

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

UMA METODOLOGIA PARA AQUISIÇÃO DE
CONHECIMENTO EM REUNIÕES DE PROJETOS DE
DESENVOLVIMENTO DE SOFTWARE

ANDERSON RICARDO YANZER CABRAL

Tese apresentada como requisito
parcial à obtenção do grau de
Doutor em Ciência da Computação
na Pontifícia Universidade Católica
do Rio Grande do Sul.

Orientador: Prof. Dr. Marcelo Blois Ribeiro

Porto Alegre

2012

Dados Internacionais de Catalogação na Publicação (CIP)

C117m Cabral, Anderson Ricardo Yanzer

Uma metodologia para aquisição de conhecimento em reuniões de projetos de desenvolvimento de software / Anderson Ricardo Yanzer Cabral. – Porto Alegre, 2012.

174 f.

Tese (Doutorado) – Fac. de Informática, PUCRS.

Orientador: Prof. Dr. Marcelo Blois Ribeiro.

1. Informática. 2. Engenharia de Software. 3. Gestão do Conhecimento. I. Ribeiro, Marcelo Blois. II. Título.

CDD 005.1

**Ficha Catalográfica elaborada pelo
Setor de Tratamento da Informação da BC-PUCRS**



Pontifícia Universidade Católica do Rio Grande do Sul
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

TERMO DE APRESENTAÇÃO DE TESE DE DOUTORADO

Tese intitulada "**Uma Metodologia para Aquisição de Conhecimento em Reuniões de Projetos de Desenvolvimento de Software**", apresentada por Anderson Ricardo Yanzer Cabral, como parte dos requisitos para obtenção do grau de Doutor em Ciência da Computação, Sistemas de Informação, aprovada em 26/01/2012 pela Comissão Examinadora:

Prof. Dr. Marcelo Blois Ribeiro -
Orientador

PPGCC/PUCRS

Prof. Dr. José Palazzo Moreira de Oliveira -

UFRGS

Prof. Dr. Ricardo Choren Noya -

IME - RJ

Prof. Dr. Ricardo Melo Bastos -

PPGCC/PUCRS

Homologada em 08/06/2012, conforme Ata No. 012, pela Comissão Coordenadora.

Prof. Dr. Paulo Henrique Lemelle Fernandes
Coordenador.

PUCRS

Campus Central
Av. Ipiranga, 6681 - P. 32 - sala 507 - CEP: 90619-900
Fone: (51) 3320-3611 - Fax (51) 3320-3621
E-mail: ppgcc@pucrs.br
www.pucrs.br/faciny/pos

*“Um homem é um sucesso se pula da cama pela manhã,
vai dormir à noite e, neste meio tempo,
faz o que gosta”*

Bob Dylan

*“Sabedoria é saber o que fazer,
habilidade é saber como fazer,
virtude é fazer”*

David Starr Jordan

AGRADECIMENTOS

Ao meu orientador, Professor Marcelo Blois Ribeiro, por seus conhecimentos, presteza, críticas nas horas adequadas, dedicação, muitos *insights* e respeito cultivado nestes mais de quatro anos de trabalho. A convivência neste período resultou para mim em uma ótima experiência acadêmica e profissional.

Aos membros da banca, professores José Palazzo, Ricardo Choren Noya e Ricardo Bastos, por aceitarem o convite de participação na avaliação deste trabalho.

A minha namorada Jana, pela compreensão, dedicação, paciência e carinho demonstrado nestes anos de convivência.

Aos meus familiares, em especial aos meus irmãos, irmã, sobrinhos e afilhados, pela torcida e apoio durante este período de muitas ausências.

Aos colegas da ULBRA pelo apoio e compreensão na dupla jornada e em especial a colega e amiga Vanessa Lindemann pela ajuda e incentivo.

Aos colegas de doutorado, Ana Paula Lemke, Maurício Escobar e Rodrigo Noll, pela parceria e ajuda em muitas tarefas.

A minha professora orientadora no mestrado, professora Lucia Giraffa, que segue como uma referência acadêmica e profissional.

Ao convênio Dell-PUCRS por proporcionar a bolsa de estudos durante o período do doutorado.

E, em especial aos meus pais, pela confiança e exemplo.

UMA METODOLOGIA PARA AQUISIÇÃO DE CONHECIMENTO EM REUNIÕES DE PROJETOS DE DESENVOLVIMENTO DE SOFTWARE

RESUMO

A engenharia de software é um domínio altamente orientado ao conhecimento, no qual os fatores de sucesso estão relacionados com a experiência das pessoas envolvidas nas diversas fases e atividades do processo. O conhecimento na engenharia de software é disperso, de proporção imensa e de crescimento contínuo, e a gestão deste conhecimento nas organizações é uma área ampla com muitas disciplinas que podem influenciar nos seus resultados. Dentre as questões que envolvem a gestão do conhecimento no desenvolvimento de software está a preferência pelo uso do conhecimento em seu modo tácito ou explícito. Essa preferência poderá definir, entre outros aspectos, a opção pela redução da explicitação do conhecimento que circula nos projetos em detrimento de um uso maior do conhecimento em seu formato tácito. Em função destas definições, podemos reduzir o uso de artefatos como diagramas e outros tipos de documentos e incrementar a comunicação direta através de conversações entre os envolvidos nos projetos (comum em metodologias ágeis). Desta forma, o fato de priorizar o uso do conhecimento no seu formato tácito, reduzindo o processo de externalização do conhecimento, faz com que as metodologias, técnicas e ferramentas para gestão do conhecimento no processo de desenvolvimento de software sejam repensadas. Diante deste contexto, o objetivo desta pesquisa é apresentar uma metodologia para aquisição do conhecimento presente nas conversações realizadas nas reuniões de projeto, de forma a classificar este conhecimento e indexá-lo através do uso de ontologias.

Palavras chave: gestão do conhecimento, aquisição do conhecimento, desenvolvimento de software, conhecimento tácito.

METHODOLOGY FOR KNOWLEDGE ACQUISITION IN MEETINGS OF SOFTWARE DEVELOPMENT PROJECTS

ABSTRACT

Software engineering is a highly knowledge-based domain, in which success factors are related to the experience of people involved in various phases and process's activities. The knowledge in software engineering is dispersed, of huge proportions and is continually growing. Consequently, the knowledge management in organizations is a wide area with many disciplines that can influence the results. The preference for the use of knowledge in tacit or explicit formats is one of the issues surrounding knowledge management in software development. This preference will determine, among other things, the option for the reducing of the externalization of the knowledge that circulates in projects at the expense of greater use of knowledge in the tacit format, reducing this way the use of artifacts such as diagrams and other documents and increasing the communication through direct talks between those involved in the projects (common and usual in agile methodologies). Thus, prioritizing the use of tacit knowledge and reducing the knowledge externalization process, mandates that methodologies, techniques and tools for knowledge management be reconsidered. In this context, the objective of this research is to present a methodology for knowledge acquisition of the project meeting conversations, in order to classify and index the knowledge through ontologies.

Keywords: knowledge management, knowledge acquisition, software development, tacit knowledge.

LISTA DE FIGURAS

Figura 1 - Progresso na área de reconhecimento automático de fala [Ada10].....	36
Figura 2 - Questões 11 e 17 do questionário	54
Figura 3 - Questões 5 e 7 da entrevista	55
Figura 4 - Fatores que influenciam e são influenciados na GC em projetos ágeis	67
Figura 5 - Mapa conceitual sobre meios de comunicação.....	68
Figura 6 - Mapa conceitual sobre fatores humanos e sociais.....	69
Figura 7 - Mapa conceitual sobre ciclo de desenvolvimento de software e artefatos	71
Figura 8 - Mapa conceitual sobre ferramentas para gestão do conhecimento	73
Figura 9 - Mapa conceitual sobre formas de representação do conhecimento	74
Figura 10 - Etapas do estudo de viabilidade	78
Figura 11 - Análise de similaridade realizada pelo ONTrace	85
Figura 12 - Metodologia proposta	94
Figura 13 - Texto etiquetado.....	97
Figura 14 - Distribuição Normal [Fonte: Wikipedia]	100
Figura 15- Exemplo de classificação das palavras em uma reunião	101
Figura 16 - Desvio padrão comparado ao número de termos selecionados.....	102
Figura 17- Ontologia de reunião	104
Figura 18- Indivíduos da classe "Meeting"	105
Figura 19 - Indivíduos da classe "Tag"	106
Figura 20 - Metodologia proposta utilizando notação BPMN	108
Figura 21 - Ontologia do domínio abordado no estudo de caso	113
Figura 22 - Lista de termos por reunião.....	116
Figura 23 - Resumo dos dados das reuniões.....	118
Figura 24 - Tela do protótipo	128
Figura 25 - Tela protótipo com texto e áudio da reunião.....	129

LISTA DE TABELAS

Tabela 1 - Atividades do método de pesquisa	44
Tabela 2 - Níveis de classificação dos artigos	62
Tabela 3 - Etapas do processo de seleção dos artigos.....	63
Tabela 4 - Síntese dos estudos encontrados	65
Tabela 5 - Classificação das palavras pelo <i>tagger</i>	82
Tabela 6 - Resumo das reuniões	83
Tabela 7 - Resultados das transcrições comparados com a ontologia.....	86
Tabela 8 - Exemplo tabulação de dados do <i>survey</i>	87
Tabela 9 - Termos apontados pelos humanos na reunião 1.....	88
Tabela 10 - Termos relevantes gerados automaticamente na reunião 1.....	89
Tabela 11 - Resumo da análise dos termos detectados nas reuniões.....	90
Tabela 12 - Palavras ordenadas pela frequência	98
Tabela 13 - Palavras ordenadas após aplicar <i>stemmer</i>	99
Tabela 14 - Análise do desvio padrão	102
Tabela 15 - Comparação de termos gerados automaticamente com avaliação humana ...	120
Tabela 16 - Resumo dos dados das reuniões.....	122
Tabela 17 - Análise dos termos por tipos de conceitos da ontologia.....	124
Tabela 18 - Exemplo de correlação entre termos gerados automaticamente e os apontados pelos humanos.....	126
Tabela 19 - Resumo dos coeficientes de correlação entre os termos gerados automaticamente e os apontados por humanos	127

SUMÁRIO

1	<i>Introdução</i>	23
1.1	Organização da proposta de tese	26
2	<i>Referencial Teórico</i>	27
2.1	Gestão do Conhecimento	27
2.2	Gestão do Conhecimento na Engenharia de Software	30
2.3	Aquisição de Conhecimento	33
2.4	Reconhecimento Automático de Fala	35
2.5	Ontologias.....	37
2.6	Considerações sobre o capítulo	41
3	<i>Método de Pesquisa</i>	43
3.1	Etapa 1 - Reconhecimento inicial do problema	45
3.2	Etapa 2 – Estudo em profundidade sobre gestão do conhecimento em projetos de software ágeis	46
3.3	Etapa 3 – Estudo de Viabilidade	47
3.4	Etapa 4 - Proposta Metodológica	48
3.5	Etapa 5 – Validação através de um estudo de caso explanatório.....	48
3.6	Etapa 6 – Refinamento da metodologia proposta e criação de um protótipo para auxiliar na avaliação da metodologia	49
3.7	Considerações sobre o capítulo	50
4	<i>Resultados preliminares</i>	51
4.1	Estudo de caso exploratório	51
4.1.1	Contexto das organizações	51
4.1.2	Instrumentos de Pesquisa	52
4.1.3	Análise dos Dados	53
4.1.4	Discussões e lições aprendidas	55
4.1.5	Considerações sobre o estudo de caso exploratório	58
4.2	Revisão sistemática sobre gestão do conhecimento em projetos de software ágeis ...	59

4.2.1	Planejamento da revisão.....	60
4.2.2	Condução da Revisão	61
4.2.3	Relatório da Revisão.....	66
4.2.4	Considerações sobre a revisão sistemática.....	76
4.3	Estudo de viabilidade	77
4.3.1	Comparando os conceitos do texto original com os da transcrição.....	83
4.3.2	Comparando os conceitos da transcrição com os da ontologia	85
4.3.3	Análise do <i>survey</i> aplicado ao estudo de viabilidade.....	86
4.3.4	Considerações sobre o estudo de viabilidade.....	90
4.4	Considerações sobre o capítulo	91
5	<i>Metodologia Proposta</i>	93
5.1	Etapa 1 - Realização da reunião seguindo o protocolo	95
5.2	Etapa 2 - Transcrição do áudio da reunião.....	96
5.3	Etapa 3 - Etiquetar as palavras da reunião.....	97
5.4	Etapa 4 - Ordenar palavras da reunião por frequência de ocorrência no texto	98
5.5	Etapa 5 - Tratar similaridade das palavras classificadas.....	98
5.6	Etapa 6 - Classificar palavras da reunião de acordo com frequência no texto	99
5.7	Etapa 7 - Associar conceitos extraídos da reunião com ontologia do domínio	103
5.8	Considerações sobre o capítulo	108
6	<i>Resultados do estudo de caso explanatório</i>	111
6.1	Contexto da Organização	111
6.2	Atividades analisadas	112
6.3	Instrumento de pesquisa.....	113
6.4	Análise dos dados	114
6.5	Uso de um protótipo para realizar consultas na ontologia.....	127
6.6	Contribuições deste capítulo	129
7	<i>Considerações Finais</i>	131
	<i>Referências Bibliográficas</i>	135
	<i>Apêndice A - Questionário do Estudo de Caso Exploratório</i>	141
	<i>Apêndice B – Sessão Results da Revisão Sistemática</i>	147

<i>Apêndice C – Formulário do survey aplicado no estudo de viabilidade</i>	<i>159</i>
<i>Apêndice D – Informações dos ensaios realizados no estudo de viabilidade</i>	<i>167</i>
<i>Apêndice E – Formulário do survey aplicado no estudo de caso explanatório</i>	<i>171</i>

1 Introdução

As organizações já perceberam que a Gestão do Conhecimento (GC) ou *Knowledge Management* (KM) apresenta-se como um grande diferencial competitivo, mas gerenciar o conhecimento tem se tornado um grande desafio, visto a complexidade dos processos envolvidos para se obter sucesso na gestão deste recurso. A gestão do conhecimento não é um tema novo no contexto organizacional, mas nos últimos anos tem recebido uma maior atenção com o surgimento de novas tecnologias e metodologias que contribuem para sistematizar e aperfeiçoar algumas de suas fases como a aquisição, a validação, o armazenamento e o compartilhamento do conhecimento.

O sucesso de uma organização está relacionado em grande parte ao conhecimento que esta possui. O conhecimento de uma organização é formado a partir do conhecimento dos seus membros e pode aumentar à medida que este conhecimento é armazenado e compartilhado. Este é um conceito associado às organizações que aprendem [Sen06].

Neste contexto de valorização do conhecimento como um recurso corporativo, surgiu a gestão do conhecimento, que tem o seu foco voltado para questões relacionadas à como as organizações podem tirar maior proveito do conhecimento existente dentro delas. A gestão do conhecimento facilita a distribuição deste conhecimento entre seus membros, encorajando o registro das soluções adotadas para resolução de problemas, evitando a perda do conhecimento de seus especialistas quando estes, por exemplo, deixam a organização [Dav98].

No desenvolvimento de software o panorama é semelhante, pois para as empresas deste seguimento o seu principal recurso é o capital intelectual e não os prédios e as máquinas [Rus02], sendo que a principal sugestão para melhor explorar este recurso é focar no uso da gestão do conhecimento.

A gestão do conhecimento em organizações de desenvolvimento de software é uma área ampla com muitas disciplinas que podem influenciar nos seus resultados. Em Bjørnson e Dingsøyr [Bjø08] é apresentada uma revisão sistemática sobre gestão do conhecimento em engenharia de software, onde são identificados estudos empíricos, discutidos os principais conceitos e as tendências desta área. Nesta revisão sistemática é observado o interesse crescente nos últimos anos sobre o tema e os investimentos realizados pelas empresas e institutos de pesquisa na busca

de soluções que envolvam a gestão do conhecimento no contexto de desenvolvimento de software.

A engenharia de software é um domínio altamente orientado ao conhecimento, no qual os fatores de sucesso estão relacionados com a experiência das pessoas envolvidas nas diversas fases e atividades do processo. O conhecimento na engenharia de software é disperso, de proporção imensa e de crescimento contínuo [Des03].

Dentre as questões que envolvem o uso da gestão do conhecimento na engenharia de software está a questão do uso do conhecimento em seu modo tácito ou explícito. Podemos exemplificar esta situação com o surgimento das metodologias ágeis, que propõem alterações nos processos de desenvolvimento de software, dentre as quais, está a opção pela redução da externalização¹ do conhecimento que circula nos projetos em detrimento de um uso maior do conhecimento em seu formato tácito [Cha03]. Portanto, é reduzido o tempo despendido no processo de conversão do conhecimento tácito para explícito, que é um dos principais desafios na gestão do conhecimento [Lev09]. Ao invés de um uso mais intenso de artefatos como diagramas e outros tipos de documentos, o conhecimento que circula pelo projeto estará preferencialmente na sua forma tácita e isso acabará incrementando, por exemplo, a comunicação direta entre os envolvidos nos projetos.

Essa mudança na forma preferencial em que o conhecimento circula nos projetos de desenvolvimento de software, no caso das metodologias ágeis quando comparadas às metodologias chamadas tradicionais ou “tayloristics”, altera a forma como deve ser concebida a comunicação e a gestão do conhecimento no ciclo de desenvolvimento de software [Mel04]. O fato de priorizar o uso do conhecimento no seu formato tácito, reduzindo o processo de externalização do conhecimento, faz com que as metodologias, técnicas e ferramentas para gestão do conhecimento devam ser repensadas. Por exemplo, o uso da comunicação direta, através das conversações, traz uma série de benefícios à comunicação em um projeto, mas também vários aspectos relacionados às questões humanas e sociais emergem e devem ser considerados para uma maior efetividade nesta forma de compartilhamento e transmissão de conhecimento [Joh05].

¹ Neologismo utilizado em (NONAKA; TAKEUCHI, 1995) que define o processo de articulação do conhecimento tácito em conceitos explícitos.

Uma das fases mais complexas na gestão do conhecimento é a aquisição, pois ela incorpora aspectos organizacionais e individuais onde são considerados os modelos do processo de aprendizado individual, o processo criativo individual, o conhecimento tácito e a intuição [Ter02]. Uma forma de tornar mais efetivo o processo de Aquisição do Conhecimento (AC) é diminuir o esforço dispensado para realização desta tarefa. Quando se trabalha com gestão do conhecimento nas organizações, uma abordagem adequada para a aquisição do conhecimento é tentar ao máximo realizar esta etapa sem um esforço extra além da própria realização da tarefa pelo usuário [Fis01]. Deste modo, evita-se despender um esforço pelo funcionário/colaborador que seja específico para a aquisição e representação do conhecimento, além da realização da própria tarefa, pois isso traz implicações nos processos organizacionais, que podem comprometer a estratégia adotada para a aquisição de conhecimento.

Diante deste contexto, caracterizado em projetos de desenvolvimento de software onde o conhecimento apresenta-se tanto no formato explícito, como no formato tácito, emerge a questão de pesquisa inicial: “como realizar a aquisição do conhecimento presente nas conversações realizadas nas reuniões de projetos de desenvolvimento de software, de forma a classificar este conhecimento e indexá-lo através do uso de ontologias?”.

Convém destacar que a questão de como gerenciar o conhecimento que circula nas conversações dos projetos, não é exclusiva de quando se utiliza metodologias ágeis, mas sim, que essa questão é intensificada quando do uso deste tipo de abordagem para o desenvolvimento de software.

Desta forma, o objetivo desta pesquisa é apresentar uma metodologia para aquisição do conhecimento presente nas conversações realizadas nas reuniões de projeto, de forma a classificar este conhecimento e indexá-lo através do uso de ontologias. Não estão no escopo desta pesquisa e da metodologia apresentada aspectos de construção e criação de conhecimento. Convém destacar que embora muitos ensaios e estudos de caso apresentados nesta pesquisa estão no contexto de projetos que seguem metodologias ágeis, a metodologia resultante apresentada não é direcionada para projetos que seguem metodologias ágeis. Ela pode ser aplicada em qualquer projeto de desenvolvimento de software que cumpra os requisitos apresentados.

