requisitos resultante do novo processo é mais detalhada.  Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente
	O novo processo de engenharia de requisitos permite um <b>conhecimento</b> mais claro sobre o sistema a ser desenvolvido.  Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente
<b>Confiança</b> [DAM02]	Com a utilização do novo processo a confiança no entendimento dos requisitos entre as equipes aumenta.  Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente
<b>Velocidade</b>	A especificação dos requisitos tornou-se mais rápida com a utilização do novo processo.  Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente
<b>Clareza</b> [OSB96]	Com a utilização do novo processo os requisitos tornaram-se mais claros.  Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente
<b>Requisitos ocultos</b> [DAM02b]	<i>Requisitos ocultos são os requisitos não explicitamente verbalizados pelos usuários ou clientes, porém fundamentais para o sistema a ser desenvolvido.</i>
	O novo processo simplifica a descoberta de requisitos ocultos.  Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente
<b>Estimativa</b> [DAM02b]	Com a utilização do novo processo é possível se realizar uma melhor <b>estimativa</b> do esforço necessário para os próximos estágios de desenvolvimento.  Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente

<b>Teste</b> [DAM02b]	O novo processo traz benefícios para as atividades de teste. Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente
<b>Documentação</b> [DAM02b]	O novo processo promove melhorias na <b>documentação</b> do software desenvolvido. Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente
<b>Benefícios para o processo de desenvolvimento</b>	
<b>Etapas de desenvolvimento afetadas</b> [DAM02b]	O novo processo de engenharia de requisitos representa um benefício para as seguintes etapas do processo de desenvolvimento:  <div style="text-align: center;"><i>Planning.</i></div> Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente  <div style="text-align: center;"><i>Developing.</i></div> Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente  <div style="text-align: center;"><i>Stabilizing.</i></div> Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente

<b>5 - Qualidade do Documento de Especificação dos Requisitos</b>	
<b>Correto</b> [IEE98]	<i>Segundo [IEE98], um SRS é correto se, e somente se, todos os requisitos nele contidos devem ser atendidos.</i> Com a utilização do novo processo o SRS torna-se mais <b>correto</b> . Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente
<b>Não-ambíguo</b> [IEE98]	<i>Segundo [IEE98], um SRS é não ambíguo se, e somente se, todos os requisitos nele contidos tenham somente uma interpretação.</i> Com a utilização do novo processo as <b>ambigüidades</b> se reduzem. Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente
<b>Completo</b> [IEE98]	<i>Segundo [IEE98], um SRS é completo se e somente se contém:</i> a) <i>todos os requisitos significativos;</i> b) <i>definições de resposta do software para todos os classes de entrada possíveis;</i> c) <i>referências e nomes para todas as figuras, tabelas e diagramas, bem como definição de todos os termos e unidades de medida.</i> Com a utilização do novo processo o SRS torna-se mais <b>completo</b> . Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente
<b>Consistente</b> [IEE98]	<i>Segundo [IEE98], consistência se refere a consistência interna. Se um SRS não está de acordo com documentos de mais alto nível, então ele não está correto.</i> Com a utilização do novo processo o SRS torna-se mais <b>consistente</b> . Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente
<b>Priorizado</b> [IEE98]	<i>Segundo [IEE98], um SRS é priorizado se cada requisito nele contido possuir um identificador para indicar a importância ou a estabilidade do requisito em particular.</i> Com a utilização do novo processo o SRS é mais facilmente <b>priorizado</b> . Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente
<b>Verificável</b> [IEE98]	<i>Segundo [IEE98], um SRS é verificável se, e somente se, todos os requisitos nele contidos são verificáveis. Um requisito é verificável se, e somente se, existir um processo finito e viável, através do qual uma pessoa ou máquina possa verificar que o software atende ao requisito.</i> Com a utilização do novo processo o SRS torna-se mais <b>verificável</b> . Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente

<b>Modificável</b> [IEE98]	<i>Segundo [IEE98], um SRS é modificável se, e somente se, sua estrutura e estilo estiverem de forma que qualquer modificação nos requisitos possa ser realizada facilmente, completamente e consistentemente, retendo a estrutura e estilo.</i>
	Com a utilização do novo processo o SRS torna-se mais <b>modificável</b> .  Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente
<b>Rastreável</b> [IEE98]	<i>Segundo [IEE98], um SRS é rastreável se a origem de cada um dos requisitos é clara e se ele simplifica a referência de cada requisito na documentação futura de desenvolvimento ou modificação.</i>
	O SRS é mais facilmente <b>rastreável</b> com a utilização do novo processo.  Discordo totalmente <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Concordo Totalmente