Esta pesquisa é relevante devido ao fato de que uma das formas mais representativas do conhecimento em um projeto de software é gerada nas conversações durante as reuniões e este conhecimento acaba não sendo devidamente aproveitado [Bru08]. Durante um projeto, este

conhecimento que circula nas reuniões pode ser transformado para ser utilizado em outras formas de representação do conhecimento, como diagramas, formulários, atas, textos, códigos, etc. Sendo que durante este processo de transformação as perdas e distorções do conhecimento são comuns, além das dificuldades em manter estes artefatos atualizados com as constantes mudanças nos projetos [Cha04]. Este conhecimento que circula nas reuniões pode também não ser transformado em outras formas de representação de conhecimento, acabando sendo somente transmitido entre os participantes sem que haja um mecanismo efetivo de retenção e indexação deste conhecimento para consultas posteriores e relacionamento com outras formas de representação de conhecimento do projeto.

1.1 Organização da proposta de tese

Este documento está dividido da seguinte forma: o Capítulo 2 apresenta a fundamentação teórica necessária para um bom entendimento das definições e conceitos relacionados à gestão do conhecimento no contexto de desenvolvimento de software; o Capítulo 3 descreve o método de pesquisa utilizado ao longo do projeto; o Capítulo 4 apresenta os resultados preliminares; o Capítulo 5 apresenta a metodologia para aquisição de conhecimento em reuniões de projetos de desenvolvimento de software; o Capítulo 6 apresenta o estudo de caso explanatório; e por fim, no Capítulo 7 são apresentadas as considerações finais seguidas das referências bibliográficas.

2 Referencial Teórico

Esta seção apresenta os principais conceitos que são abordados nesta pesquisa, que envolvem a gestão do conhecimento e suas implicações no âmbito organizacional, algumas classificações e metodologias. Em seguida, será focada a gestão do conhecimento no processo de desenvolvimento de software, ressaltando diferenças entre metodologias tradicionais e ágeis e a evolução do interesse da academia e do mercado nesta área. A fase de aquisição do conhecimento será tratada destacadamente, pois o foco desta pesquisa se concentra nesta etapa de uma metodologia de gestão do conhecimento. Também será abordado o reconhecimento automático de fala, que será tratado como uma das formas de automatizar o processo de aquisição do conhecimento dentro da proposta apresentada. E por último, será abordado o tema ontologias, pois o processo de organização do conhecimento que será adquirido através das reuniões de projeto terá sua estratégia de indexação baseada no uso de ontologias.

2.1 Gestão do Conhecimento

Para que as organizações garantam sua permanência no mercado é fundamental que possuam a capacidade de realizar mudanças que aumentem suas vantagens competitivas. No entanto, a realização dessas mudanças nem sempre trazem os resultados esperados devido a pouca compreensão da organização sobre ela mesma, ou seja, a realização de mudanças que almejam benefícios à organização é dificultada pelo escasso conhecimento da organização sobre a forma como os seus processos de negócio são realizados e sobre a sua própria estrutura organizacional. Sendo assim, uma das maneiras das organizações tornarem-se mais competitivas é aumentar o seu conhecimento organizacional [Non95].

O conhecimento organizacional abrange, mas não se restringe, ao conhecimento relacionado aos processos de negócio, conhecimento sobre o relacionamento entre os diversos setores organizacionais, além do conhecimento sobre o mercado, tecnologias, clientes e competidores [Dav98]. Este conhecimento constitui o que é definido como capital intelectual e, portanto, deve ser gerenciado de forma eficiente para garantir a sua preservação e permitir a sua constante evolução e isso pode ser garantido através de uma política de gestão do conhecimento.

Vários autores [Carb05], [Car03], [Dav98], [Non95], [Tur04], [Cho05] apresentam definições sobre a gestão do conhecimento, seus benefícios e desafios. Novas abordagens surgem em forma de metodologias e tecnologias para auxiliarem nos processos da gestão do conhecimento, pois ainda são muitos os desafios para as empresas e especialistas da área. Mesmo com a gestão do

conhecimento não sendo uma área de estudos recente e apesar de todo incremento tecnológico dos últimos anos que contribuíram, direta ou indiretamente, para alavancar a gestão do conhecimento nas organizações, tem-se muitas dificuldades para implantação de projetos bem sucedidos nesta área. Pode-se atribuir a estas dificuldades muitas variáveis, que vão desde aspectos tecnológicos e metodológicos a fatores organizacionais e comportamentais.

“Organizações aprendem somente através de indivíduos que aprendem. Aprendizagem individual não garante aprendizagem organizacional. Mas sem ela a aprendizagem organizacional não ocorre” [Sen04].

Buono and Poulfelt [Buo05] apontam que a gestão do conhecimento está saindo da primeira para a segunda geração. Na primeira geração, o conhecimento era considerado uma possessão, sendo algo que poderia ser capturado. A gestão do conhecimento era baseada em questões técnicas de como capturar e distribuir o conhecimento através de ferramentas como sistemas de gerenciamento de informações, repositórios de dados, etc. A segunda geração é caracterizada pelo *knowing-in-action*. O conhecimento é concebido como um fenômeno social e as soluções devem considerar sistemas humanos complexos, comunidades de práticas, zonas de conhecimento e estruturas de suporte.

Uma forma de compreender a abrangência e estruturação da gestão do conhecimento no panorama organizacional atual é apresentada em Earl [Ear01], que classificou a gestão do conhecimento em “escolas”. As escolas são categorizadas em “Tecnocrática”, “Econômica” e “Comportamental”. As escolas Tecnocráticas são:

- 1) Escola de sistemas: foca na tecnologia para compartilhar conhecimento usando repositórios de conhecimento.
- 2) Escola cartográfica: foca em mapas de conhecimento e criação de diretórios de conhecimento.
- 3) Escola de engenharia: foca em processos e fluxo de conhecimento nas organizações.

As escolas Econômicas focam em como os ativos de conhecimento, ou o capital intelectual, relaciona-se com os resultados financeiros da organização.

A escola Comportamental consiste em três sub-escolas:

- 1) Escola organizacional: foca em redes para compartilhar o conhecimento.

- 2) Escola espacial: foca em como o ambiente de trabalho pode ser projetado para promover o conhecimento.
- 3) Escola estratégica: foca em como o conhecimento pode ser a essência da estratégia nas organizações.

Em Dingsoyr *et al.* [Din09] é apresentada uma relação entre estas escolas e aspectos de desenvolvimento de software, onde, por exemplo, é mencionado que as escolas tecnocráticas estão mais relacionadas a projetos de desenvolvimento de software que seguem metodologias tradicionais. Enquanto que escolas comportamentais estão mais relacionadas às abordagens ágeis.

Quando uma organização pretende aplicar gestão do conhecimento para obter retornos da gerência do seu capital intelectual, isso deve ser feito através da adoção de uma metodologia. Uma metodologia para gestão do conhecimento pode ser apresentada de diversas formas, com diferentes etapas e componentes. Na literatura sobre este tema, encontram-se algumas sugestões que vem sendo aplicadas e servem de referência para trabalhos nesta área.

Para Turban *et al.* [Tur04], uma metodologia para a gestão do conhecimento deve seguir seis passos em um ciclo. Estes passos são:

- criar conhecimento: o conhecimento cria-se à medida que as pessoas descobrem novas formas de fazer as coisas ou que desenvolvem *know-how*;
- capturar conhecimento: é preciso reconhecer o valor do novo conhecimento e representá-lo de forma razoável;
- depurar conhecimento: o novo conhecimento precisa ser colocado dentro do contexto correto para que possa ser acionado;
- armazenar conhecimento: o conhecimento útil deve ser armazenado em formato razoável em um repositório de conhecimento;
- administrar o conhecimento: o conhecimento precisa ser mantido em movimento;
- difundir o conhecimento: o conhecimento precisa ser disponibilizado em formato útil para qualquer pessoa da empresa que dele necessite.

Para Davenport [Dav98] a gestão do conhecimento compõe-se de, pelo menos, três etapas, não necessariamente consecutivas ou ordenadas:

- geração do conhecimento: refere-se principalmente aos processos de aquisição, fusão, adaptação e redes de conhecimento;

- codificação e coordenação do conhecimento: o objetivo da codificação é apresentar o conhecimento num formato que o torne acessível àqueles que precisam dele;
- transferência do conhecimento: discute sobre as estratégias para transferência de conhecimento entre os membros da organização. Ressaltando que a cultura organizacional deve ser favorável ao compartilhamento do conhecimento, afirmando que “na economia regida pelo conhecimento, conversar é trabalhar”.

Os autores Nonaka e Takeuchi [Non95] também sugerem um modelo de cinco fases, focando o processo de criação do conhecimento:

- compartilhamento do conhecimento tácito: o conhecimento tácito de uma organização está nos indivíduos que a compõe. E esse conhecimento individual deve ser compartilhado com a organização;
- criação de conceitos: o conhecimento tácito compartilhado anteriormente deve ser transformado em conhecimento explícito;
- justificação de conceitos: a organização aplica um filtro sobre os conceitos criados, a fim de verificar se “os conceitos recém-criados valem realmente a pena para a organização e para a sociedade”;
- construção de um arquétipo: o conhecimento justificado é transformado em algo tangível ou concreto, ou seja, construir algo com o conhecimento justificado, que pode ser um protótipo;
- difusão interativa do conhecimento: o conhecimento criado, justificado e aplicado se transforma em base para a criação de novos conhecimentos.

As metodologias apresentam variações, mas possuem uma estrutura dorsal baseada na geração, aquisição, depuração, aplicação, armazenamento e compartilhamento do conhecimento. Não estando estas vinculadas a ferramentas ou tecnologias específicas.

2.2 Gestão do Conhecimento na Engenharia de Software

É aceito pela comunidade que é desejável realizar o esforço possível para que todas as formas de conhecimento de engenharia de software sejam capturadas e armazenadas em repositórios [Lan01]. Forma-se um consenso entre empresas que atuam no desenvolvimento de software que a gestão do conhecimento, ou do seu capital intelectual, em nível individual, dos times e da organização é extremamente relevante para a melhoria dos processos e, por conseguinte, constitui-se em um diferencial.

“Um dos maiores problemas do capital intelectual é que ele tem pernas e volta para casa todos os dias. Na mesma proporção que a experiência caminha porta à fora, inexperiência caminha porta à dentro” [Rus02].

Uma organização que atua em desenvolvimento de software tem como seu principal ativo o capital intelectual, que é um reflexo, ou uma síntese, de todas as suas atividades, processos, tarefas e projetos que geram, consomem e transformam conhecimento. Portanto, uma organização de desenvolvimento de software possui os mesmos desafios que empresas de consultoria, investimento, publicidade, advocacia, entre tantos outros ramos de atividade que possuem como seu principal *asset* o conhecimento e precisam fazer dele o seu diferencial no mercado em que atuam.

Gestão do conhecimento nas organizações de desenvolvimento de software é uma área ampla com várias disciplinas que podem influenciar em seus resultados. Em Bjørnson e Dingsøyr [Bj08] é apresentada uma revisão sistemática sobre gestão do conhecimento em engenharia de software. Esta revisão sistemática discute as principais implicações que devem ser ponderadas na decisão sobre qual metodologia de gestão do conhecimento é adequada, considerando, por exemplo, se uma empresa pretende ter seus processos de desenvolvimento baseados em metodologias ágeis ou em metodologias tradicionais.

Quando metodologias tradicionais estão em uso, o objetivo da gestão do conhecimento é tentar o máximo possível transformar o conhecimento tácito em conhecimento explícito, vindo a representar este conhecimento através de diversos artefatos previstos nas metodologias de desenvolvimento de software. Já para as metodologias ágeis, o foco está em trabalhar com o conhecimento tácito. As iniciativas estão focadas em promover a troca mais direta de conhecimento através da conversação entre os membros do projeto. Nas metodologias ágeis uma das principais questões avaliadas para estimular a troca do conhecimento no seu formato tácito é o esforço que se despende para converter o conhecimento tácito em explícito, a chamada “explicitação do conhecimento”.

Na engenharia de software temos algumas definições para as chamadas “organizações de software que aprendem”. Feldmann e Althoff [Fel01], definem uma organização de software que aprende, aquela que cria uma cultura que promove aprendizagem contínua e promove a troca de experiências. Dyba [Dyb02] enfatiza este tipo de empresa como aquela que promove ações de melhoria através de um melhor conhecimento e entendimento dos seus processos.

Temos muitos estudos de casos e iniciativas de gestão do conhecimento na área de engenharia de software [Rus02]. Como exemplo, podemos apontar uma que está fundamentada no reuso da experiência do ciclo de vida, processos e produtos para o desenvolvimento de software, que é a *Experience Factory* [Bas94]. Esta proposta está baseada no fato de que projetos de desenvolvimento de software podem melhorar sua *performance* (custos, qualidade e cronograma) através da utilização das experiências de projetos anteriores. A *Experience Factory* deve analisar e sintetizar todos os tipos de experiência, incluindo lições aprendidas, dados de projetos, relatório de tecnologia, entre outros e, fornecer serviços de repositório para estas experiências.

A *Experience Factory* emprega alguns métodos para empacotar experiência, incluído aí medições de projeto de vários processos de software e características de produtos e, então, constrói modelos destas características que descrevem seu comportamento em diferentes contextos. Os dados destes modelos são provenientes de projetos de desenvolvimento obtidos de pessoas, documentos e suporte automatizado.

Podemos observar o interesse e crescimento de iniciativas de gestão do conhecimento em engenharia de software pelo incremento de publicações nesta área, conforme destacado na revisão sistemática de [Bjø08]. Neste estudo foram identificados 68 trabalhos como “lessons learned reports” ou “empirical studies”. Os trabalhos foram publicados entre 1992 e 2007, sendo que é perceptível o incremento e a constância de publicações a partir do ano de 2000. Entre 1992 e 2000 foram 12 publicações e o restante, 66, a partir do ano 2000.

Nesta mesma revisão é colocado que no ano de 1999 ocorreu o “*first workshop on learning software organizations*”, organizado em conjunto com a conferência SEKE². Este *workshop* apresenta-se como um dos principais fóruns para estudos empíricos, bem como desenvolvimento tecnológico, relacionado à *knowledge management in software engineering*.

Em maio de 2002 a IEEE Software [Rus02] tratou especificamente do tema *knowledge management in software engineering*, trazendo muitos exemplos de aplicações da gestão do conhecimento em empresas que desenvolvem software. Empresas como Chrysler, Infosys, Nasa, Departamento de Defesa dos EUA (DoD), entre outros, apresentaram relatos de experiências do uso de gestão do conhecimento no processo de desenvolvimento de software.

² <http://www.informatik.uni-trier.de/~ley/db/conf/seke/index.html>

O livro *Managing Software Engineering Knowledge* publicado em 2003 [Aur03] focou em tópicos que tentaram identificar o motivo pelo qual gestão do conhecimento é importante para engenharia de software.

Em [Rus02] são apresentados *overviews* de trabalhos em gerenciamento do conhecimento em engenharia de software. Neste trabalho são tratados aspectos sobre motivações e abordagens para gerenciar o conhecimento e fatores que devem ser considerados quando estratégias para gerenciar o conhecimento em empresas de software são implantadas. Estes aspectos envolvem questões tecnológicas, organizacionais e de pessoas. Neste artigo os autores também descrevem os tipos de ferramentas que são relevantes para gerenciar conhecimento, incluindo ferramentas para gerenciar documentos, competências e colaboração.

Em [Din02] os autores utilizaram um *survey* para estudar iniciativas de gestão do conhecimento em engenharia de software. Foram apontados oito relatórios no formato de lições aprendidas, os quais tratam como as companhias realizaram suas ações nesta área, quais os efeitos sofridos, ações realizadas, benefícios reportados e que tipos de estratégias para gerenciar o conhecimento foram empregadas.

No ano de 2009 temos o lançamento de dois importantes livros nesta área. Schneider [Sch09] publicou um livro que trata de aspectos sobre experiência e gestão do conhecimento na engenharia de software, abordando: que tipo de conhecimento é importante para engenharia de software; o reuso de experiências e conhecimento; as representações formais e estruturas como ontologias e; estudos de casos. Babar *et al.* [Bab09] organizou um livro que fornece uma visão geral dos principais conceitos de arquitetura de software e gestão do conhecimento, abordando tópicos como ferramentas e tecnologias para gestão do conhecimento de arquitetura de software onde também são apresentadas experiências observadas em estudos de casos.

2.3 Aquisição de Conhecimento

Considerando que a fase de Aquisição do Conhecimento (AC) é uma das mais complexas e importantes em um ciclo de gestão de conhecimento, é de suma importância que ao elaborar-se uma estratégia para aquisição do conhecimento, que esta seja bem definida e apropriada à realidade da organização, considerando aspectos tecnológicos, organizacionais e pessoais. Rubin e Dai [Rub04] destacam a importância desta definição, pois eles consideram a fase de aquisição um “gargalo” dentro de um ciclo de gestão do conhecimento.

O processo de gestão do conhecimento em uma empresa despende muitos recursos (pessoas, tempo, investimento financeiro, etc.), tornando-se normalmente uma atividade custosa. Portanto, é indicado que a fase que contempla o processo de aquisição de conhecimento deva ser proposta e implantada considerando que esta venha a ter um impacto mínimo nos processos organizacionais [Kom02].

Existem diversas abordagens de aquisição de conhecimento que podem ser definidas, devido à diversidade dos tipos de conhecimento que uma organização pode considerar como úteis e de acordo com os recursos tecnológicos e humanos que dispõem [Mon03].

Para a efetivação do processo de aquisição de conhecimento, deve-se considerar qual a infraestrutura disponível que permitirá a aplicação de técnicas variadas de aquisição e a representação do conhecimento em diversos formatos, assim como possibilitará a aquisição de conhecimento apoiada por recursos computacionais [Mil99].

As abordagens de aquisição podem ser definidas segundo a estrutura do conhecimento. Estas abordagens podem ser de duas formas: manual ou automática. O conhecimento tácito, complexo e de valor, como melhores práticas, lições aprendidas ou conhecimento do domínio é geralmente adquirido manualmente. A infraestrutura de aquisição de conhecimento (técnicas, ferramentas, modelos e especialistas) deve, então, prover mecanismos que facilitem a captura manual do conhecimento [Mon03]. O conhecimento explícito e menos complexo pode ser adquirido através de componentes de aquisição automática de conhecimento [Abe99], [Rus02].

Existem muitos conhecimentos de difícil captura no contexto organizacional, e o conhecimento sobre o domínio do negócio é um exemplo. Técnicas de aquisição manuais como entrevistas estruturadas são bastante adequadas para capturar esse tipo de conhecimento [Mil99].

O conhecimento tácito de experiências pessoais de membros da organização também constitui um recurso de conhecimento muito importante e de difícil extração, pois muitas vezes os membros da organização têm dificuldade de expressar suas idéias e seu conhecimento de forma adequada. Este problema pode ser minimizado através de questionários que permitam adquirir de forma uniforme e estruturada o conhecimento absorvido por um membro da organização ao longo de vários anos de trabalho. Também podem ser utilizadas técnicas como mapas conceituais para realizar este tipo de aquisição do conhecimento [Cañ06].

Abecker *et al.* [Abe98] sugerem a utilização de componentes de aquisição automática de conhecimento para analisar fontes de conhecimento internas ou externas à organização e alimentar as bases de conhecimento da memória organizacional. Esta abordagem de aquisição de conhecimento permite que as informações distribuídas em diversas fontes, como as bases da organização, sejam analisadas e relacionadas automaticamente sem interação humana. É claro que esta abordagem está associada ao uso de conhecimento explícito e estruturado, pois é difícil a tarefa de realizar a aquisição automática de conhecimento tácito, de forma que o mesmo possa ser depurado posteriormente.

Para a fase de aquisição do conhecimento nem sempre é necessário o uso de uma ferramenta específica. Pode-se, por exemplo, fazer a aquisição do conhecimento em uma organização através dos documentos, modelos, formulários e diagramas gerados durante a execução das tarefas. Entrevistas formais e informais podem ser usadas como forma de captura do conhecimento das pessoas da organização. Pode-se utilizar tecnologias e ferramentas já implantadas na empresa para outros propósitos, mas que podem acabar sendo aproveitadas para aquisição do conhecimento, como intranets, data warehouse, ferramentas de groupware, FAQs, etc.

Um aspecto importante na adoção de uma ferramenta para aquisição do conhecimento, assim como para qualquer outra fase na gestão do conhecimento, é que esta esteja vinculada a uma política organizacional e contemplada em uma metodologia abrangente para todo o processo de gestão do conhecimento [Lie02].

2.4 Reconhecimento Automático de Fala

O uso de aplicações para o reconhecimento automático de fala pode ser um aspecto facilitador no processo de aquisição de conhecimento, já que um dos requisitos-chaves a ser considerado durante o processo de aquisição é o mínimo de intervenção na atividade realizada de onde se pretende extrair o conhecimento [Kom02]. Considerando que em muitas situações o conhecimento que circula nos projetos é exposto em reuniões e conversas entre membros das equipes e clientes, facilitaria para o processo de aquisição o uso de um mecanismo que converta o conhecimento gerado nestas reuniões para um formato mais apropriado de ser indexado e relacionado com outras informações do projeto.

Podemos observar o avanço da área de reconhecimento automático de fala (ASR - *Automatic Speech Recognition*) nos últimos anos devido ao incremento de tecnologias e

ferramentas relacionadas a esta área. No estudo realizado por Adami [Ada10] é apresentado um panorama da evolução das técnicas, métodos e tecnologias aplicadas ao reconhecimento automático de fala. Na Figura 1 é ilustrado um gráfico de evolução de aspectos que compõem o reconhecimento automático de fala. Neste gráfico os dois eixos apresentam variáveis que influenciam diretamente no desempenho dos sistemas de reconhecimento de fala, que são o estilo da fala (*speaking style*) e o tamanho do vocabulário (*vocabulary size*). Um panorama considerado simples contemplaria palavras isoladas e um vocabulário pequeno e restrito. Enquanto que um panorama complexo tenderia a utilizar um estilo de fala espontâneo e um vocabulário com um grande número de palavras tendendo a irrestrito. As três linhas plotadas no gráfico fornecem uma idéia da evolução de tipos de aplicações empregadas no decorrer das últimas décadas.

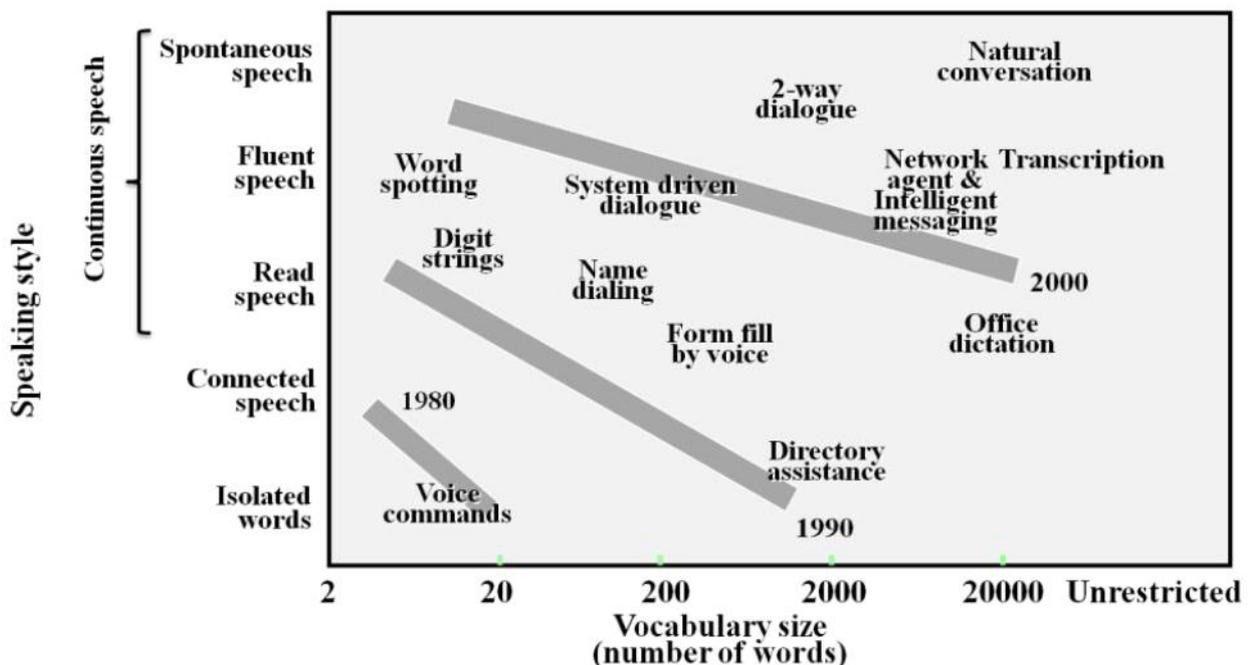


Figura 1 - Progresso na área de reconhecimento automático de fala [Ada10]

Temos como um exemplo que demonstra o investimento que esta área vem sofrendo, o projeto CALO - *Cognitive Assistant that Learns and Organizes* [Tur08], desenvolvido pelo SRI (Stanford Research Institute) que possui projetos de *speech recognition* direcionados para reuniões. O CALO é um projeto patrocinado pela DARPA (Defense Advanced Research Projects Agency) e desenvolvido em parceria com: CMU (Carnegie Mellon University), GA Tech (Georgia Institute of Technology), MIT (Massachusetts Institute of Technology), OHSU (Oregon Health & Science University), Stanford CSLI (Center for the Study of Language and Information) e University of Washington. Este é um projeto desenvolvido juntamente com outras iniciativas do SRI para

reconhecimento automático de fala em reuniões, que objetiva proporcionar não só a transcrição das conversações compostas por diversas pessoas, mas fornecer um ambiente composto por uma série de ferramentas com recursos para dar suporte aos participantes com *feedbacks* de assuntos relacionados aos tópicos que surgem na reunião [Ver09].

Nos estudos observados, uma das maiores dificuldades encontradas para a transcrição das conversações das reuniões é a dificuldade em ter como fonte um áudio composto por diversos padrões de fala diferentes, ruídos, interrupções, transposição de falas e diálogos, entre outros [Mag07], [Fri09], [Ada10].

Uma série de aplicações para reconhecimento automático de fala já estão disponíveis. Algumas destas ferramentas são gratuitas como o Sphinx³, que é mantida pela CMU - *Carnegie Mellon University*, sendo uma das mais conhecidas no meio acadêmico. As soluções da empresa Nuance⁴, como o *Dragon Naturally Speaking*, estão entre as mais conhecidas no mercado e possuem várias versões, inclusive para áreas específicas como saúde e direito.

Muitos estudos relacionados à área de *speech recognition* preocupam-se também com questões acústicas e tratamento de áudio [Mag07]. Outros estudos envolvem muitos aspectos das estruturas linguísticas e comportamentais dos participantes das conversações [Jay09], [Mag07a], [Rot07]. Outro aspecto que tem recebido atenção da comunidade são as técnicas associadas à *Diarization* que identificam em uma conferência quem falou o quê e quando [Sto10], distinguindo o que é fala e o que não é fala. Muitas vezes estas técnicas empregam simultaneamente recursos de áudio e vídeo.

Percebe-se então que o reconhecimento automático de fala, mesmo com restrições de desempenho em determinados contextos, apresenta-se com uma técnica adequada para auxiliar no processo de aquisição de conhecimento oriundo de conversações realizadas em reuniões [Tur10].

2.5 Ontologias

O termo ontologia é derivado da filosofia, onde seu significado está associado à natureza do ser, da realidade, da existência dos entes e das questões metafísicas em geral. Na visão de Aristóteles significa “uma explicação sistemática da existência”, que pode ser interpretado como a definição de um domínio do conhecimento em um nível genérico, utilizada para especificar o que

³ <http://cmusphinx.sourceforge.net>

⁴ <http://www.nuance.com>

existe ou o que se pode dizer sobre o mundo. Neste contexto, filósofos tentam responder as questões: “O que é um ser? e quais são as características comuns de todos os seres?”

Para as áreas da Ciência da Computação e da Ciência da Informação, uma ontologia é um modelo de dados que representa um conjunto de conceitos dentro de um domínio e os relacionamentos entre estes. Uma ontologia é utilizada para realizar inferência sobre os objetos do domínio. Uma ontologia também representa a aquisição do conhecimento a partir de dados semiestruturados utilizando um conjunto de métodos, técnicas ou processos automáticos ou semiautomáticos [Won08].

O termo foi recentemente adotado pelas comunidades de Inteligência Artificial e gestão de conhecimento para se referir a conceitos e termos usados para descrever alguma área do conhecimento ou construir uma representação deste. Em Inteligência Artificial pode-se assumir uma interpretação para ontologia como sendo um conjunto de entidades com suas relações, restrições, axiomas e vocabulário. Uma ontologia define um domínio, ou, mais formalmente, especifica uma “conceituação” acerca dele [Gru93].

Ontologias ajudam a formalizar o conhecimento dividido por um grupo de pessoas quando o conhecimento precisa ser modelado, estruturado e relacionado e abrem o caminho para substituir a visão orientada a documentos por uma visão orientada a conteúdo na qual itens de conhecimento são relacionados, combinados e utilizados [Sta01]. Desta forma, o compartilhamento efetivo de conhecimento pode ser alcançado através de acesso a múltiplas bases de conhecimento utilizando ontologias que permitam usuários definir os recursos que eles necessitam e requerem [O’le98].

Ontologias descrevem explicitamente modelos conceituais de um domínio e, portanto, são úteis na construção de memórias organizacionais, pois permitem definir as estruturas e os relacionamentos de itens de conhecimento armazenados nas bases de conhecimento, além de definir as características e visões dessas bases e prover modelos que ajudam na definição e acesso a elas [O’le98].

Ontologias ajudam a indexar a memória organizacional para permitir pesquisas posteriores e recuperação de conhecimento na memória corporativa materializada em documentos ou outros arquivos e facilitam também a identificação de comunidades de prática. Em uma comunidade de prática um grupo distribuído de pessoas pode compartilhar os mesmos interesses em uma tarefa, problema, ou prática. Desta forma, ontologias possibilitam relacionar membros de uma

organização ou várias organizações segundo conhecimento específico que cada um possui ou tem interesse [Ala03].

A formalização do conhecimento através de ontologias facilita a comunicação entre especialistas de um domínio e evita falhas no processo de aquisição e transferência do conhecimento, pois colocam restrições na estrutura e no conteúdo do conhecimento do domínio. Além do mais, a formalização e restrições do conhecimento do domínio impostas pelas ontologias facilitam a integração de múltiplas bases de conhecimento e minimizam o risco de ambiguidades entre elas [O'le98].

Para se definir e manipular ontologias se sugere a utilização de linguagens que suportem estruturas para representação do conhecimento. Esta representação é realizada através da descrição formal de um conjunto de termos sobre um domínio específico. A definição de uma linguagem é necessária para a representação e descrição formal da estrutura que especifica uma conceituação.

Apontamos como exemplo de uma linguagem a OWL (*Web Ontology Language*), que é recomendada pela W3C⁵ como linguagem para manipulação de ontologias e seu diferencial é a capacidade de processamento semântico através de inferência. A linguagem OWL é construída sobre RDF e RDF *Schema* e baseada na sintaxe XML. O modelo básico de dados do RDF, e herdado por OWL, é definido através de [Noy00], [NoI09]:

- recurso: qualquer entidade referenciada através de um URI (*Universal Resource Identifier*);
- propriedade: representam recursos, características que representam recursos ou relacionamento entre recursos;
- declaração: corresponde a uma propriedade ou valor dessa propriedade associada a um recurso específico. Uma declaração é dividida em três partes: sujeito (recurso), predicado (propriedade do recurso) e objeto (valor da propriedade).

Para se estruturar um documento OWL, define-se em alto nível:

- classes: conjunto de instâncias com características comuns, podendo ser consideradas superclasses, relacionamentos e disjunções;

⁵ <http://www.w3.org/standards/semanticweb/>

- propriedades: representam tipos (*datatype Properties*), que identificam os valores primitivos das instâncias, como *integer, float, string, boolean, etc.*;
- objetos (*object Properties*), que representam o relacionamento de duas instâncias; inversa (*inverseOf*), que representam um relacionamento bidirecional; e transitivas (*Transitive Property*), usado principalmente em relações de transitividade do tipo “parte de”;
- indivíduos: representam os objetos em um domínio, isto é, instâncias específicas. Verifica-se aqui que dois nomes podem representar o mesmo objeto no mundo real.

As ontologias possuem várias classificações e aplicações, que diferem quanto à função, ao grau de formalismo, à estrutura e ao conteúdo. Para o uso no contexto de desenvolvimento de software, também encontramos diversificadas aplicações, onde podemos destacar o estudo de Happel e Seedorf [Hap01]. Estes autores apontam os diversos usos de ontologias na engenharia de software durante o ciclo de vida de desenvolvimento, onde destacam seu uso para as fases de:

- Análise e Projeto: ontologias podem beneficiar a área de engenharia de requisitos em termos de representação do conhecimento e suporte ao processo. Durante o projeto também podem auxiliar no reuso de componentes.
- Implementação: durante o ciclo de desenvolvimento, a transição da análise e projeto para a implementação é uma etapa crítica e depende da maneira como o domínio do problema é mapeado. Para esta etapa sugere-se a modelagem de software com auxílio de ontologia. As ontologias também podem fornecer suporte para codificação e documentação do código.
- Manutenção: ontologias podem ser usadas para unificar as diversas fontes de informação sobre os sistemas e evitar trabalho redundante. Também são sugeridos *frameworks* específicos para auxiliar na atualização de versões dos sistemas e para a fase de testes.

Para o contexto desta tese serão utilizadas ontologias de domínio, que descrevem os conceitos e relacionamentos de um determinado domínio de conhecimento. Levando em consideração a classificação de Happel e Seedorf [Hap01] uma ontologia do domínio poderia ser utilizada em todo o ciclo de desenvolvimento de um software.

2.6 Considerações sobre o capítulo

Considerando a questão de pesquisa deste projeto, que envolve a aquisição do conhecimento presente nas conversações realizadas em reuniões de projeto, as seções anteriores apresentaram uma revisão da literatura, buscando contextualizar dentro deste foco de pesquisa, definições sedimentadas sobre os tópicos que envolvem gestão do conhecimento e engenharia de software, aquisição do conhecimento, reconhecimento automático de fala e ontologias.

A revisão na literatura apresentou a importância da gestão do conhecimento como uma das formas de uma empresa obter um diferencial competitivo, esta visão também foi estendida para o contexto de desenvolvimento de software, onde foi percebida uma divisão entre o uso preferencial do conhecimento tácito (metodologias ágeis) e o uso preferencial do conhecimento explícito (metodologias tradicionais). Um aspecto chave na gestão do conhecimento, considerando esta dicotomia (tácito/explicito) é a fase de aquisição do conhecimento. Durante a revisão sobre a fase de aquisição, percebeu-se a importância de aplicações para reconhecimento automático de fala, como uma opção para auxiliar a automatização da aquisição do conhecimento na forma tácita. Também foram realizados estudos sobre ontologias, pois se apresentou como uma alternativa para a indexação deste conhecimento a ser adquirido das conversações.

Este capítulo destacou ainda questões em aberto que necessitam de maior investimento para uma maior efetividade das soluções de gestão do conhecimento no contexto de desenvolvimento de software, considerando aspectos como, por exemplo, o uso preferencial do conhecimento em sua forma tácita, que é mais comum em metodologias ágeis.

3 Método de Pesquisa

Este capítulo apresenta o método de pesquisa empregado neste projeto, sendo que este faz uso de estudos de caso, estudos de viabilidade, revisão sistemática e a definição de uma metodologia a ser utilizada no contexto de desenvolvimento de software. No decorrer das próximas seções será apresentado o método utilizado e a forma como este foi estruturado.

Em Oates [Oat06] é apresentado um sumário de diferentes tipos de produtos derivados de pesquisas com contribuições para o conhecimento que podem ser: evidência, metodologia, análise, conceitos e teorias e, produto.

Esta tese apresenta uma metodologia como contribuição principal, que segundo Oates [Oat06], “também pode ser chamado de Método e é um guia de orientações sobre os modelos a serem produzidos e etapas do processo a serem seguidas para resolver problemas usando Tecnologias da Informação”.

Como usaremos estudos de caso nesta pesquisa, a seguir temos algumas definições para estes tipos de estudos de caso, segundo Oates [Oat06]:

- estudo de caso exploratório: é usado para definir as questões ou hipóteses a serem usadas no decorrer do estudo. É usado para ajudar o pesquisador a entender o problema de pesquisa;
- estudo de caso explanatório: além de analisar detalhadamente um fenômeno particular em seu contexto, tenta explicar porque ele aconteceu e os seus resultados. Procura identificar os múltiplos e inter-relacionados fatores que influenciam nos resultados. Também pode comparar os resultados encontrados com teorias na literatura.

Como também foi utilizada uma revisão sistemática na metodologia, cabe destacar as definições, e um guia para realizações de revisões sistemáticas, que são encontradas em Kitchenham [Kit04]. Em Dyba e Dingsøyr [Dyb08] é apresentado um estudo sobre a realização de revisões sistemáticas em engenharia de software e é posto que uma revisão sistemática “é um resumo conciso das melhores evidências disponíveis, que utiliza métodos explícitos e rigorosos para identificar, avaliar criticamente e sintetizar os estudos relevantes sobre um determinado tema. Estes métodos são definidos com antecedência e documentados em um protocolo para que outros possam apreciar criticamente e replicar a revisão”.

Para o desenvolvimento da metodologia proposta foram seguidas seis grandes etapas que constituem esta pesquisa. Para uma melhor compreensão das atividades que compõem estas seis etapas, a Tabela 1 apresenta uma descrição das principais atividades propostas para cada etapa.

Tabela 1 - Atividades do método de pesquisa

Método de Pesquisa						
Atividades	Etapa 1	Etapa 2	Etapa 3	Etapa 4	Etapa 5	Etapa 6
Estudos abrangentes sobre aquisição e gestão do conhecimento						
Estudos em gestão do conhecimento no desenvolvimento de software						
Realização do estudo de caso exploratório em projeto que aplica métodos ágeis						
<i>Survey</i> e análise dos resultados do estudo de caso exploratório						
Definição do foco e protocolo para a Revisão Sistemática						
Revisão Sistemática sobre gestão do conhecimento em projetos de software ágeis						
Análise dos resultados da Revisão Sistemática						
Elaboração e execução de um estudo de viabilidade						
<i>Survey</i> para avaliação dos resultados do estudo de viabilidade						
Definição de uma metodologia para aquisição do conhecimento em reuniões de projetos de desenvolvimento de software						
Elaboração e organização de um estudo de caso explanatório						
Análise dos resultados do estudo de caso explanatório						
<i>Survey</i> para avaliação dos resultados do estudo de caso explanatório						
Refinamento da metodologia e elaboração de um protótipo para auxiliar na avaliação da metodologia.						
	Etapa 1	Etapa 2	Etapa 3	Etapa 4	Etapa 5	Etapa 6

Como pode ser visto na Tabela 1, as etapas da metodologia são sequenciais e serão detalhadas a seguir:

3.1 Etapa 1 - Reconhecimento inicial do problema

Essa etapa foi eminentemente exploratória. Nela foram estudados conceitos e aspectos relevantes à formulação da metodologia proposta, entre os quais, conceitos de aquisição e gestão do conhecimento. A área de gestão do conhecimento é ampla e multidisciplinar, sendo que focamos em aspectos mais relacionados à sua influência no desenvolvimento de software. Também foram realizados estudos mais aprofundados em fases específicas como a etapa de aquisição de conhecimento, que está no foco da nossa pesquisa. Devido à percepção inicial do contexto mais adequado para aplicação da metodologia, foram realizados estudos sobre aspectos da gestão do conhecimento em metodologias ágeis para o desenvolvimento de software. Convém destacar que embora a percepção inicial fosse da aplicação da metodologia proposta em ambientes de desenvolvimento de projetos de software que seguissem metodologias ágeis, ao final da pesquisa, apontamos que a metodologia proposta não se restringe e não está focada somente em projetos ágeis.