<b>6 - Questões abertas</b>	
<b>Vantagens</b>	Cite as 3 principais vantagens do novo processo.
<b>Desvantagens</b>	Cite as 3 principais desvantagens do novo processo.
	Você substituiria o processo atual pelo novo?  Sim <input type="checkbox"/> <input type="checkbox"/> Não

<b>7 - Questões referentes ao projeto</b>	
<b>Nome</b>	Qual o nome do projeto?
<b>Objetivos</b>	Quais os objetivos do projeto?
<b>Tipo (novo ou melhoria)</b>	O projeto envolve um software novo ou melhorias em um software existente?
<b>Tecnologias</b>	Quais as tecnologias utilizadas no projeto?
<b>Duração</b>	Qual a estimativa de duração do projeto?
<b>Recursos</b>	Quantas pessoas estão envolvidas no projeto?
<b>Distribuição</b>	Quais as localidades envolvidas? Onde se localizava a equipe responsável pela especificação dos requisitos? Onde se localizava a equipe responsável pela revisão da especificação? Onde se localizava a equipe responsável pelo projeto do software? Onde se localizava a equipe responsável pela codificação do software? Onde se localizava a equipe responsável pelo teste do software? Onde se localizavam os clientes do sistema a ser desenvolvido? Onde se localizavam os usuários do sistema a ser desenvolvido?

## APÊNDICE 2 - PROTOCOLO PARA ESTUDO DE CASO 2

### Protocolo para Estudo de Caso: Engenharia de Requisitos em ambientes de Desenvolvimento Distribuído de Software (DDS)

---

#### Objetivo

Identificar o impacto dos fatores relacionados à engenharia de requisitos em ambientes de desenvolvimento distribuídos de software na especificação de requisitos.

#### Questão de pesquisa

Como desenvolver a engenharia de requisitos em um ambiente de desenvolvimento distribuído de software visando melhorar a qualidade das especificações geradas e simplificar as tarefas de gerenciamento?

#### Característica-chave do método de estudo de caso

Este é um roteiro para o desenvolvimento e aplicação de uma entrevista estruturada com questões, que se caracteriza como uma pesquisa do tipo transeccional. O objetivo é identificar os problemas ampliados na especificação de requisitos devido a fatores relacionados à engenharia de requisitos em DDS.

#### Organização desse Protocolo

O protocolo será organizado com o segue:

#### 1. Procedimentos

A. Levantamento das questões e estruturação do guia para a entrevista	
Participantes:	Leandro Lopes
Local:	Curso de Pós-Graduação em Ciência da Computação - PUCRS
Datas:	15 de junho a 20 de julho

B. Reuniões para revisão do guia para a entrevista	
Participantes:	Jorge Luis Nicolas Audy. Doutor em Administração - UFRGS - 2001 (Pesquisador Sênior)
Local:	Curso de Pós-Graduação em Ciência da Computação - PUCRS
Datas:	26 de junho e 18 de julho de 2004

D. Validação de Face e Conteúdo	
Participantes:	Marcelo Blois Ribeiro. Doutor em Ciência da Computação - PUC-RJ - 2002 Miriam Saião. Doutoranda em Ciência da Computação - PUC-RJ
Local:	Curso de Pós-Graduação em Ciência da Computação - PUCRS
Data:	agosto de 2004



E. Pré-teste	
Participantes:	Engenheiro de requisitos da organização selecionada
Local:	Porto Alegre
Data:	28/07/2004

F. Entrevista	
Participantes:	Leandro Lopes e engenheiros de requisitos entrevistados
Local:	Porto Alegre
Data:	02/08/2004 a 10/08/2004

## 2. Relação respondentes x dimensões

Respondentes	Dimensões		
	1	2	3
Engenheiros de Requisitos	x	x	x

## 3. Outros recursos utilizados

### A. Recursos materiais

→ Sistema de gerenciamento de e-mails para envio e recebimento das entrevistas

## 4. Modelo do estudo e Dimensões da Pesquisa

O esquema a seguir representa graficamente os principais aspectos enfocados no desenvolvimento deste trabalho.