Esta etapa foi finalizada com a realização de um estudo de caso exploratório realizado em uma grande empresa multinacional fabricante de computadores. Esta empresa contratou outra empresa para desenvolvimento de uma aplicação, sendo que as equipes (da contratante e da contratada) realizaram o projeto seguindo uma metodologia híbrida (cascata e *Scrum*). Foi aplicado um *survey* e realizadas entrevistas com os membros das equipes de desenvolvimento e o cliente. Os detalhes do contexto onde foi aplicado o estudo de caso, os instrumentos de pesquisa, a análise dos dados e os resultados foram descritos em um artigo apresentado na 11th International Conference – ICEIS 2009 [Yan09]. Entre as lições aprendidas apresentadas neste artigo, podemos destacar de forma genérica os problemas relacionados à comunicação entre os membros da equipe e entre estes e o cliente; os problemas nos repositórios de artefatos do projeto; o uso consciente de documentação desatualizada pelos membros da equipe; o intenso uso de comunicação informal em questões relacionadas ao projeto e; uma diferença do ponto de vista da gestão do conhecimento quando é empregada uma metodologia ágil de projeto, pois o conhecimento passa a ser utilizado preferencialmente no formato tácito. O estudo dos conceitos envolvidos na pesquisa e o estudo de caso confirmaram a necessidade de uma revisão aprofundada sobre o tema. As conclusões detalhadas do estudo de caso exploratório são apresentadas na seção 4.1 Estudo de caso explanatório.

3.2 Etapa 2 – Estudo em profundidade sobre gestão do conhecimento em projetos de software ágeis

Percebeu-se na Etapa 1 que a gestão do conhecimento quando aplicada a projetos de software que seguem as chamadas metodologias tradicionais priorizam o uso do conhecimento em seu formato explícito, enquanto que, quando segue-se metodologias ágeis, o uso do conhecimento é priorizado em seu formato tácito, com um incremento das conversações através da comunicação direta entre os participantes do projeto.

Com o objetivo de identificar, avaliar e interpretar a literatura relevante sobre estes temas decidiu-se pela realização de uma revisão sistemática [Kit04], sobre gestão do conhecimento em desenvolvimento de software ágil. Esta revisão objetivou responder, entre outras questões, que influências são percebidas na gestão do conhecimento quando o conhecimento que circula no projeto é trabalhado preferencialmente em seu formato tácito e quando o processo de externalização do conhecimento é preterido pelo incremento da comunicação direta entre os membros da equipe.

A revisão sistemática objetivou fornecer uma visão geral de estudo dentro da gestão do conhecimento em projetos de software ágeis, que tipos de conceitos têm sido explorados, quais são os principais resultados encontrados e os métodos de pesquisa utilizados. Mais especificamente nós procuramos responder as seguintes questões de pesquisa:

- Quais são os principais conceitos em gestão do conhecimento que têm sido investigados em projetos de software ágeis?
- Quais são os principais resultados encontrados em pesquisas sobre gestão do conhecimento em projetos de software ágeis?

Ao responder estas duas questões também verificamos a existência de uma área que surge da intersecção da gestão do conhecimento e de projetos de software ágeis e se há contribuição suficiente para caracterizá-la como uma área de pesquisa.

As contribuições da revisão sistemática são apresentadas através de uma síntese dos resultados dos trabalhos relevantes encontrados, das discussões sobre estes e as conclusões do trabalho. O planejamento, o desenvolvimento e as conclusões detalhadas da revisão sistemática são apresentadas na seção 4.2 Revisão sistemática sobre gestão do conhecimento em projetos de software ágeis.

Diante dos resultados obtidos nas etapas 1 e 2, foi elaborado um estudo de viabilidade para se verificar a possibilidade de utilização do conhecimento gerado e compartilhado nas reuniões de projeto como uma forma de representação do conhecimento, que durante o seu processo de aquisição possa ser classificado e indexado, como será mostrado na etapa a seguir.

3.3 Etapa 3 – Estudo de Viabilidade

Nesta etapa foram realizados vários ensaios para se verificar a viabilidade de aproveitar os diálogos de uma reunião como forma de conhecimento que possa ser indexado e relacionado com a ontologia do domínio de um projeto de desenvolvimento de software. A proposta para este estudo de viabilidade, de forma genérica, foi realizar reuniões seguindo um roteiro baseado no estudo de caso exploratório e gravá-las em um ambiente acusticamente adequado e com equipamentos apropriados. Para cada reunião foi gerado um arquivo de áudio, que foi submetido a um software de reconhecimento de fala, que gerou a transcrição textual das conversações. Estes arquivos com as transcrições foram “etiquetados”, por um aplicativo “tagger”, de forma a extrair alguns tipos de palavras, como os substantivos, que serviram para representar os conceitos associados à reunião. Estes conceitos foram classificados e comparados à ontologia do domínio. Por último foi realizado um *survey* para avaliar a “qualidade” destes conceitos gerados.

Foram realizados ensaios para verificarmos a viabilidade dos seguintes tópicos:

- utilização de roteiros de condução das reuniões de forma a facilitar a extração e identificação dos conceitos relevantes;
- requisitos das condições acústicas para a geração e gravação do áudio nas reuniões de forma a atender os requisitos mínimos dos softwares utilizados para reconhecimento automático de fala;
- desempenho do software de reconhecimento automático de fala ao analisar o áudio gerado das reuniões gravadas para o ensaio;
- uso de uma aplicação do tipo “tagger” para identificação do tipo de palavras que podem ser utilizadas como conceitos relevantes nas reuniões;
- associação dos conceitos extraídos automaticamente das reuniões com os conceitos da ontologia do domínio.

Após a realização dos ensaios acima, foi realizado um *survey* para confrontar os conceitos que foram extraídos das reuniões automaticamente e que poderiam ser utilizados como indexadores destas reuniões, com os conceitos que humanos apontariam como indexadores para

estas mesmas reuniões. Os resultados e detalhes da execução dos ensaios realizados neste estudo de viabilidade estão descritos na seção 4.3 Estudo de caso explanatório.

3.4 Etapa 4 - Proposta Metodológica

O objetivo desta quarta etapa é a proposição de uma metodologia que possibilite a aquisição de conhecimento das reuniões de projetos de desenvolvimento de software, de forma a indexar este conhecimento através da ontologia do domínio, possibilitando, por exemplo, recuperar reuniões que abordaram determinado tema durante o projeto de desenvolvimento de software; listar conceitos associados a uma determinada reunião do projeto e; associar conceitos das reuniões a conceitos de outros artefatos do projeto.

Os detalhes de como foi concebida e elaborada a proposta metodológica, assim como suas etapas e fases são apresentadas no Capítulo 5 – Metodologia Proposta.

3.5 Etapa 5 – Validação através de um estudo de caso explanatório

Um estudo de caso foca em uma instância do que está sendo investigado: uma organização, um departamento, um sistema de informação, um fórum de discussão, um desenvolvedor de sistemas, um projeto de desenvolvimento, etc. Esta instância deve ser estudada em profundidade, usando uma variedade de métodos de geração de dados (entrevistas, observação, análise da documentação, questionários, etc.) e o objetivo é obter uma visão rica e detalhada sobre a vida desse caso, suas relações e os seus processos complexos. Além da definição anterior, Oates [Oat06] também apresenta alguns pontos que devem ser observados na seleção de um estudo de caso. Considerando que um estudo de caso é uma instância da “coisa” a ser investigada, é importante a escolha correta da instância e esta pode ser baseada em:

- instância típica: o caso escolhido é típico e pode por isso ser dado como representante de toda uma classe. Resultados deste caso podem ser generalizados para toda esta classe;
- instância extrema: o caso não é típico, mas fornece um contraste com a situação normal;
- *test-bed* para teoria: o caso contém elementos que podem ser utilizados para testar uma teoria existente;
- conveniência: pessoas do caso escolhido concordaram em fornecer acesso e é conveniente em termos de tempo e recursos. É claro que está não deve ser a principal razão para escolher este caso;

- oportunidade única: quando surge a chance de estudar algo que não se tinha planejado anteriormente e isso pode ser que não ocorra novamente. Talvez se esteja no lugar certo e na hora certa, ou aconteceu de se falar com a pessoa certa. Bons pesquisadores atentam para este tipo de oportunidade.

O nosso estudo de caso explanatório foi classificado como “conveniência” pelas definições anteriores, mas um fator decisivo para escolha da instância foi atender os nossos requisitos elencados de forma a aplicar todas as etapas da metodologia proposta. Os requisitos especificados para a realização do estudo de caso foram:

- as reuniões deveriam ser realizadas na língua inglesa, ou serem traduzidas e regravadas neste idioma, e deveriam estar em conformidade quanto aos aspectos tecnológicos e instrumentais (acústica, microfones, etc.);
- o contexto deve ser desenvolvimento de software, com um processo definido (ágil, tradicional ou híbrido);
- no processo devem constar reuniões periódicas da equipe de desenvolvimento.

O principal aspecto avaliado em nossa proposta através deste estudo de caso diz respeito à “qualidade” ou “relevância” dos conceitos extraídos das conversações e apontados como indexadores das reuniões. O objetivo principal desta avaliação foi verificar os resultados gerados com a aplicação da metodologia proposta, quando comparados com os resultados de um processo realizado de forma manual pelas pessoas que participaram das reuniões. A avaliação utilizou a aplicação de um instrumento de *survey*. Os detalhes sobre o planejamento do estudo de caso explanatório, da sua execução e dos resultados obtidos, são apresentados no Capítulo 6 – Resultados do estudo de caso explanatório.

3.6 Etapa 6 – Refinamento da metodologia proposta e criação de um protótipo para auxiliar na avaliação da metodologia

De acordo com as etapas propostas para o nosso método de pesquisa, após a realização do estudo de viabilidade (Etapa 3), foram reunidos subsídios para a proposição de uma metodologia (Etapa 4) para aquisição de conhecimento de reuniões de projeto. A metodologia neste ponto foi proposta com base no resultado das etapas anteriores, mas com a realização do estudo de caso explanatório, alguns pontos foram revistos e modificados.

O estudo de caso explanatório serviu para modificarmos e aperfeiçoarmos pontos da metodologia proposta, em relação ao que havia sido concebido até o estudo de viabilidade, que são os seguintes:

- no roteiro para a realização das reuniões havia sido previsto o uso de “tags” para demarcar partes da reunião, com intuito de facilitar a localização de conceitos na reunião e até atribuir pesos diferentes aos conceitos em função de em qual parte da reunião ele foi mais utilizado. No estudo de caso explanatório percebeu-se que não se justificava a demarcação de “tags” separando partes da reunião, por motivos apontados na Etapa 1 da metodologia, seção 5.1;
- no estudo de viabilidade o “ponto de corte” que indicaria se um termo deveria ser considerado relevante para ser usado como indexador de uma reunião, era quando este apresentava uma frequência de ocorrência no texto igual ou maior a três. No próprio estudo de viabilidade foi apontada a necessidade de intensificar testes para aperfeiçoar a definição deste “ponto de corte” que indicaria a faixa de frequência na qual os termos deveriam ser considerados. Na Etapa 6 da metodologia proposta, seção 5.6, é apresentado como foi aperfeiçoado e definido, após a realização do estudo de caso explanatório, a faixa de frequência de ocorrência que inclui os conceitos relevantes de uma reunião.

Além do refinamento destes pontos da metodologia proposta, após o estudo de caso explanatório também foi elaborado um protótipo para realizar consultas à ontologia do domínio, acrescida de informações adquiridas das reuniões. A proposta para o uso do protótipo é realizar *queries* na ontologia, através de um mecanismo de inferência, que possam retornar, por exemplo, em quais reuniões determinado conceito, ou conjunto de conceitos, da ontologia foi abordado. O protótipo, além de apontar quais as reuniões estão associadas a determinado conceito, possibilita a visualização dos outros conceitos relevantes da reunião, a visualização do texto transcrito da reunião e a execução do áudio da reunião. Detalhes do funcionamento do protótipo são apresentados no Capítulo 6 – Resultados do estudo de caso explanatório.

3.7 Considerações sobre o capítulo

O objetivo deste capítulo foi apresentar as etapas necessárias para a execução do método de pesquisa proposto. Destacamos no método de pesquisa apresentado a realização de dois estudos de casos, um exploratório e outro explanatório, acompanhados de dois *surveys* que contribuíram significativamente para os requisitos de qualidade das conclusões apresentadas. Também destacamos a realização de uma revisão sistemática para apontar de forma mais precisa, dentro de um rigor científico desejado, as questões em aberto na área de pesquisa focada nesta tese.

4 Resultados preliminares

O objetivo deste capítulo é analisar os resultados obtidos com a realização das três etapas iniciais do método de pesquisa para esta tese. Estas três etapas apresentaram resultados que serviram de subsídios para a quarta etapa, onde é apresentada a proposta metodológica para aquisição de conhecimento em reuniões de projetos de desenvolvimento de software. Como estas etapas já foram introduzidas e apresentadas no capítulo 3, neste capítulo focaremos no desenvolvimento das mesmas e na análise dos resultados obtidos.

4.1 Estudo de caso exploratório

Entre julho e novembro de 2008 foi realizado um estudo de caso exploratório em uma empresa fabricante de computadores, que possui uma de suas fábricas de software instalada em Porto Alegre. Neste estudo de caso, esta empresa fabricante de computadores, utilizou uma empresa terceira para desenvolver parte de uma aplicação.

Durante o estudo de caso as empresas foram referenciadas como a Organização Proprietária (A) e a Organização *Offshore* (B). Foi analisado o trabalho das equipes que trabalharam em dois projetos diferentes, mas correlacionados nas organizações citadas. O primeiro projeto (Projeto 1) consistiu no desenvolvimento de um novo software e o outro (Projeto 2) foi um projeto de manutenção de software (ambos projetos possuem o mesmo escopo de negócio).

Para detalhar o estudo de caso, serão apresentados os seguintes tópicos: (i) uma explanação sobre o contexto de cada organização; (ii) os instrumentos utilizados para coleta de dados e; (iii) as considerações obtidas a partir de análise de dados. O método de pesquisa aplicado neste estudo de caso utilizou dados qualitativos e quantitativos coletados em questionários, análise documental e entrevistas semi-estruturadas.

4.1.1 Contexto das organizações

A chamada organização A desenvolveu um produto P que consiste em um software que controla as suas importações, e tem como intuito diminuir os impostos para este tipo de operação. Além disso, esse produto é dividido em dois subprodutos: subproduto 1, que implementa as regras e o processo de redução de impostos e o subproduto 2, que deve acelerar o processo de alfândega destes bens importados, reduzindo o esforço de controle e vigilância.

Estes subprodutos foram implementados com tecnologias atualmente obsoletas e não automatizaram algumas novas regras do negócio, o que implica em uma grande quantidade de

trabalho que acaba sendo realizado manualmente. No entanto, estes subprodutos tem sido mantidos continuamente pelo Projeto 2.

Devido às necessidades da organização, um novo projeto (Projeto 1), com tecnologias atualizadas, visa proporcionar uma automação de todas as regras de negócios e ficar conectado com as interfaces dos sistemas de comunicação do Governo. Devido ao grande número de *stakeholders*, à interface com vários outros sistemas e muitas restrições de acesso, este novo projeto tem um alto nível de complexidade.

Para o desenvolvimento deste novo projeto foram utilizadas duas equipes distribuídas. A primeira equipe, chamada S-Team, está localizada na região sul do Brasil. A outra equipe, chamada N-Team, está localizada na região nordeste. Ambas as equipes possuem colaboradores com o *SCRUM Master Certify*.

4.1.2 Instrumentos de Pesquisa

Para avaliar o perfil das pessoas envolvidas com os projetos, bem como para identificar as responsabilidades de cada um durante a execução dos projetos, definimos uma pesquisa chamada "survey de aspectos gerais da equipe". Esta pesquisa foi dividida em três partes: (i) questões gerais; (ii) questões sobre a experiência dos membros da equipe em diferentes metodologias de desenvolvimento de software e; (iii) questões sobre características específicas dos projetos.

As questões incluídas na primeira parte procuravam verificar: o nível educacional dos membros da equipe, quantos anos cada membro tinha de experiência profissional em Tecnologia da Informação, a relação de trabalho atual dos membros da equipe com as organizações (se empregado, contratado ou estagiário), há quantos meses estão trabalhando para as organizações, fluência dos membros da equipe no idioma Inglês e idade dos membros da equipe.

A segunda parte da pesquisa teve como objetivo avaliar o conhecimento dos membros da equipe em metodologias ágeis, tradicionais e híbridas. Para este item, cada membro da equipe foi questionado sobre: os anos de experiência profissional com cada metodologia, o número de projetos distintos que trabalharam com as metodologias durante a sua experiência profissional e; a utilidade da documentação gerada durante os projetos (se a documentação gerada sempre foi útil, útil na maioria das vezes que usada, útil algumas vezes, ou ela não era útil nas tarefas do projeto). Cada membro da equipe respondeu apenas o conjunto de questões relacionadas com as metodologias em que tinha experiência anterior.

Os membros da equipe com experiência em metodologias ágeis responderam mais quatro questões. As questões adicionais eram destinadas a verificar: (i) qual foi a metodologia ágil utilizada em projetos anteriores; (ii) se a empresa forneceu um treinamento formal em processos de desenvolvimento ágeis para a equipe; (iii) se os membros da equipe tinham alguma certificação em metodologias ágeis e; (iv) quais as características de metodologias ágeis estavam presentes nos processos utilizados em projetos anteriores.

A terceira parte do *survey* destinava-se a verificar os papéis e responsabilidades dos membros da equipe nos projetos. As questões incluídas nesta parte tiveram como objetivo identificar: (i) os papéis dos membros da equipe; (ii) a experiência do membro da equipe no domínio do problema; (iii) o tempo (percentual) de alocação estimado de cada membro da equipe para o projeto; (iv) a percepção de cada membro da equipe sobre o processo utilizado nos projetos (se o processo definido era principalmente uma metodologia tradicional ou ágil) e; (v) a percepção dos membros da equipe sobre a utilidade da documentação gerada durante os projetos.

O *survey* foi respondido por 15 dos 17 membros das equipes da Organização Proprietária e da Organização *Offshore* que participaram do Projeto 1 e do Projeto 2. O formulário respondido pelos membros da equipe encontra-se no Apêndice A deste trabalho.

Além da aplicação do questionário, os membros das equipes também foram entrevistados para ampliar a coleta de dados. O objetivo principal das entrevistas foi discutir sobre a documentação produzida e utilizada durante os projetos. Para tanto, os funcionários foram questionados sobre o tipo de documentação gerada nos projetos, quem gerava e disponibilizava a documentação, e como conhecimento tácito era trocado entre eles (quais canais de comunicação foram utilizados para troca de conhecimento de forma mais direta). As entrevistas foram realizadas de forma semi-estruturada. Quinze funcionários foram selecionados para serem entrevistados em reuniões individuais de cerca de uma hora. As questões abordadas nas entrevistas estão vinculadas aos documentos que abordam os requisitos do projeto que serviu de estudo de caso.

4.1.3 Análise dos Dados

O objetivo desta seção é analisar os dados obtidos através do questionário e entrevistas já descritos. Vamos apresentar apenas as informações pertinentes obtidas através de uma análise estatística. O objetivo foi identificar quais as tendências e as correlações entre os dados analisados podem trazer alguma contribuição no âmbito do presente estudo.

O foco desta análise é compreender aspectos relevantes fazendo uma comparação entre as metodologias ágeis e tradicionais, com foco na mudança do uso do conhecimento explícito para um maior uso do conhecimento tácito.

Nesse sentido, o núcleo da nossa análise estava centrado em questões relacionadas com documentação, conversações entre a equipe, forma de representação do conhecimento e como o conhecimento é adquirido e compartilhado.

A Figura 2 apresenta alguns dos dados relevantes obtidos através de análise de dados de questionários e entrevistas.

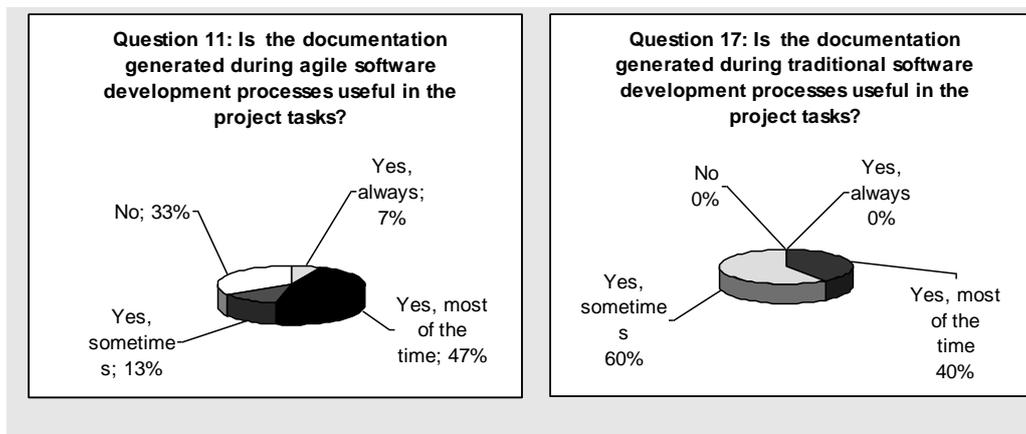


Figura 2 - Questões 11 e 17 do questionário

A Figura 2 ilustra as respostas às perguntas 11 e 17 do questionário. A pergunta 11 tentou verificar se os membros da equipe consideram útil a documentação em um processo ágil. Isso pode mostrar o grau de confiança que a equipe tem sobre os documentos gerados durante o processo de desenvolvimento de software. Houve mais respostas indicando que eles não consideram útil (33%) do que sempre é útil (7%).

A pergunta 17 do questionário tentou verificar com os membros da equipe como estes consideram útil a documentação em um processo tradicional. Não houve resposta indicando que eles não considerem útil (0%), nem que sempre é útil (0%).

A Figura 3 ilustra as respostas às perguntas 5 e 7 da entrevista. A pergunta 5 examinou se os entrevistados consideram atualizada a documentação gerada durante o processo de desenvolvimento de software. Quase todos (93%) consideram a documentação desatualizada.

A pergunta 7 examinou se os membros das equipes utilizam a documentação ou os demais membros da equipe e colegas como fonte prioritária de conhecimento sobre o projeto. A maioria (73%) consulta a documentação primeiramente. Deve ser enfatizado que esta questão define a

ordem de prioridade preferida pelo entrevistado no momento de tentar conseguir algum conhecimento sobre o projeto.

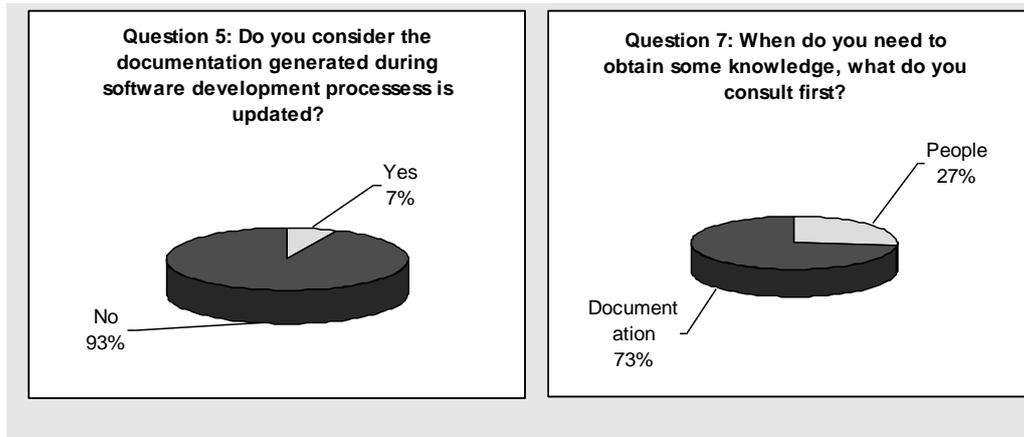


Figura 3 - Questões 5 e 7 da entrevista

A Figura 3 mostra os resultados sobre a utilidade da documentação no processo de desenvolvimento de software. O que chama a atenção nesses dados é que os entrevistados consideram mais importante a documentação em metodologias ágeis. Durante as entrevistas percebemos que estas respostas estão associadas ao fato de que em metodologias tradicionais eles consideram uma boa parte da documentação não útil. No entanto, a documentação que é utilizada em projetos ágeis é considerada mais útil.

Foi utilizado o coeficiente de correlação estatística de Pearson para avaliar a correlação entre as respostas das questões 11 e 17 e das questões 5 e 7. Para a questões 11 e 17 se obteve uma medida de correlação de 0,536. Esse valor não indica uma forte correlação entre os conjuntos de respostas duas questões.

A Figura 3 mostra que a maior parte dos entrevistados considerou a documentação desatualizada, mas ainda usa a documentação antes de consultar os colegas. O coeficiente da correlação de Pearson encontrado entre as questões 05 e 07 é de 0,99. Esse valor indica uma forte correlação entre o conjunto de respostas duas questões.

As principais análises apontadas nas lições aprendidas foram obtidas através da análise qualitativa dos dados dos questionários e, especialmente, a partir das entrevistas.

4.1.4 Discussões e lições aprendidas

De acordo com as observações extraídas do estudo de caso, sugerimos alguns itens a serem discutidos, como resultado das análises dos dados, que são os seguintes tópicos:

- a) O uso de metodologias ágeis em desenvolvimento de software distribuído intensifica os problemas de comunicação: neste estudo de caso o processo de desenvolvimento é

caracterizado como Desenvolvimento Distribuído de Software (DDS), onde as equipes estão localizadas em dois *sites*. Foram percebidos por meio de entrevistas e do acompanhamento das reuniões (*daily Scrum*), alguns problemas de comunicação entre os grupos distribuídos. A comunicação em DSD é, naturalmente, identificada como uma grande dificuldade [Pik08]. No contexto de projetos ágeis a comunicação se torna ainda mais relevante porque nesse tipo de projeto a comunicação com base em discussões é priorizada em relação a outras formas de intercâmbio de conhecimento. Korkala e Abrahamsson [Kor07] destacam que o desenvolvimento ágil de software envolve requisitos altamente voláteis, que são geridos através de uma comunicação verbal eficiente. Como a comunicação é um fator importante a ser observado, no nosso estudo de caso verificou-se que cada grupo tem seu próprio repositório de documentos e artefatos que não funcionam em sincronia. Não há uma prática estabelecida para o acesso e compartilhamento de artefatos nos repositórios. Isso levou os membros das equipes a intensificar a comunicação direta entre os integrantes dos dois *sites*. Se existissem mecanismos de compartilhamento de conhecimento (repositórios) que funcionassem de modo planejado e coordenado entre os grupos, a necessidade de comunicação através de conversações tenderia a diminuir. Isso diminuiria o impacto do fator comunicação, como um problema neste tipo de projeto, ágil em DSD. É importante ressaltar que não estamos sugerindo uma maior utilização dos repositórios e artefatos, em conflito com alguns pressupostos das metodologias ágeis, mas os artefatos que já estão disponíveis para os grupos devem ser devidamente compartilhados, reduzindo a necessidade de conversações sobre a informação que já está disponível em um dos sites.

- b) O uso intenso de comunicação informal em projetos ágeis cria problemas de aquisição de conhecimento e de armazenamento. Outro aspecto percebido sobre a comunicação entre os membros da equipe e os clientes é que a maioria das negociações realizadas durante o projeto é informal. Este formato torna a comunicação mais ágil, mas em compensação o conhecimento gerado durante as conversações só se mantém na forma tácita quando, mesmo em projetos ágeis, muitos detalhes devem ser registrados e armazenados para consultas posteriores ou confirmações [Rup03]. Um exemplo desta situação percebida no estudo de caso foi em relação aos requisitos levantados junto aos clientes. Muitos entrevistados (53%) relataram que é importante registrar as conversas com os clientes, de acordo com as constantes mudanças e atualizações nos

requisitos. Muitas vezes as conversas não são gravadas ou registradas criando um conflito de informações e interesses entre as partes envolvidas no processo. Esta questão suscita um dos principais pontos discutidos neste trabalho, que é a necessidade de uma nova perspectiva para uma metodologia de gestão do conhecimento a ser empregada nos processos de desenvolvimento ágeis. Esta metodologia deve ser centrada no conhecimento tácito que circula nos projetos. Como o objetivo não é tornar o processo de comunicação entre as partes interessadas burocrático, uma sugestão é usar mecanismos de reconhecimento automático de voz para organizar e resumir as conversações. Em outras palavras, haveria um esforço mínimo da equipe, com quase nenhuma interferência na forma como as negociações são realizadas, mas que poderia resultar na captura e registro do conhecimento em um formato que pode ser utilizado em conjunto com mecanismos de indexação e classificação, facilitando as consultas e o compartilhamento deste conhecimento.

- c) Apesar da documentação ser reduzida e ultrapassada, a equipe a usa como fonte de conhecimento para extrair o contexto do domínio e reduzir a comunicação direta.

Durante o processo de entrevistas foi observado que os entrevistados fizeram algumas críticas sobre a documentação disponível por ser insuficiente e muitas vezes desatualizada. Em um primeiro momento, poderíamos imaginar que os respondentes não utilizavam a documentação por considerá-la ultrapassada e inadequada. No entanto, apesar desta situação, os entrevistados utilizam a documentação. Eles explicaram que, pelo menos, os documentos fornecem o contexto do conhecimento que procuram. Eles também os utilizavam para reduzir o tempo de comunicação direta. Isto é, quando eles precisavam obter algum conhecimento, se eles já haviam consultado os documentos, eles acreditavam que isso iria diminuir o tempo das conversações com os colegas e clientes. A maioria dos entrevistados comentou que os documentos no início do projeto são atualizados, mas no decorrer do projeto vão se tornando desatualizados. Os documentos de requisitos são os mais citados entre os que estão desatualizados durante o projeto.

- d) Problemas de comunicação com o cliente ao utilizar uma metodologia híbrida. Muitos entrevistados comentaram sobre a dificuldade de contato com o cliente. Quando eles consultavam os documentos com os requisitos e estes estavam desatualizados, eles precisavam então entrar em contato com o cliente que não estava sempre disponível. A forma como os requisitos foram tratados neste projeto não se encaixa corretamente,

nem com os princípios de metodologias ágeis, nem das metodologias tradicionais. Se o projeto segue os princípios ágeis, o cliente deve estar mais acessível e disponível para toda a equipe. Se ele segue os princípios de metodologias tradicionais, então, existe a exigência dos documentos serem atualizados durante todo o projeto. Sob o enfoque da gestão do conhecimento, quando se adota uma metodologia ágil, a escolha é pelo aumento de comunicação entre a equipe em si e entre esta e o cliente. Quando se adota uma abordagem tradicional, a escolha é por priorizar o conhecimento explícito, dando mais ênfase ao uso de artefatos e documentos, que devem ser atualizadas ao longo do ciclo de vida do projeto [Cha03]. Portanto, quando se utiliza uma metodologia híbrida, deve-se tomar cuidado para evitar que estas questões de comunicação e documentação acabem não sendo atendidas do ponto de vista das metodologias ágeis, nem das metodologias tradicionais, como foi o caso dos requisitos neste projeto. Há uma falta de definição do que é uma metodologia híbrida. Qumer e Henderson-Sellers [Qum08] apresentam um *framework* para tentar determinar o grau de agilidade de algumas metodologias. A fronteira entre ser ágil e não ser híbrido, e ser tradicional e não ser híbrido é muito tênue e merece um melhor debate na comunidade.

4.1.5 Considerações sobre o estudo de caso exploratório

Entre alguns pontos elencados na conclusão do artigo que retratou o estudo de caso, destacamos: “Neste estudo de caso e em trabalhos correlatos percebemos a existência de uma lacuna na gestão do conhecimento para metodologias ágeis. A gestão do conhecimento em projetos tradicionais é focada na utilização do conhecimento explícito e é uma tarefa complexa. Mas em projetos ágeis, a gestão do conhecimento agrega ainda outros componentes, pois o conhecimento na forma tácita envolve muitos aspectos subjetivos”.

Do ponto de vista da pesquisa aqui proposta, este estudo de caso exploratório serviu adequadamente para o reconhecimento inicial do problema, onde também foram estudados conceitos relevantes à formulação da metodologia proposta, entre os quais, conceitos de gestão do conhecimento associado às metodologias ágeis, pela preferência no uso do conhecimento em seu formato tácito e pelo incremento da comunicação entre os *stakeholders*. Os resultados detalhados do estudo de caso foram apresentados em artigo na 11th International Conference – ICEIS 2009 [Yan09].

4.2 Revisão sistemática sobre gestão do conhecimento em projetos de software ágeis

Com o objetivo de sumarizar os principais aspectos envolvidos na gestão do conhecimento em projetos de desenvolvimento que seguem metodologias ágeis e identificar possíveis lacunas na área, foi proposta a execução de uma revisão sistemática. A motivação para a realização da revisão surgiu de estudos preliminares na literatura, onde não foram encontrados trabalhos que sumarizassem, no contexto da engenharia de software, os principais aspectos que emergem da gestão do conhecimento quando o uso do conhecimento explícito é preterido ao uso do conhecimento tácito, situação comum em metodologias ágeis.

De acordo com Kitchenham [Kit04], uma revisão sistemática permite identificar, avaliar e interpretar toda a literatura relevante para uma determinada questão de pesquisa, tópico ou fenômeno de interesse. Entre as razões para se realizar uma revisão sistemática, citam-se: a sumarização dos benefícios ou limitações de determinado tratamento ou tecnologia; a identificação de lacunas na pesquisa corrente, o que possibilita a sugestão de novas áreas para investigação futura; e a aquisição de conhecimento sobre determinada área ou assunto, permitindo o correto posicionamento de novas atividades de pesquisa.

Em Dyba e Dingsøyr [Dyb08a] é apresentado um trabalho sobre revisões sistemáticas em engenharia de software, onde são destacados aspectos que evidenciam a relevância deste tipo de pesquisa para a área de TI. Estes autores também apresentam, adaptado do trabalho de Kitchenham [Kit04], os estágios do processo de uma revisão sistemática, que são:

1. Planejamento da revisão
 - a. Identificação da necessidade para um revisão
 - b. Desenvolvimento de um protocolo
2. Condução da revisão
 - a. Identificação da pesquisa
 - b. Seleção dos estudos primários
 - c. Avaliação da qualidade do estudo
 - d. Extração dos dados
 - e. Síntese dos dados
3. Relatório da revisão

A seguir apresentaremos as principais etapas do processo da elaboração da revisão proposta, de acordo com Brereton *et al.* [Ber07], Kitchenham [Kit04] e Dyba e Dingsøyr [Dyb08a].

4.2.1 Planejamento da revisão

Nesta fase da revisão é definido um protocolo que descreve, entre outras coisas, o propósito da revisão e os procedimentos que serão adotados na sua execução. A definição de um protocolo garante também a replicação da revisão sistemática.

Um fator importante da revisão sistemática proposta é que ela tomou como base outras duas revisões já publicadas que serviram de partida para os nossos estudos. Uma delas é uma revisão sobre gestão do conhecimento em engenharia de software [Bjø08]. Nesta revisão o contexto de engenharia de software é genérico, não focando, por exemplo, em metodologias tradicionais ou ágeis. A outra revisão sistemática é sobre estudos empíricos de desenvolvimento de software ágil [Dyb08]. Nesta revisão não são focados aspectos de gestão do conhecimento aplicados a este tipo de metodologia de desenvolvimento de software.

Diante dos estudos preliminares que já haviam sido realizados, como o estudo de caso exploratório e, diante do quadro em que as duas revisões sistemáticas mais próximas do nosso trabalho não abordam o foco de nossa pesquisa, emergiu então a necessidade de realizarmos uma revisão sistemática que apontaria justamente na intersecção entre gestão do conhecimento em engenharia de software e projetos de desenvolvimento de software ágeis. Esta revisão serviu para identificar, avaliar e interpretar em toda a literatura relevante uma base teórica para os próximos passos da nossa metodologia.

A nossa revisão sistemática tinha como objetivo fornecer uma visão geral de estudo dentro da gestão do conhecimento em projetos de software ágeis, que tipos de conceitos têm sido explorados, quais são os principais resultados encontrados e os métodos de pesquisa utilizados. Mais especificamente nós procuramos responder as seguintes questões de pesquisa:

- Quais são os principais conceitos em gestão do conhecimento que têm sido investigados em projetos de software ágeis?
- Quais são os principais resultados encontrados em pesquisas sobre gestão do conhecimento em projetos de software ágeis?

Ao responder estas duas questões também estaremos verificando se existe uma área que surge da intersecção da gestão do conhecimento e projetos de software ágeis e se há contribuição suficiente para caracterizá-la como uma área de pesquisa.

O protocolo definido para esta revisão sistemática identificou então: (i) a questão de pesquisa (planejamento da revisão); (ii) a estratégia de pesquisa e fonte de dados, critérios de

inclusão e exclusão, seleção dos estudos primários, critérios de qualidade e métodos de síntese (condução da revisão) e; (iii) os resultados, as discussões e as conclusões (relatório da revisão). Os tópicos “condução da revisão” e “relatório da revisão” serão detalhados nas seções a seguir.

4.2.2 Condução da Revisão

Na condução da revisão as etapas definidas no protocolo são executadas, sendo que algumas etapas são realizadas em seqüência e outras puderam ser executadas simultaneamente. Algumas destas etapas foram realizadas por um dos autores da revisão, sendo que outras, por uma questão de exigência de qualidade do método, tiveram de ter a participação de dois dos autores da revisão sistemática.

4.2.2.1 Estratégia de pesquisa e fontes de dados

Para definir a *string* de busca a ser utilizada, foi aproveitado o esforço já realizado nas revisões sistemáticas *Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used* [Bjø08] e *Empirical studies of agile software development: A systematic review* [Dyb08]. Como a nossa revisão é uma intersecção entre estas duas revisões, partimos das *strings* utilizadas nos dois trabalhos para elaborarmos a nossa. Embora tenhamos utilizados as duas *strings* das revisões mencionadas, criamos outras e fizemos alguns testes comparativos para verificar a relevância dos resultados, de forma qualitativa. Após estes testes a *string* resultante utilizada em nossa revisão é a soma das *strings* das revisões mencionadas, que ficou a seguinte:

```
("software engineering") OR ("software process") OR ("learning software organization")
OR ("software development")) AND (("knowledge management") OR ("tacit knowledge") OR
("explicit knowledge") OR ("knowledge creation") OR ("knowledge acquisition") OR ("knowledge
sharing") OR ("knowledge retention") OR ("knowledge evaluation") OR ("knowledge use") OR
("knowledge application") OR ("Organization knowledge") OR ("knowledge engineering")) AND
(("agile software") OR ("agile methodologies") OR ("agile methods") OR ("agile process") OR
("extreme programming") OR ("xp software") OR ("scrum software") OR ("crystal software") OR
("dscdm software") OR ("fdd software") OR (feature driven development software) OR (lean
software development))
```

Simultaneamente ao processo de elaboração das *strings* foram selecionadas as fontes para a realização da busca dos estudos com a *string* resultante. Os serviços de indexação/bibliotecas digitais selecionados foram:

- ACM <http://portal.acm.org/dl.cfm>
- IEEE <http://ieeexplore.ieee.org>
- Scopus <http://www.scopus.com>
- Science@Direct <http://www.sciencedirect.com/>

4.2.2.2 Critérios de inclusão e exclusão

Para definirmos um critério de inclusão e exclusão dos trabalhos a serem selecionados, foi feita uma classificação com cinco níveis para enquadrar os trabalhos que atenderam a *string* de busca nos mecanismos de pesquisa definidos.

A revisão sistemática incluiu artigos publicados do ano 2000 em diante, sendo que a pesquisa foi realizada em Setembro de 2009.

Os níveis definidos para inclusão e exclusão dos artigos na revisão foram elaborados para serem aplicados em todos os artigos que foram retornados nos mecanismos de busca selecionados, como apresentado na Tabela 2.