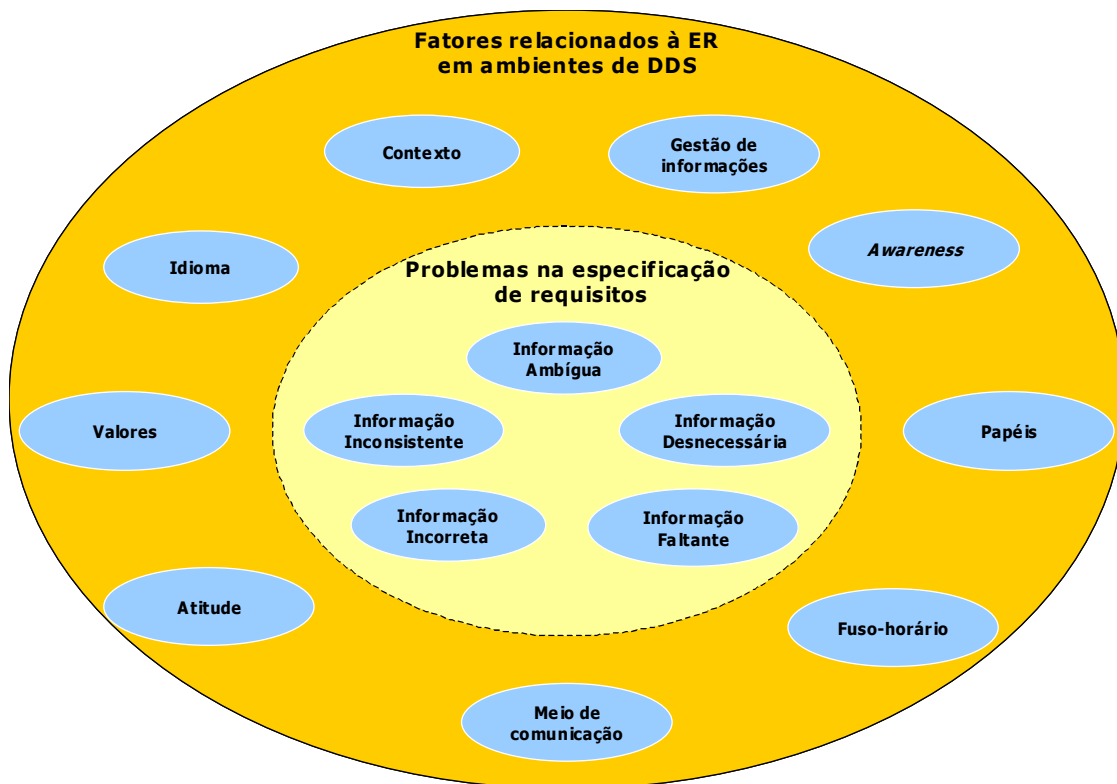


Figura 33 - Aspectos enfocados no trabalho

## 5. Análise de dados

A análise de dados utilizará a técnica de análise de conteúdo [YIN01] e para tabulação dos dados coletados pretende-se utilizar o módulo estatístico do Excel. A coleta de dados envolve fontes primárias (resultado da aplicação do instrumento) e fontes secundárias (documentação e registros de arquivos). A triangulação dos dados coletados permitirá maior confiabilidade nos resultados obtidos. Esta entrevista insere-se em uma pesquisa de base qualitativa, exploratória, sendo o estudo de caso o principal método de pesquisa, aplicado conforme proposto por [YIN01].