Tabela 2 - Níveis de classificação dos artigos

Nível	Descrição
5	O foco do artigo é gestão do conhecimento em metodologias ágeis.
4	O artigo trata sobre gestão do conhecimento em metodologias ágeis, mas de forma periférica. Não é o foco, mas é tratada de forma secundária no artigo.
3	O artigo é sobre gestão do conhecimento ou sobre métodos ágeis. O artigo foca sobre um destes tópicos, mas o outro aparece no contexto do artigo.
2	O artigo é sobre gestão do conhecimento ou sobre métodos ágeis.
1	O foco do artigo não é gestão do conhecimento, nem métodos ágeis.

4.2.2.3 Seleção dos estudos primários

Para a seleção dos estudos primários, foi definido um processo que iniciou com a aplicação da *string* definida nos mecanismos de busca/indexação escolhidos e, é finalizado com a escolha dos artigos de onde foram retiradas as sínteses e as contribuições para a revisão sistemática.

A Tabela 3 mostra o processo da seleção dos estudos primários e o número de artigos identificados em cada estágio.

Tabela 3 - Etapas do processo de seleção dos artigos

Estágio	Task	Result
1	Aplicação da string aos mecanismos de busca selecionados	2467 artigos encontrados
2	Aplicação da string aos mecanismos de busca selecionados, considerando: - a busca foi restrita ao título, palavras-chave e resumo - a busca foi restrita a artigos completos (<i>full paper</i>)	2249 artigos descartados 218 artigos restantes
3	Os 218 artigos restantes foram importados para ferramenta Mendeley. Classificação destes artigos por um dos critérios (de 1 a 5 de acordo com a Tabela 2) baseado na leitura do título e <i>abstract</i>	Os 218 artigos restantes foram classificados: 48 artigos como Nível 1; 96 artigos como Nível 2; 42 artigos como Nível 3; 18 artigos como Nível 4; 14 artigos como Nível 5
4	Exclusão dos artigos classificados com Níveis 1 e 2	74 artigos restantes: 42 artigos como Nível 3; 18 artigos como Nível 4; 14 artigos com Nível 5
5	Leitura da Introdução e Conclusão dos artigos classificados com Nível 3 e exclusão dos artigos que não contribuem para o foco da pesquisa	Exclusão de 38 dos 42 artigos de Nível 3: 4 artigos restantes como Nível 3 Número total de artigos restantes: 36 4 artigos como Nível 3; 18 artigos como Nível 4; 14 artigos como Nível 5.
6	Leitura dos 36 artigos restantes e aplicação do questionário de avaliação de qualidade	16 artigos eliminados pela avaliação de qualidade 20 artigos restantes usados como estudos primários para a <i>synthesis of findings</i>

4.2.2.4 Critérios de qualidade

Nosso trabalho ancorou os aspectos relativos à qualidade nos seguintes pontos:

- para classificar os artigos encontrados pelos mecanismos de busca, os autores utilizaram o níveis de 1 a 5 conforme Tabela 2. O uso destes níveis de classificação

assegurou que as contribuições usadas eram centradas no foco da pesquisa proposta. Os artigos foram classificados pelos autores independentemente e quando as classificações eram diferentes, era feita uma discussão para chegar a um consenso;

- para avaliar os 36 artigos restante no estágio 5 (Tabela 3), os dois autores individualmente usaram o *Critical Appraisal Skills Programme* – CASP para avaliarem estudos com resultados empíricos, pois este método possui uma aplicabilidade ampla para avaliar qualidade de estudos de engenharia de software. Também foi elaborado um instrumento para avaliar os estudos sem resultados empíricos.

Não separamos os trabalhos com evidências de resultados empíricos dos que só apresentavam *Lições aprendidas*, pois manter na revisão sistemática somente trabalhos com resultados empíricos reduziria muito a quantidade de artigos a serem utilizados para sintetizar as descobertas e conclusões.

4.2.2.5 Síntese dos estudos encontrados

Esta síntese é um resumo dos principais conceitos encontrados nos trabalhos selecionados que foram lidos na íntegra. Estes tópicos foram extraídos por fazerem parte do foco principal dos artigos selecionados, ou mesmo não sendo o tópico principal do artigo, terem sido amplamente abordados no decorrer do mesmo.

Para a elaboração da síntese dos estudos desta revisão sistemática foi usado o método de *meta-ethnography* apresentado por Noblit e Hare [Nob88]. Também foram consideradas algumas sugestões propostas de técnicas de *ethnography* e análise qualitativa de dados encontradas em Oates [Oat06], Dixon-Woods *et al.* [Dix06], Dybå *et al.* [Dyb07].

Para realizar os passos do método de *meta-ethnography*, os autores fizeram uma leitura completa dos artigos usando a ferramenta Mendeley, onde foram registrados os principais conceitos e definições identificados em cada artigo. Após, os autores criaram um mapa conceitual para cada artigo lido. Depois disso, foram criados mapas conceituais para co-relacionar todos os conceitos dos artigos lidos. Esta síntese foi extraída dos mapas resultantes.

Tabela 4 - Síntese dos estudos encontrados

Tópico	Qtde	Referências
Problemas e aspectos específicos de documentação em métodos ágeis	6	[S1] [S2] [S3] [S4] [S5] [S6]
Características que envolvem a transferência e colaboração do conhecimento tácito durante o desenvolvimento de software	5	[S1] [S2] [S6] [S7] [S8]
Adoção de uma metodologia de gestão do conhecimento em projetos ágeis	5	[S9] [S10] [S11] [S12] [S13]
Uso de ferramentas para gestão do conhecimento no contexto de projetos ágeis	5	[S2] [S3] [S14] [S15] [S16]
A influência dos fatores humanos e sociais nas equipes e entre a equipe e os clientes	3	[S1] [S17] [S18]
O uso de comunidades de prática para compartilhar conhecimento	3	[S2] [S6] [S19]
Artefatos de conhecimento e conhecimento de experiência e como estes são usados no processo de desenvolvimento de software	2	[S11] [S20]

Relação dos artigos selecionados como *primary studies*:

- [S1] G. Melnik and F. Maurer, "Direct verbal communication as a catalyst of agile knowledge sharing," Proceedings of the Agile Development Conference, ADC 2004, 2004, pp. 21-31.
- [S2] H. Holz and F. Maurer, "Knowledge management support for distributed agile software processes," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 2640, 2003, pp. 60-80.
- [S3] H. Holz and J. Schafer, "Collaborative, task-specific information delivery for agile processes," Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on, 2003.
- [S4] T. Dingsøyr and G.K. Hanssen, "Extending agile methods: Postmortem reviews as extended feedback," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 2640, 2003, pp. 4-12.
- [S5] B. Crawford, C. Castro, and E. Monfroy, "Knowledge management in different software development approaches," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 4243 LNCS, 2006, pp. 304-313.
- [S6] A.R. Yanzer Cabral, M. Blois Ribeiro, A.P. Lemke, M.T. Silva, M. Cristal, and C. Franco, "A case study of knowledge management usage in agile software projects," 11th International Conference – ICEIS 2009, Milan, Italy: 2009, pp. 627-638.
- [S7] G. Canfora, A. Cimitile, F. Garcia, M. Piattini, and C.A. Visaggio, "Confirming the influence of educational background in pair-design knowledge through experiments," Symposium on Applied Computing, 2005.
- [S8] B. Ramesh, P. Cao, L. Mohan, and K. Xu, "Can distributed software development be agile?," Communications of the ACM, vol. 49, 2006, pp. 41-46.

- [S9] T. Kahkonen and P. Abrahamsson, "Digging into the fundamentals of extreme programming building the theoretical base for agile methods," Euromicro Conference, 2003. Proceedings. 29th, 2003.
- [S10] J. Ken H. and K.-B. Ilio, "Using Agile Practices to Spark Innovation in a Small to Medium Sized Business," 2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07), 2007, p. 275b-275b.
- [S11] J.K. Kokkonen, "Gathering Experience Knowledge from Iterative Software Development Processes," Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008), 2008, pp. 333-333.
- [S12] B. Bahli and E.S.A. Zeid, "The Role of Knowledge Creation in Adopting Extreme Programming Model: An Empirical Study," 2005 International Conference on Information and Communication Technology, IEEE, 2005, pp. 75-87.
- [S13] F.K.Y. Chan and J.Y.L. Thong, "Acceptance of agile methodologies: A critical review and conceptual framework," Decision Support Systems, vol. 46, 2009, pp. 803-814.
- [S14] M. Kuniavsky and S. Raghavan, "Guidelines are a tool: building a design knowledge management system for programmers," DUX '05: Proceedings of the 2005 conference on Designing for User eXperience, San Francisco, California: 2005, p. 8.
- [S15] M. Tomic, V. Milicevic, and M. Stankovic, "Collaborative Knowledge Acquisition for Agile Project Management," EUROCON 2005 - The International Conference on "Computer as a Tool", IEEE, 2005, pp. 1081-1084.
- [S16] T. Chau and F. Maurer, "Knowledge sharing in agile software teams," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 3075, 2004, pp. 173-183.
- [S17] A. Law and R. Charron, "Effects of agile practices on social factors," International Conference on Software Engineering, vol. 30, 2005.
- [S18] S. Ryan and R.V. O'Connor, "Development of a team measure for tacit knowledge in software development teams," Journal of Systems and Software, vol. 82, 2009, pp. 229-240.
- [S19] T. Kahkonen, "Agile Methods for Large Organizations - Building Communities of Practice," Agile Development Conference, 2004, pp. 2-11.
- [S20] G. Salazar-Torres, E. Colombo, F.S. Correa Da Silva, C.A. Noriega, and S. Bandini, "Design issues for knowledge artifacts," Knowledge-Based Systems, vol. 21, 2008, pp. 856-867.

Com isso encerramos a parte da condução da revisão, sintetizando os resultados resultantes das leituras dos estudos primários selecionados de acordo com os critérios estabelecidos no protocolo.

4.2.3 Relatório da Revisão

Esta seção irá apresentar as discussões, contribuições e as conclusões extraídas da revisão sistemática. A seção "Resultados" da revisão será apresentada no Apêndice B, pois ela é bastante extensa e é um detalhamento de cada tópico que já foi sumarizado na Tabela 4. Desta sumarização que foi detalhada na seção "Resultados", surgiram as discussões sobre os principais tópicos elencados da área, que são apresentadas na próxima seção.

4.2.3.1 Discussões sobre o tema

Nesta seção irão ser apresentadas as discussões que emergiram da análise dos resultados observados nas seções anteriores e também iremos apontar as questões em aberto oriundas destas discussões. Estas discussões irão focar em como identificar e tratar os principais fatores que

influenciam e são influenciados pela preferência do uso do conhecimento tácito, ao invés do conhecimento explícito, em projetos ágeis. A Figura 4 ilustra os tópicos a serem discutidos nesta seção.

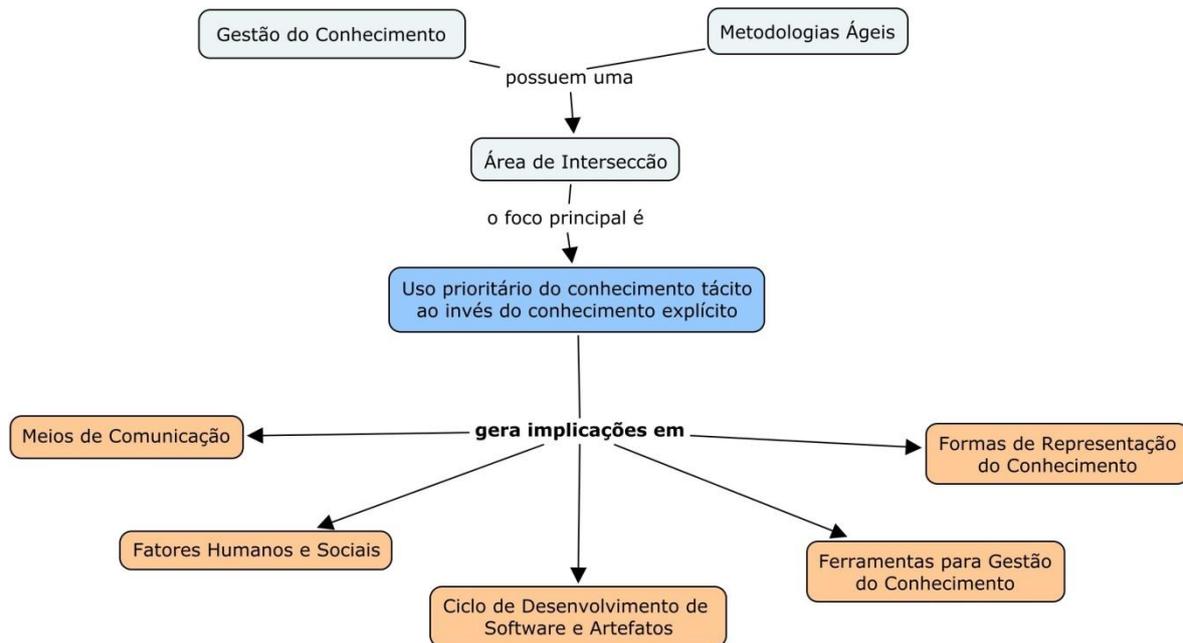


Figura 4 - Fatores que influenciam e são influenciados na GC em projetos ágeis

- Meios de comunicação:

Podemos observar na Figura 5 que o fator "Meios de comunicação" é derivado de todos os principais temas mencionados na síntese dos resultados (Tabela 4). No entanto, dos dez artigos que envolvem discussões sobre os meios de comunicação, seis deles se concentram em "Problemas e aspectos específicos de documentação em métodos ágeis" e "Características que envolvem a transferência e colaboração do conhecimento tácito durante o desenvolvimento de software". Portanto, problemas envolvendo a documentação em projetos ágeis e a manipulação de conhecimento tácito no desenvolvimento de software, estão fortemente associados com os meios de comunicação.

É também significativo que dos cinco artigos que abordam o uso de ferramentas de gestão do conhecimento em projetos ágeis, quatro deles envolvem discussões sobre os meios de comunicação. Consequentemente, as ferramentas que estão sendo desenvolvidas para apoiar a gestão do conhecimento em projetos ágeis estão focadas nos aspectos de comunicação.

A comunicação em projetos tradicionais prioriza o uso de conhecimento explícito. Sobre projetos ágeis, a prioridade é a comunicação direta dentro da equipe e entre a equipe e clientes, e

essa comunicação deve ser realizada de uma maneira mais direta (face a face), priorizando a conversa ao invés de outras formas de comunicação.

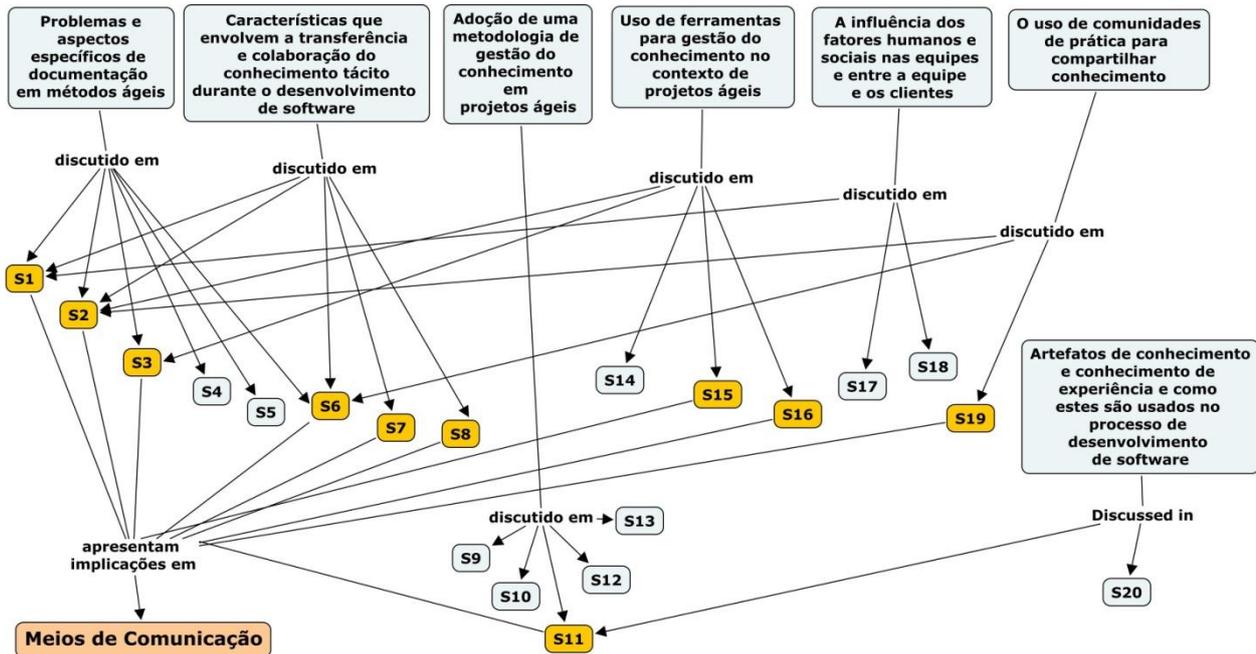


Figura 5 - Mapa conceitual sobre meios de comunicação

A questão em aberto apresentada para este tópico está focada em como uma metodologia de gestão do conhecimento deve lidar com o volume de informação gerado pela forma prioritária de comunicação em projetos ágeis, que é a conversação.

Um exemplo de como resolver este problema é o uso de técnicas e ferramentas que gerenciam as conversações entre os envolvidos no projeto. Isto pode ser feito pela adoção de comunidades de prática, que podem melhorar o compartilhamento do conhecimento através da comunicação direta entre os participantes. Também podemos fazer uso de ferramentas que auxiliam a gestão do conhecimento tácito. Poder-se-ia utilizar ferramentas para auxiliar no armazenamento e indexação das discussões entre a equipe e extrair algum conhecimento a partir das conversações de uma forma (semi) automática. Uma proposição para esta questão em aberto é o uso de mecanismos de gestão do conhecimento em projetos ágeis com base em sistemas de reconhecimento automático de voz [Ada10], [Fri09], [Jay09]. Um exemplo de um projeto como este, que já mostrou resultados é o CALO [Tur10].

- Fatores humanos e sociais:

Em artigos que abordaram "A adoção de uma metodologia de gestão do conhecimento em projetos ágeis" e "Artefatos de conhecimento e conhecimento de experiência e como estes são usados no processo de desenvolvimento de software", fatores humanos e sociais não foram

discutidos. Nos documentos que tratam de problemas de documentação, este tópico foi abordado de forma superficial e ocorre quando o artigo discute a questão do aumento da comunicação face a face.

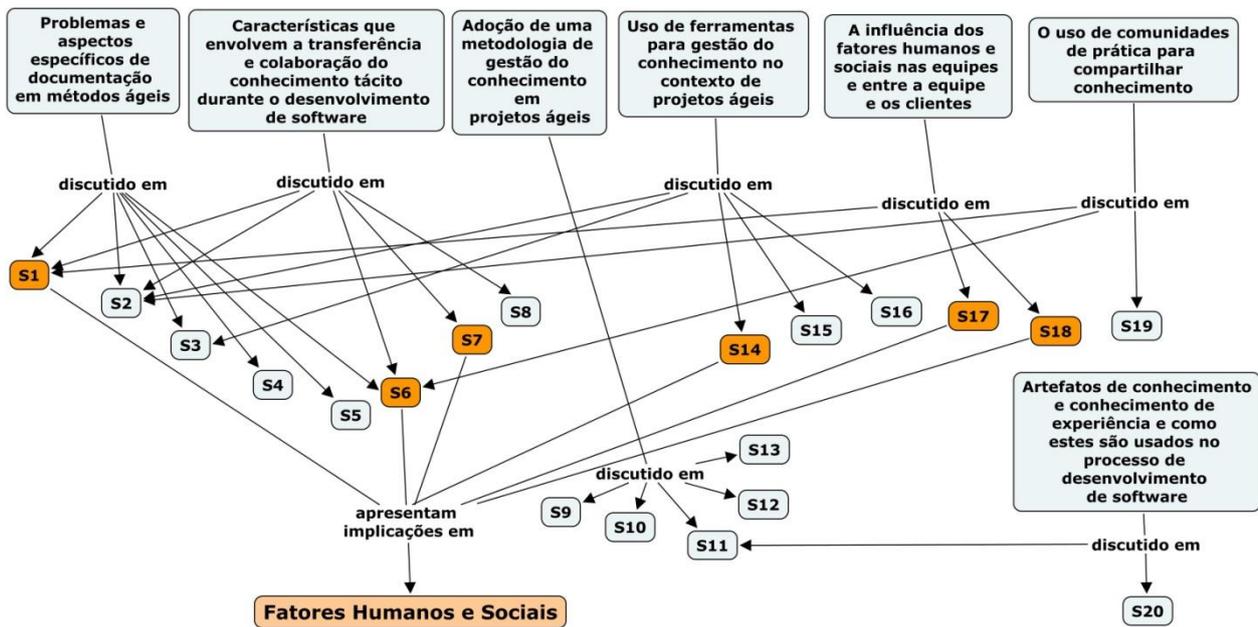


Figura 6 - Mapa conceitual sobre fatores humanos e sociais

Além dos artigos que focaram na questão da "influência dos fatores humanos e sociais nas equipes e entre a equipe e os clientes", este é também um tema de grande discussão em trabalhos sobre comunidades de prática, onde foi abordado em dois dos três artigos (Figura 6). É também uma questão muito discutida em artigos sobre "as características que envolvem a transferência de conhecimento tácito e colaboração durante o desenvolvimento do software", onde a maioria (3 de 5) dos artigos apontam as discussões sobre questões relacionadas a fatores humanos e sociais. Por este motivo, trabalhos que envolvem a manipulação de conhecimento tácito tendem a discutir os fatores humanos e sociais e também envolvem discussões sobre a comunicação face a face.

Fatores humanos e sociais tem sido uma questão muito discutida por um longo tempo na gestão do conhecimento e na engenharia de software. Podemos ver nestes artigos que este tópico reabriu novas discussões e ganhou novos contornos com o surgimento das metodologias ágeis.

A interação entre equipe, clientes e funcionários é alterada em projetos ágeis, porque eles precisam interagir com mais frequência e mais diretamente. Portanto, em projetos ágeis, para o conhecimento ser tratado de forma mais eficaz, é necessário que os envolvidos sejam preparados para interagir e colaborar na geração, compartilhamento e aquisição do conhecimento.

A questão em aberto neste tópico discute como os fatores humanos e sociais são tratados quando um projeto de desenvolvimento de software segue as práticas ágeis. Fatores como a empatia, educação, cultura, nível educacional, entre outros, podem influenciar o nível de integração do grupo e, portanto, interferem no canal de comunicação em que o conhecimento tácito flui. Quanto melhor esses fatores forem tratados no grupo, melhor o canal de comunicação será. Se este tipo de comunicação (direta) ocorre em um ambiente harmonioso, a comunicação direta acaba sendo mais eficaz do que a comunicação mediada (mais comum em projetos tradicionais). Por outro lado, se muitos problemas de cunho humano e sociais ocorrem, a comunicação direta é altamente prejudicada.

Devido à importância dos fatores humanos e sociais em projetos ágeis, propomos o uso, na formação de equipes para trabalhar com projetos ágeis, de uma avaliação do perfil da equipe que deve incluir aspectos que influenciam na gestão do conhecimento. Esta situação pode ser observada nos trabalhos apresentados por Chan e Thong [Cha09] e Young *et al.* [You05]. Nestes trabalhos os autores incluíram uma avaliação para verificar se determinadas personalidades e características pessoais emergem em diferentes papéis a serem desempenhados em um projeto. A sugestão é então estender a estas avaliações, um conjunto de características interpessoais que influenciam no desempenho de atividades relacionadas à gestão do conhecimento, principalmente focando na criação, retenção e transferência do conhecimento através da comunicação direta.

- Ciclo de desenvolvimento de software e artefatos:

O tópico "ciclo de desenvolvimento de software e artefatos" foi derivado de todos os tópicos, exceto de "ferramentas para gestão do conhecimento no contexto de projetos ágeis", sendo desta forma o tópico com o maior número de artigos analisados. Como mencionado anteriormente, os artigos que abordaram ferramentas de gestão do conhecimento oferecem discussões mais focadas sobre as formas de comunicação. No entanto, todos os trabalhos que discutem a "adoção de uma metodologia de gestão do conhecimento em projetos ágeis" sugerem mudanças e contribuições para o ciclo de desenvolvimento de software e artefatos. Por exemplo, em [S10] os autores sugeriram associar atividades e perspectivas de um modelo de processo de desenvolvimento ágil de software para as cinco etapas da criação do conhecimento organizacional propostas por Nonaka e Takeuchi [Non95]. Os autores sugerem um redesenho do ciclo de desenvolvimento de software de acordo com as fases de uma metodologia para gestão do conhecimento.

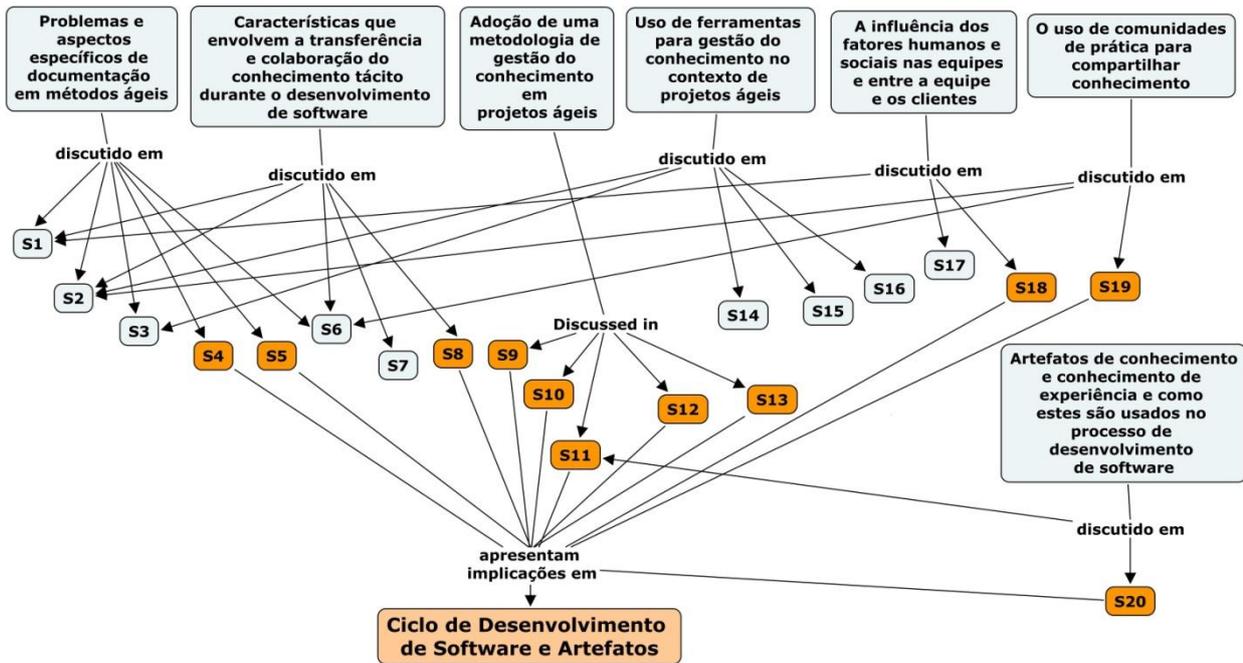


Figura 7 - Mapa conceitual sobre ciclo de desenvolvimento de software e artefatos

Ao comparar os métodos tradicionais de desenvolvimento de software com métodos ágeis, existem muitas diferenças no que diz respeito ao ciclo de desenvolvimento de software, que são amplamente discutidas na literatura [Dyb08]. Podemos observar que muitas dessas diferenças derivam do fato de que as metodologias ágeis priorizam o uso do conhecimento tácito. Este paradigma acaba influenciando o ciclo de desenvolvimento de software. Como exemplo, temos a metodologia *Scrum*, que promove reuniões diárias com a equipe, de modo que o conhecimento é compartilhado diretamente entre a equipe. Em métodos ágeis, reuniões diárias e outras atividades de interação entre as partes interessadas são comuns. Conhecimento é potencialmente armazenado na memória das pessoas e estas reuniões de trabalho servem para dar um *refresh* na memória. Como a maioria do conhecimento não é explícita, as reuniões freqüentes entre outras atividades servem para atualizar o conhecimento tácito.

Em abordagens tradicionais há outras formas priorizadas de compartilhar conhecimento entre os membros da equipe usando artefatos (conhecimento explícito). Portanto, a forma como o conhecimento flui no projeto influencia o ciclo de desenvolvimento de software. Quanto mais conhecimento é exteriorizado durante o projeto, mais artefatos são incluídos no ciclo de vida (e as atividades correspondentes para criá-los e mantê-los).

A questão em aberto envolve a influência da gestão do conhecimento no ciclo de desenvolvimento de software. Atividades e artefatos serão inseridos ou removidos no ciclo de desenvolvimento de acordo com o conhecimento que deve ser explicitado durante o projeto.

Assim, a definição de um ciclo de desenvolvimento de software deve estar diretamente ligada à concepção de gestão do conhecimento para o projeto. A maneira como o conhecimento é gerado, armazenado, validado, distribuído, apresentado e usado, influencia diretamente na definição do ciclo de desenvolvimento de software e de seus artefatos.

Nossa proposta para esta questão é a incorporação de artefatos de conhecimento para o ciclo desenvolvimento de software. Isto proporciona um tratamento mais direto do conhecimento em sua forma tácita. Esses artefatos podem ser baseados em formas de representação do conhecimento [Sow00], tais como áudio, vídeo e arquivos de imagem, transcrições automáticas, etc. No intuito de evitar que este tipo de conhecimento seja perdido, situação comum em empresas que trabalham com grandes volumes de informação, deve haver mecanismos de indexação desse conhecimento. Esses mecanismos de indexação podem ser baseados em uma representação formal do conhecimento que é computacionalmente processável, como por exemplo, as ontologias.

- Ferramentas para a gestão do conhecimento:

O tópico que discute ferramentas para gestão do conhecimento deriva claramente de todos os artigos de "ferramentas utilizadas em alguma fase da gestão do conhecimento no contexto projetos ágeis". Ele também deriva de alguns trabalhos que abordaram "problemas e aspectos específicos de documentação em métodos ágeis" e "características que envolvem a transferência e colaboração do conhecimento tácito durante o desenvolvimento de software". Com exceção do artigo de Kuniavsky e Raghavan [Kun05], todos os outros artigos que tratam de ferramentas de gestão do conhecimento também lidam com os meios de comunicação (Figura 8). Portanto, isso reforça que as ferramentas discutidas nos artigos que constituem os estudos primários desta revisão estão fortemente associadas com meios de comunicação em projetos ágeis.

Em projetos que seguem metodologias tradicionais, as ferramentas de gestão do conhecimento são mais focadas em situações que lidam com o conhecimento em formato explícito. Por outro lado, em projetos ágeis as ferramentas devem estar preparadas para oferecer alternativas eficazes para gerir amplamente o conhecimento de forma tácita.

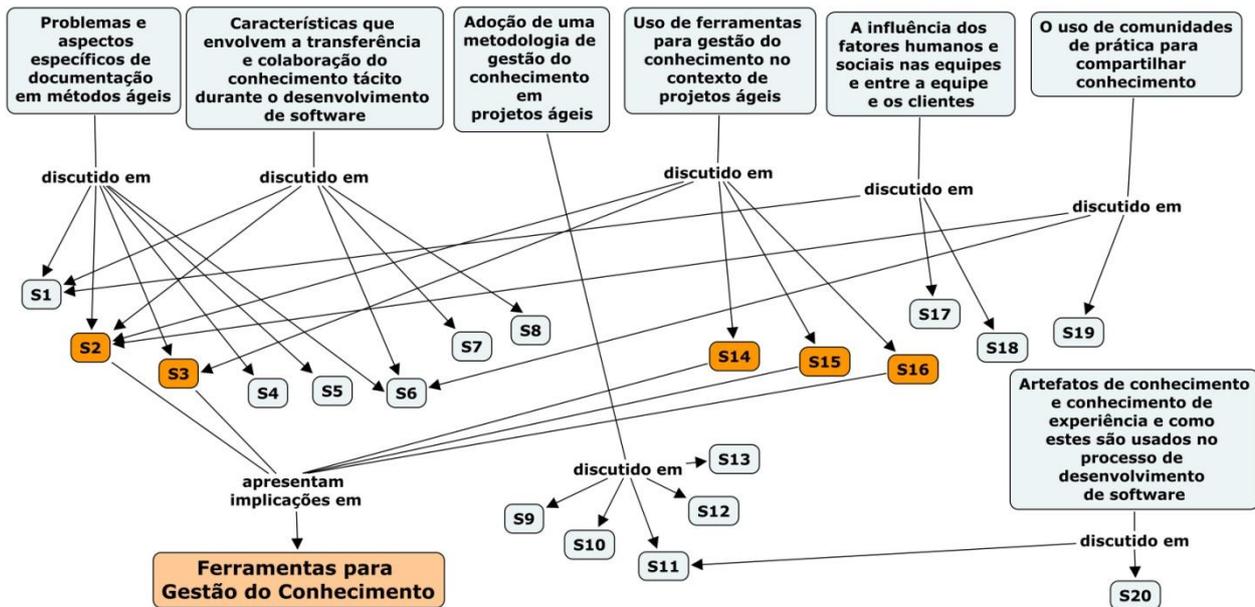


Figura 8 - Mapa conceitual sobre ferramentas para gestão do conhecimento

Não podemos ignorar o fato de que as ferramentas que trabalham com o conhecimento explícito fazem uso de uma fase de explicitação do conhecimento que cria um processo intermediário em que a "qualidade" da pessoa que faz este processo vai acabar influenciando na qualidade da aquisição e da representação do conhecimento. A aquisição de conhecimento tácito é geralmente feita de uma forma mais direta. No entanto, temos vários outros problemas associados com a gestão do conhecimento na forma tácita, como indexação, recuperação, compartilhamento, etc.

A questão em aberto neste tópico refere-se a como as ferramentas de gestão de conhecimento para projetos ágeis devem manter as características comuns usadas para projetos tradicionais, que manipulam os artefatos baseados em conhecimento explícito, e o que devem fazer a fim de incorporar funcionalidades para o gerenciamento de outras formas de representação do conhecimento.

Para abordar esta questão em aberto, enfatizamos a necessidade de aprofundar estudos para verificar a viabilidade tecnológica de ferramentas para lidar com o conhecimento tácito. Considerando o avanço em áreas como reconhecimento fala e imagem, processamento da linguagem natural, ontologias, entre outros. Seria então, viável o uso de ferramentas baseadas nessas tecnologias para a gestão do conhecimento tácito em projetos de desenvolvimento de software? É inviável o uso destes tipos de ferramentas devido a questões tecnológicas ou devido a sua ineficiência na adaptação aos processos de desenvolvimento de software? Estas são questões

que podem ser aprofundadas no uso de ferramentas para apoio à gestão do conhecimento em projetos ágeis.

- Formas de representação do conhecimento:

Artigos que apresentam implicações para "formas de representação do conhecimento" estão fortemente associados com o uso de artefatos de conhecimento e conhecimento de experiência (Figura 9). Mas eles também derivam, mais superficialmente, de discussões sobre documentação, colaboração e transferência de conhecimentos na forma tácita. Como apresentado em Torres-Salazar *et al.* [Tor05] existe uma preocupação em propor formas alternativas de representação do conhecimento em projetos ágeis, devido ao maior uso do conhecimento tácito.

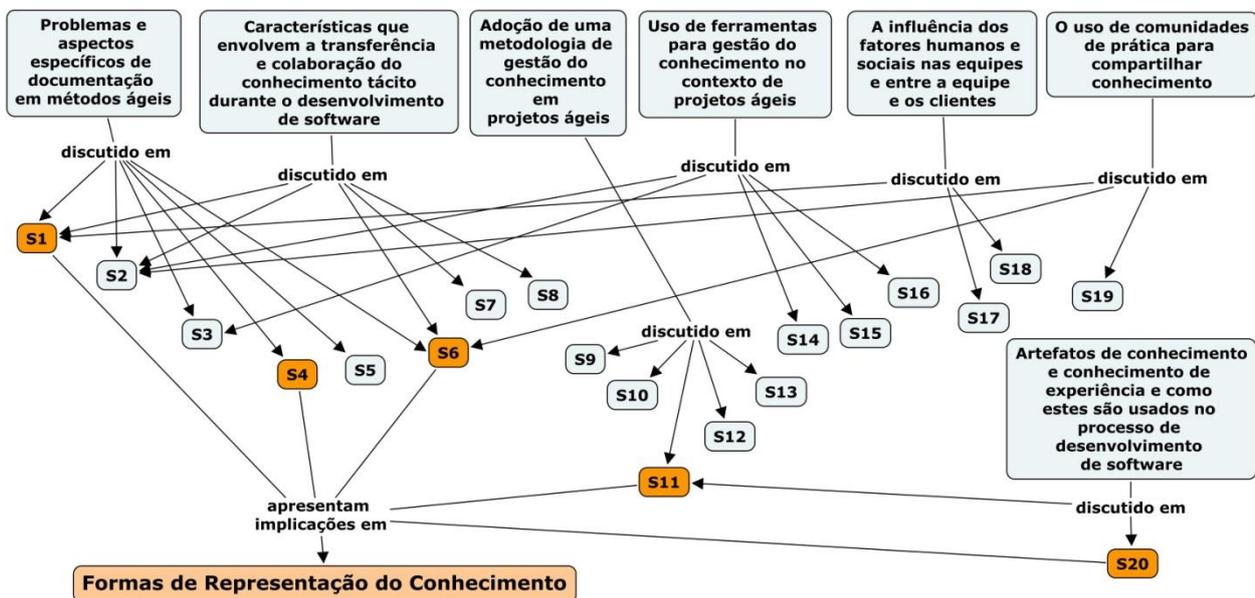


Figura 9 - Mapa conceitual sobre formas de representação do conhecimento

A questão em aberto para discussão neste tópico diz respeito a como o conhecimento deve ser representado nas diversas fases e etapas do ciclo de desenvolvimento de software de projetos ágeis, de forma que possa ser mais bem gerenciado. Se o conhecimento deve ficar somente concentrado, ou retido, nas pessoas, ou se é interessante tentar capturar o máximo possível deste conhecimento, mesmo que na forma tácita.

Poderíamos dar como um exemplo de proposição para este tópico, a utilização de formas de representação do conhecimento adequadas ao conhecimento tácito, como as gravações em áudio dos diálogos entre as pessoas ou os vídeos das conversações ou da execução das tarefas explanando e demonstrando como realizá-las. Também podem ser usadas formas de representação do conhecimento onde o mesmo foi explicitado, mas que procuram manter

aspectos mais comuns ao conhecimento tácito (difícil de ser representado em artefatos tradicionais) tais como mapas conceituais e mapas mentais. Novamente surge a questão da importância de conseguir indexar este conhecimento para que possa ser recuperado e distribuído de forma eficiente.

Como resultado das discussões realizadas ao longo da análise desses mapas conceituais, as nossas principais conclusões são:

- a prioridade do uso do conhecimento tácito ao invés de conhecimento explícito é o foco das discussões que envolvem a gestão do conhecimento em projetos que seguem metodologias ágeis. Em alguns trabalhos, esta questão é apresentada diretamente e em outros de uma maneira subjetiva. Esta questão central influencia os temas que foram levantados para discussão nesta revisão, como mostrado na Figura 4;
- o tópico mais impactado pelas discussões nos estudos foi "ciclo de desenvolvimento de software e artefatos". Isto significa que quando uma empresa adota práticas de gestão do conhecimento haverá mudanças no seu processo de desenvolvimento. Isto se deve ao fato de que as empresas precisam incluir ou alterar os artefatos, que possuem custos de implantação e adaptação ao ciclo de desenvolvimento;
- as ferramentas abordadas nos artigos estão focadas em melhorar a comunicação dos participantes dos projetos. Artigos que tratavam sobre ferramentas para gestão do conhecimento não apresentaram implicações diretas no ciclo de desenvolvimento de software;
- o segundo tema com maior número de implicações para as discussões foi "meios de comunicação". É derivado de todos os temas discutidos nos artigos. Ele está diretamente associado com a priorização do conhecimento tácito através da comunicação direta em projetos ágeis;
- questões relacionadas a fatores humanos e sociais, bem como as diferenças entre gestão do conhecimento tácito e conhecimento explícito não são novas discussões na área de gestão do conhecimento. No entanto, elas ainda são consideradas como tendo implicações significativas em projetos de desenvolvimento de software, especialmente com o surgimento das metodologias ágeis;

- a preocupação de propor novas formas de representação e manipulação de conhecimento foi observada nas discussões, especialmente em formatos mais semelhantes ao conhecimento tácito.