## 6. Relação entre Construtos e Autores

Construto	Referências Relacionadas
Problemas na especificação de requisitos	[SHU00]
Contexto	[DAM02] [ZOW02] [MAH98]
Valores	[MAH98] [ZOW02]
Atitudes	[DAM02]
Idioma	[DAM02] [ZOW02]
Meio de comunicação	[LLO02] [ZOW02] [DAM00]
Fuso-horário	[DAM02][ZOW02]
<i>Awareness</i>	[DAM02] [ZOW02] [DAM03]
Gestão de informações	[MAH98] [ZOW02]
Papéis	[DAM02]

## 7. Dimensões e questões do instrumento de pesquisa

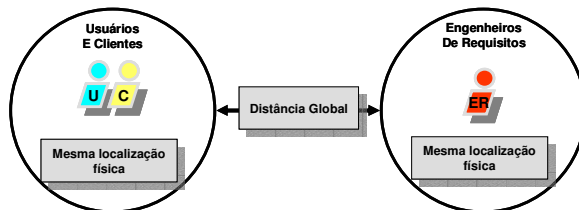
1 - Dados Demográficos	
Nome	Qual o seu nome?
Idade	Qual a sua idade?
Formação Acadêmica	Qual a sua maior formação acadêmica?
Função	Qual a função que você desempenha na organização?
Experiência na função	Há quantos anos você desempenha a sua função na carreira profissional? E na organização?

## 2 - Dispersão entre Engenheiro de Requisitos e Grupo de Usuários e Clientes

### Cenário 1:

Considerando um cenário de desenvolvimento distribuído de software (DDS) com a seguinte distribuição (Figura 34):

- Engenheiros de requisitos em uma mesma localização física;
- Usuários e clientes em uma mesma localização física, distante globalmente dos Engenheiros de Requisitos;



**Figura 34 - Engenheiro de requisitos distribuído em relação a usuários e clientes**

Exemplo 1: Engenheiros de requisitos em São Paulo e usuários e clientes em um prédio em Londres.

Exemplo 2: Engenheiros de requisitos em Bangalore e usuários e clientes em Tóquio

Para o cenário descrito acima, identifique quais dos problemas na especificação de requisitos são ampliados em virtude dos FATORES listados na tabela abaixo, relacionados ao ambiente de desenvolvimento distribuído de software (marque com um X utilizando caneta azul). Caso identifique algum FATOR não listado, liste no espaço ao final da tabela.

**Tabela 1 - FATORES relacionados ao DDS X problemas na especificação**

FATOR relacionado ao desenvolvimento distribuído de software	Problemas na Especificação <sup>6</sup> ampliados pelo FATOR				
	Informação Ambígua	Informação Faltante	Informação Inconsistente	Informação Incorreta	Informação Desnecessária
Idioma nativo dos <i>stakeholders</i>					
Diferença de fuso horário entre as localidades dos <i>stakeholders</i>					
Valores pessoais dos <i>stakeholders</i> (influenciada pela cultura)					
Atitudes e forma de agir (influenciada pela cultura)					
Comunicação entre as localidades via e-mail					
Comunicação entre as localidades via teleconferência					
Gerenciamento de informações dispersas em múltiplas localidades					
Consciência das mudanças (no negócio, equipe, etc.) ou progressos nas localidades envolvidas ( <i>Awareness</i> )					
Dificuldade de identificação dos <i>stakeholders</i>					
Diferenças de contexto entre as localidades					

<sup>6</sup> **Informação Ambígua:** Múltiplas interpretações causadas pela utilização de diferentes termos para a mesma característica ou múltiplos significados de um termo em um contexto particular.

**Informação Faltante:** Qualquer requisito significativo relacionado à funcionalidade, performance, restrições de projeto, atributos ou interfaces externas não incluído.

**Informação Inconsistente:** Dois ou mais requisitos conflitantes entre si.

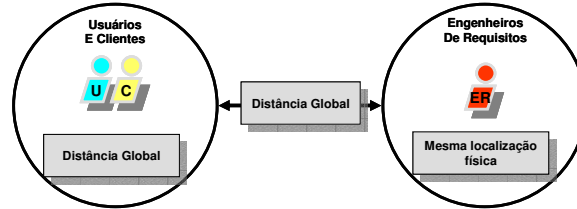
**Informação Incorreta:** Informação falsa, que não reflete a realidade ou necessidade sendo especificada.

**Informação Desnecessária:** Informação desnecessária ou não utilizada.

**Cenário 2:**

Considerando um cenário de desenvolvimento distribuído de software com a seguinte distribuição (Figura 35):

- Engenheiros de Requisitos em uma mesma localização física;
- Usuários e Clientes distribuídos globalmente e distantes globalmente dos engenheiros de requisitos.



**Figura 35 - Usuários e clientes distribuídos entre si e em relação aos Engenheiros de Requisitos**

Exemplo 1: Engenheiros de requisitos em Porto Alegre, usuários no Canadá, Inglaterra e Japão e clientes nos EUA.

Exemplo 2: Engenheiros de requisitos em Bangalore, usuários dispersos na Ásia e Oceania e clientes na Austrália.

Na sua percepção, a distribuição do grupo de usuários e clientes (além da distribuição entre engenheiros de requisitos e usuários e clientes) amplia os problemas causados pelos fatores, conforme representado na Tabela 1?

SIM        NÃO

Você identifica algum problema na especificação devido aos FATORES relacionado ao desenvolvimento distribuído de software que emerge devido à distribuição do grupo usuários e clientes (além da distribuição entre engenheiros de requisitos e usuários e clientes)?

- NÃO  
 SIM. Quais (Marque com um X na tabela 1, utilizando caneta vermelha)?

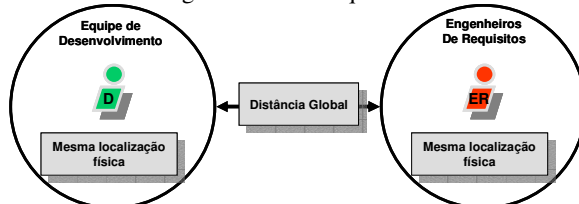
Obs.: Caso seja necessário invalidar uma resposta marcada, utilize um círculo ao redor do X.

### 3 - Dispersão entre Engenheiro de Requisitos e Desenvolvedores

#### Cenário 3:

Considerando um cenário com a seguinte distribuição, como exemplificado na Figura 36:

- Engenheiros de requisitos em uma mesma localização física;
- Equipe de desenvolvimento (incluindo projeto, codificação e testes) em uma mesma localização física, distante globalmente dos Engenheiros de Requisitos.



**Figura 36 - Engenheiros de requisitos distribuído globalmente em relação à equipe de desenvolvimento**

Exemplo 1: Engenheiros de requisitos em Porto Alegre e equipe de desenvolvimento em Bangalore.

Exemplo 2: Engenheiros de requisitos em Londres e equipe de desenvolvimento em Porto Alegre..

Para o cenário descrito acima, identifique quais dos problemas na especificação de requisitos são ampliados em virtude dos FATORES listados na tabela abaixo, relacionados ao ambiente de desenvolvimento distribuído de software (marque com um X, utilizando caneta azul). Caso identifique algum FATOR não listado, liste no espaço ao final da tabela.

**Tabela 2 - FATORES relacionados ao DDS X problemas na especificação**

FATOR relacionado ao desenvolvimento distribuído de software	Problemas na Especificação <sup>7</sup> ampliados pelo FATOR				
	Informação Ambígua	Informação Faltante	Informação Inconsistente	Informação Incorreta	Informação Desnecessária
Idioma nativo dos <i>stakeholders</i>					
Diferença de fuso horário entre as localidades dos <i>stakeholders</i>					
Valores pessoais dos <i>stakeholders</i> (influenciada pela cultura)					
Atitudes e forma de agir (influenciada pela cultura)					
Comunicação entre as localidades via e-mail					
Comunicação entre as localidades via teleconferência					
Gerenciamento de informações dispersas em múltiplas localidades					
Consciência das mudanças (no negócio, equipe, requisitos etc.) ou progressos nas localidades envolvidas ( <i>Awareness</i> )					
Dificuldade de identificação dos <i>stakeholders</i>					
Diferenças de contexto entre as localidades					

<sup>7</sup> **Informação Ambígua:** Múltiplas interpretações causadas pela utilização de diferentes termos para a mesma característica ou múltiplos significados de um termo em um contexto particular.

**Informação Faltante:** Qualquer requisito significativo relacionado à funcionalidade, performance, restrições de projeto, atributos ou interfaces externas não incluído.

**Informação Inconsistente:** Dois ou mais requisitos conflitantes entre si.

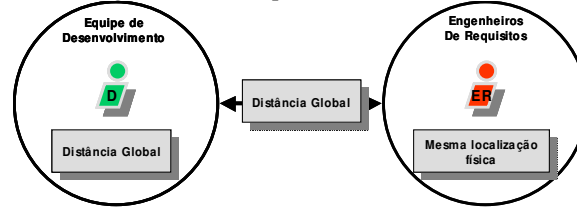
**Informação Incorreta:** Informação falsa, que não reflete a realidade ou necessidade sendo especificada.

**Informação Desnecessária:** Informação desnecessária ou não utilizada.

**Cenário 4:**

Considerando um cenário com a seguinte distribuição, como exemplificado na Figura 37:

- Engenheiros de requisitos co-localizados;
- Equipe de desenvolvimento (incluindo projeto, codificação e testes) distribuída globalmente e distante globalmente dos engenheiros de requisitos;



**Figura 37 - Equipe de desenvolvimento distribuída entre si e em relação aos engenheiros de requisitos**

Exemplo 1: Engenheiros de requisitos em Toronto, equipe de desenvolvimento em Toronto, Porto Alegre e Pequim.

Exemplo 2: Engenheiros de requisitos em Porto Alegre, desenvolvimento (codificação) em Porto Alegre e Bangalore, testes em Lisboa.

Na sua percepção, a distribuição da equipe de desenvolvimento (além da distribuição entre engenheiros de requisitos e equipe de desenvolvimento) amplia os problemas causados pelos fatores, conforme representado na Tabela 2?

SIM   NÃO

Você identifica algum problema na especificação devido aos FATORES relacionado ao desenvolvimento distribuído de software que emerge devido à distribuição da equipe de desenvolvimento (além da distribuição entre engenheiros de requisitos e equipe de desenvolvimento)?

NÃO

SIM. Quais (Marque com um X na tabela 2, utilizando caneta vermelha)?

## APÊNDICE 3 - DADOS CONSOLIDADOS DO ESTUDO DE CASO 2

Cenário 1	Problemas na especificação ampliados pelo FATOR				
Fator relacionado ao DDS	Informação Ambígua	Informação Faltante	Informação Inconsistente	Informação Incorreta	Informação Desnecessária
<b>Idioma nativo dos <i>stakeholders</i></b>	6	3	5	3	1
<b>Diferença de fuso horário entre as localidades dos <i>stakeholders</i></b>	0	1	1	2	0
<b>Valores pessoais dos <i>stakeholders</i> (influenciada pela cultura)</b>	3	5	3	1	2
<b>Atitudes e forma de agir (influenciada pela cultura)</b>	4	5	3	2	2
<b>Comunicação entre as localidades via e-mail</b>	3	1	1	0	1
<b>Comunicação entre as localidades via teleconferência</b>	5	2	2	2	0
<b>Gerenciamento de informações dispersas em múltiplas localidades</b>	3	5	4	4	1
<b>Consciência das mudanças (no negócio, equipe, etc.) ou progressos nas localidades envolvidas (<i>Awareness</i>)</b>	2	6	4	6	0
<b>Dificuldade de identificação dos <i>stakeholders</i></b>	2	5	5	4	3
<b>Diferenças de contexto entre as localidades</b>	1	5	4	5	2

Cenário 2	Problemas na especificação ampliados pelo FATOR				
Fator relacionado ao DDS	Informação Ambígua	Informação Faltante	Informação Inconsistente	Informação Incorreta	Informação Desnecessária
Idioma nativo dos <i>stakeholders</i>	0	1	0	0	1
Diferença de fuso horário entre as localidades dos <i>stakeholders</i>	1	2	2	1	0
Valores pessoais dos <i>stakeholders</i> (influenciada pela cultura)	1	0	0	0	0
Atitudes e forma de agir (influenciada pela cultura)	1	0	0	1	0
Comunicação entre as localidades via e-mail	1	0	2	2	0
Comunicação entre as localidades via teleconferência	0	0	2	3	0
Gerenciamento de informações dispersas em múltiplas localidades	1	1	0	0	1
Consciência das mudanças (no negócio, equipe, etc.) ou progressos nas localidades envolvidas ( <i>Awareness</i> )	2	0	1	0	0
Dificuldade de identificação dos <i>stakeholders</i>	1	0	0	1	0
Diferenças de contexto entre as localidades	1	0	0	0	1



Cenário 3	Problemas na especificação ampliados pelo FATOR				
Fator relacionado ao DDS	Informação Ambígua	Informação Faltante	Informação Inconsistente	Informação Incorreta	Informação Desnecessária
Idioma nativo dos <i>stakeholders</i>	5	5	5	4	2
Diferença de fuso horário entre as localidades dos <i>stakeholders</i>	1	3	1	1	0
Valores pessoais dos <i>stakeholders</i> (influenciada pela cultura)	5	5	1	1	0
Atitudes e forma de agir (influenciada pela cultura)	5	3	1	1	0
Comunicação entre as localidades via e-mail	4	5	2	3	1
Comunicação entre as localidades via teleconferência	4	3	2	4	0
Gerenciamento de informações dispersas em múltiplas localidades	4	4	4	5	1
Consciência das mudanças (no negócio, equipe, etc.) ou progressos nas localidades envolvidas ( <i>Awareness</i> )	2	4	5	5	1
Dificuldade de identificação dos <i>stakeholders</i>	2	5	5	6	1
Diferenças de contexto entre as localidades	3	6	3	3	2

Cenário 4	Problemas na especificação ampliados pelo FATOR				
Fator relacionado ao DDS	Informação Ambígua	Informação Faltante	Informação Inconsistente	Informação Incorreta	Informação Desnecessária
Idioma nativo dos <i>stakeholders</i>	1	0	0	0	0
Diferença de fuso horário entre as localidades dos <i>stakeholders</i>	0	0	2	0	1
Valores pessoais dos <i>stakeholders</i> (influenciada pela cultura)	0	1	2	0	1
Atitudes e forma de agir (influenciada pela cultura)	1	0	1	0	1
Comunicação entre as localidades via e-mail	0	0	1	1	1
Comunicação entre as localidades via teleconferência	1	0	2	2	0
Gerenciamento de informações dispersas em múltiplas localidades	0	1	0	0	1
Consciência das mudanças (no negócio, equipe, etc.) ou progressos nas localidades envolvidas ( <i>Awareness</i> )	0	1	0	0	0
Dificuldade de identificação dos <i>stakeholders</i>	0	0	0	0	0
Diferenças de contexto entre as localidades	1	0	0	0	1

## APÊNDICE 4 - ARTIGOS PUBLICADOS

Leandro Lopes; Jorge Audy. Em busca de um modelo de referência para engenharia de requisitos em ambientes de desenvolvimento distribuído de software. In: Workshop on Requirements Engineering (WER03). 6., 2003, Piracicaba. **Proceedings...** Brasil. Nov. 2003.

Leandro Lopes; Azriel Majdenbaum; Jorge Audy. Uma proposta para processo de requisitos em ambientes de desenvolvimento distribuído de software. In: Workshop on Requirements Engineering (WER03). 6., 2003, Piracicaba. **Proceedings...** Brasil. Nov. 2003.

Leandro Lopes; Rafael Prikladnicki; Azriel Majdenbaum; Jorge Audy. Distributed Requirement Specification: Minimizing the effect of geographic dispersion. In: International Conference on Enterprise Information Systems (ICEIS 2004). 2004, Porto. **Proceedings...** Portugal. Abr. 2004.

Leandro Lopes; Jorge Audy. Towards a reference model for requirement engineering in distributed software development. In: International Conference on Advanced Information Systems Engineering. 16., 2004, Riga. **Proceedings...** Letônia. Jun. 2004.

Rafael Prikladnicki; Leandro Lopes; Roberto Evaristo; Jorge Audy. Desenvolvimento Distribuído de Software: Um Modelo de Classificação dos Níveis de Dispersão dos Stakeholders. In: Simpósio Brasileiro de Sistemas de Informação. 1., 2004. Porto Alegre. **Proceedings...** Brasil. Out. 2004.

## **O Grupo de Pesquisa MuNDDoS na Internet**

<http://www.inf.pucrs.br/munddos>

---

Esta pesquisa foi parcialmente financiada pelo Convênio Dell/PUCRS, através da Lei de Informática Brasileira (Lei nº. 8.248/91).

---

---

Este trabalho foi digitado conforme o *Guia para Apresentação de Trabalhos Acadêmicos, Teses e Dissertações* da Biblioteca Central Irmão José Otão da PUCRS, segundo a NBR 14724 proposta pela ABNT, atualizado em 06 de agosto de 2003.

---