Levando em conta estas considerações, destaca-se como oportunidade de pesquisa o uso de ferramentas de gestão do conhecimento com implicações diretas no processo de desenvolvimento de software, e não apenas focadas em ferramentas de comunicação e colaboração. O foco dessas ferramentas deve ser em minimizar o impacto da aplicação de técnicas de gestão do conhecimento. Por exemplo, elas podem manipular o conhecimento em formatos alternativos (como áudio, vídeo, experiência, etc) no ciclo de desenvolvimento de software, o que implica no uso de ferramentas que manipulam o conhecimento prioritariamente em um formato mais próximo ao tácito.

4.2.4 Considerações sobre a revisão sistemática

A motivação e a justificativa para a realização desta revisão emergiram após: (i) estudos preliminares na literatura, onde não foram encontrados trabalhos que sumarizassem, no contexto da engenharia de software, os principais aspectos que emergem da gestão do conhecimento quando o uso do conhecimento explícito é preterido ao uso do conhecimento tácito, situação comum em metodologias ágeis e; (ii) após o estudo de caso exploratório que apontou para a existência de uma lacuna na gestão do conhecimento para metodologias ágeis e traz à tona problemas de comunicação neste tipo de projeto sugerindo, entre outros pontos, um mecanismo (semi) automático para aquisição e compartilhamento de conhecimento, baseado por exemplo, no uso de aplicações de reconhecimento automático de fala, confirmando a necessidade de uma revisão aprofundada sobre o tema.

Percebemos desta forma, que a revisão sistemática alcançou seus objetivos no contexto desta proposta de tese, ajudando a responder através do planejamento formal, da execução metódica e do alto rigor científico, que influências são percebidas na gestão do conhecimento quando o conhecimento que circula no projeto é trabalhado preferencialmente em seu formato tácito e quando o processo de explicitação do conhecimento é preterido pelo incremento da comunicação direta entre os membros da equipe.

A revisão sistemática apontou que existe uma grande contribuição desta área de pesquisa para a comunidade, pois aspectos que já vem sendo tratados pela academia e profissionais, como por exemplo: uso de artefatos; a aquisição, retenção e compartilhamento do conhecimento; formas de comunicação; fatores humanos e sociais; ferramentas e técnicas para gestão do

conhecimento; formas de representação do conhecimento, entre outros tópicos, estão ganhando um novo olhar em decorrência da mudança de um paradigma que prioriza o uso do conhecimento explícito, para um paradigma que prioriza o uso do conhecimento tácito.

As questões em aberto apresentadas na revisão sistemática e, as proposições de como estas podem ser abordadas, subsidiariam a elaboração da metodologia proposta apresentada neste trabalho.

4.3 Estudo de viabilidade

O intuito deste estudo foi realizar um conjunto de ensaios para verificar a viabilidade da aquisição de conhecimento a partir dos diálogos de reuniões de projetos. Este conhecimento adquirido servirá de base para criação de um formato que possa ser utilizado como um indexador das reuniões, em conjunto com uma ontologia do domínio. A aquisição do conhecimento foi feita pela análise do áudio da gravação das reuniões para extração dos conceitos relevantes. Consequentemente, um dos focos deste estudo de viabilidade foi analisarmos a relevância, ou qualidade, dos conceitos que são extraídos das reuniões de projeto.

Os ensaios utilizados para avaliar a qualidade dos conceitos, onde consideramos a capacidade dos mesmos para indexar o conhecimento capturado das transcrições das reuniões, foram realizados de três formas:

- comparando os conceitos gerados pelo texto original das reuniões com os da transcrição do áudio da reunião;
- comparando os conceitos presentes na ontologia do domínio com os da transcrição;
- comparando a análise de humanos sobre os conceitos relevantes das reuniões com os conceitos que foram gerados de forma automática.

A Figura 10 apresenta a estrutura do estudo de viabilidade de forma a melhorar a compreensão de suas fases, ferramentas e produtos gerados no decorrer da execução de suas etapas.

	Fases	Ferramenta	Artefato / Subproduto
1	Elaboração do roteiro para a reunião	Editor de Texto	Texto das conversações com marcadores
2	Realização da reunião	Sala Acústica; microfones alta sensibilidade; mixer; sw de captura de áudio	Arquivos MP3 das conversações
3	Transcrição da reunião	Dragon Naturally Speaking Speech Recognition	Texto transcrito do audio da reunião
4	Análise da WER (Word Error Rate) da transcrição	Editor de Texto	WER da transcrição
5	Etiquetar a transcrição	Tagger Stanford	Texto etiquetado
6	Comparar conceitos gerados: texto original X transcrição	Planilha Eletrônica	Análise da comparação dos conceitos gerados
7	Comparar conceitos gerados: ontologia X transcrição	Protégé; ONTrace; Planilha Eletrônica	Análise da comparação dos conceitos gerados
8	Survey: conceitos gerados automat. X análise humana	Editor de Texto Planilha Eletrônica	Avaliação humana dos conceitos gerados

Figura 10 - Etapas do estudo de viabilidade

1. Elaboração do roteiro para a reunião: utilizou-se o documento onde foram relatados todos os requisitos funcionais e não funcionais, assim como a visão geral do sistema referente ao projeto onde foi realizado o estudo de caso exploratório, da Etapa 1 do método de pesquisa (Reconhecimento inicial do problema). Este documento não é apresentado nos anexos por conter informações sigilosas da empresa onde foi aplicado o estudo de caso. A elaboração do roteiro seguiu o formato das reuniões que o grupo realizava de acordo com diretrizes da metodologia SCRUM. Foram colocados marcadores para dividir partes da reunião, onde, por exemplo, o condutor da reunião utilizava palavras chaves como “início da

reunião”, “detalhes do projeto”, “andamento das tarefas”, “encerramento da reunião”, etc. Portanto, criaram-se diálogos baseados em conversas que foram realizadas durante o projeto do estudo de caso exploratório, utilizando informações sobre o projeto e seguindo um protocolo de marcação dividindo a reunião em etapas.

2. Realização da reunião: as reuniões foram realizadas em salas com acústica especial utilizadas pelo departamento de EAD (Ensino a Distância) da PUCRS para gravação de aulas. As salas contam com equipamento especial, incluindo microfones de alta sensibilidade, mesa com *mixer* e software para captura de áudio. Como os melhores softwares para reconhecimento de voz não possuem versões para a língua portuguesa, as reuniões foram realizadas em inglês. Duas pessoas com fluência na língua inglesa e mais o autor desta tese foram os participantes das reuniões. As reuniões seguiram o roteiro definido no passo 1 (Elaboração do roteiro para a reunião), com cada participante lendo sua parte do texto e ao final de cada reunião eram gerados arquivos em formato “mp3” de todas as conversações. As reuniões eram gravadas sempre de duas formas: (i) cada participante realizava as suas falas no decorrer da reunião; (ii) após, um participante gravava a mesma reunião fazendo a fala de todos os participantes. O objetivo de ter estas duas versões da mesma reunião era verificar o nível de acerto do software de reconhecimento de fala quando comparado a fala de uma só pessoa, com a fala de diversas pessoas para o mesmo roteiro. Foram realizadas quatro reuniões.
3. Transcrição da reunião: foi utilizada a última versão do software *Dragon Naturally Speaking Speech Recognition*⁶, que é o principal produto deste segmento para fazer as transcrições das falas das reuniões⁷. Cada um dos participantes teve de gerar um perfil de usuário no software, sendo que essa tarefa serve para criar um padrão de reconhecimento de fala para cada usuário. Para cada reunião foram gerados dois arquivos texto com as transcrições, conforme as duas formas descritas no passo 2: (i) cada participante realizava as suas falas no decorrer da reunião; (ii) cada participante gravava a mesma reunião fazendo a fala de todos os participantes.

⁶ <http://www.nuance.com>

⁷ <http://voice-recognition-software-review.toptenreviews.com/>

4. Análise da WER – Word Error Rate da transcrição: esse passo consistiu em comparar, o texto que serviu de roteiro para a reunião, com a transcrição gerada pelo software de reconhecimento de fala. Foi feita uma comparação palavra a palavra, de forma manual, em todos os textos para encontrar as palavras que na transcrição saíram diferentes da fala durante a reunião, baseado no texto original das falas. Os resultados foram os seguintes:
- a. Reunião 1 - WER com a fala de todos: 13,9%; WER com a fala de um participante: 8,7 %.
 - b. Reunião 2 - WER com a fala de todos: 13,0%; WER com a fala de um participante: 7,9 %.
 - c. Reunião 3 - WER com a fala de todos: 11,6%; WER com a fala de um participante: 7,6 %.
 - d. Reunião 4 - WER com a fala de todos: 12,8%; WER com a fala de um participante: 7,4 %.

WER com a fala de todos significa que a reunião foi gravada com cada participante realizando a sua própria fala. Essa reunião foi transcrita e o WER aponta o percentual de palavras que foram transcritas erroneamente, baseado na comparação com o texto que foi lido pelos participantes na reunião.

WER com a fala de um participante significa que a reunião foi gravada com um participante realizando a fala de todos os “personagens” da reunião.

Essas duas medições foram feitas para comparar situações que depreciam o desempenho dos softwares de reconhecimento de fala, que é interpretar a fala de uma única pessoa, ou mais de uma pessoa ao mesmo tempo. Podemos observar então que na média, quando os três participantes da reunião realizaram as suas falas o WER foi de 12,9%, sendo que quando somente um dos participantes realizou a reunião, lendo a fala de todos, o WER caiu para 8,0%. Evidenciando desta forma, que ao analisar o áudio com a fala de mais de um participante, o desempenho do software de reconhecimento automático de fala é degradado.

5. Etiquetar a transcrição: neste passo é aplicado o processo de etiquetar o texto da transcrição das reuniões, que consiste em classificar cada palavra, atribuindo-lhe um rótulo conforme a Tabela 5. Para etiquetar foi utilizado o *tagger* produzido pela

universidade de Stanford⁸. Foi gerado, portanto, para cada reunião: um arquivo com o roteiro original etiquetado e um arquivo com a transcrição etiquetada. Desta forma, pode-se extrair somente os tipos de palavras que interessam, que no nosso caso foram, de acordo com a Tabela 5: *CD Cardinal Number*, *FW Foreign Word*, *NN Noun singular or mass*, *NNS Noun plural*, *NNP Proper noun singular*, *NNPS Proper noun plural*. Também foram feitos testes com sumarizadores, sendo que a saída destes normalmente são frases. Alguns até apontam os principais conceitos além das frases, como o Copernic⁹. Mas como o objetivo é extrair os conceitos das falas das reuniões, um etiquetador é mais adequado que um sumarizador, que também apresenta problema em ser aplicado em textos transcritos que não possuem pontuação, dificultando o entendimento de estrutura das frases e parágrafos.

6. Comparar conceitos gerados pelo texto original com os da transcrição: neste passo é feita uma análise dos conceitos que foram identificados pelo etiquetador no texto original das reuniões e, estes são comparados com os conceitos extraídos do texto transcrito das reuniões. O objetivo é verificar a correlação entre os conceitos que foram extraídos de um texto, que a priori representa fielmente a fala dos participantes, pois foi lido pelos mesmos e, os conceitos que foram extraídos da transcrição da fala dos mesmos. Os resultados obtidos dos ensaios são apresentados na seção 4.3.1.
7. Comparar os conceitos da ontologia do domínio com os da transcrição: neste passo o objetivo foi comparar os conceitos extraídos das transcrições das reuniões, com os da ontologia do domínio do projeto que serviu para o nosso estudo de caso exploratório. Para auxiliar nesta comparação foi utilizada uma aplicação (ONTrace) desenvolvida em um trabalho realizado junto ao ISEG na PUCRS [No109]. Esta aplicação possui um de seus módulos (*matcher*), que resumidamente, recebe uma lista de termos em formato texto e verifica a similaridade destes com os conceitos da ontologia. É utilizado um algoritmo de *stemmer* para realizar esta verificação. A comparação consistiu então, em pegar todos os conceitos que foram extraídos das transcrições das reuniões e verificar se estes conceitos ocorriam na ontologia do

⁸ <http://nlp.stanford.edu/software/tagger.shtml>

⁹ <http://www.copernic.com>

domínio. Os resultados obtidos e os detalhes deste ensaio são apresentados na seção 4.3.2.

Tabela 5 - Classificação das palavras pelo *tagger*

1. CC Coordinating conjunction	25. TO <i>to</i>
2. CD Cardinal number	26. UH Interjection
3. DT Determiner	27. VB Verb, base form
4. EX Existential <i>there</i>	28. VBD Verb, past tense
5. FW Foreign word	29. VBG Verb, gerund/present
6. IN Preposition/subordinating participle conjunction	30. VBN Verb, past participle
7. JJ Adjective	31. VBP Verb, non-3rd ps. sing. present
8. JJR Adjective, comparative	32. VBZ Verb, 3rd ps. sing. present
9. JJS Adjective, superlative	33. WDT wh-determiner
10. LS List item marker	34. WP wh-pronoun
11. MD Modal	35. WP\$ Possessive wh-pronoun
12. NN Noun, singular or mass	36. WRB wh-adverb
13. NNS Noun, plural	37. # Pound sign
14. NNP Proper noun, singular	38. \$ Dollar sign
15. NNPS Proper noun, plural	39. . Sentence-final punctuation
16. PDT Predeterminer	40. , Comma
17. POS Possessive ending	41. : Colon, semi-colon
18. PRP Personal pronoun	42. (Left bracket character
19. PP\$ Possessive pronoun	43.) Right bracket character
20. RB Adverb	44. " Straight double quote
21. RBR Adverb, comparative	45. ' Left open single quote
22. RBS Adverb, superlative	46. " Left open double quote
23. RP Particle	47. ' Right close single quote
24. SYM Symbol (mathematical or scientific)	48. " Right close double quote

8. Survey para avaliar os conceitos gerados automaticamente: esta etapa do estudo de viabilidade consistiu em submetermos à avaliação de humanos os conceitos que foram extraídos das reuniões do estudo de caso exploratório e que já foram comparados à ontologia do domínio no passo 7. Esta avaliação consistiu em comparar os “conceitos relevantes” identificados de forma automática em uma reunião, através do uso da nossa metodologia proposta, com os “conceitos relevantes” da reunião que foram apontados através da percepção humana. A partir de uma lista de termos, que são os conceitos da ontologia do domínio, os voluntários analisaram as reuniões e apontaram os termos relevantes para os tópicos tratados em cada reunião. O ensaio foi composto pelos seguintes passos: (i) analisar o glossário fornecido para um melhor entendimento das siglas e termos específicos do domínio que foi tratado nas reuniões; (ii) analisar cada uma das reuniões realizadas sobre o projeto (três reuniões) através da leitura completa de todas as reuniões e; (iii) apontar na planilha de avaliação os termos relevantes de

cada reunião. No apêndice C é apresentado o instrumento utilizado neste *survey*, com mais detalhes sobre a aplicação do mesmo. Participaram do *survey* oito professores de cursos superiores da área de computação e informática, que preencheram individualmente seus formulários. O autor desta tese reuniu-se com cada um dos participantes, antes da aplicação do formulário, para a explicação do contexto do estudo de viabilidade.

Os resultados das comparações propostas e do *survey* apresentados nos passos 6, 7 e 8 serão detalhadas nas próximas seções.

4.3.1 Comparando os conceitos do texto original com os da transcrição

Nesta seção são apresentadas as análises das reuniões realizadas no estudo de viabilidade, no intuito de verificar a margem de acerto da transcrição textual do áudio quando comparado ao texto no qual foi baseada a gravação da reunião.

No Apêndice D são apresentadas as informações de cada uma das reuniões realizadas e os dados coletados durante os ensaios. Nas próximas seções, estes dados serão comparados e analisados de forma a sintetizar os resultados destes ensaios.

4.3.1.1 Considerações sobre os resultados das reuniões

Nesta seção faremos algumas análises em função dos resultados obtidos com os experimentos, comparando os dados das reuniões (Apêndice D) e apontando conclusões e questões em aberto, que foram posteriormente testadas em experimentos da Etapa 5 do método de pesquisa proposto, onde é realizada uma validação através de um estudo de caso explanatório.

Tabela 6 - Resumo das reuniões

Reunião	WER >1	WER =1	Qtde Palavras	Qtde Conceitos		Porcentagem de acerto			
						Frequência >= 2	Frequência >= 3	Frequência >= 4	Frequência >= 5
1	13,9%	8,7%	1057	302	27,9%	84,5%	97,2%	100,0%	100,0%
2	13,0%	7,9%	1128	415	35,3%	81,4%	83,3%	80,0%	80,0%
3	11,6%	7,6%	854	321	36,3%	86,2%	93,0%	91,2%	94,1%
4	12,8%	7,4%	596	211	32,9%	80,0%	91,3%	92,3%	90,9%
Média	12,9%	7,9%	909	312	34,3%	83,0%	91,2%	90,9%	91,3%

No intuito de sistematizar as análises derivadas dos resultados das reuniões apresentados na Tabela 6, elencaram-se alguns tópicos relevantes que serão detalhados, a seguir:

- WER (*Word Error Rate*) significa a taxa de erro de palavras transcritas. Se consultarmos a literatura sobre reconhecimento automático de fala, observaremos que são informados vários índices sobre o desempenho destes softwares, e o principal índice é o WER. Eles irão variar de acordo como o tamanho do dicionário, condições acústicas, língua e estilo da fala, treinamento realizado com o usuário, entrada de áudio composta pela voz de uma única pessoa, ou não, entre outros [Ada10]. Alguns produtos atestam índices próximos a 99% de reconhecimento, em condições ditas ideais¹⁰. Em nossos ensaios obtivemos os índices de WER apontados na Tabela 6, que dizem respeito a transcrição de um áudio composto com a voz de mais de um participante (WER >1) e com a voz de somente um participante (WER =1). Quando o áudio é composto por somente uma voz este índice melhora substancialmente, pois a taxa cai de 12,9% para 7,9%. Percebemos que estes índices podem melhorar, pois foram usados muitos substantivos próprios como siglas e termos do domínio do negócio, assim como termos comuns à área de engenharia de software. O software pode ser treinado para que reconheça palavras que não fazem parte do seu dicionário. Também pode ser melhorado o treinamento do perfil do condutor/mediador das reuniões, que normalmente é a pessoa que mais fala e acaba sumarizando os assuntos tratados, de acordo com o protocolo sugerido. Mas consideramos bastante aceitável o índice encontrado (12,9%), baseado em contextos similares [Ada10];
- O que chamamos de “Porcentagem de acerto” é a relação entre os conceitos que foram extraídos pelo *tagger* do texto original da fala, comparados aos conceitos extraídos pelo *tagger* na transcrição das falas. O item frequência diz respeito ao número de vezes em que o conceito aparece no texto original. Como pode ser observado não consideramos a frequência =1, pois sem um melhor refinamento e melhoria do WER e sem a utilização de um processo de exclusão de *stop words* e utilização de um algoritmo de *stemmer*¹¹, acabam sendo incluídas muitas palavras que podemos considerar “lixo”.
- A “Frequência >=3” nos ensaios realizados apresenta-se como um balizador, pois da “Frequência >=2” para esta, temos um salto significativo e, depois desta, os

¹⁰ <http://www.nuance.com/naturallyspeaking/>

¹¹ <http://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/process/Stemmer.html>

valores se mantém, ou até reduzem, como pode ser observado na Tabela 6. Atribuímos este comportamento, ao fato que, com frequências baixas de ocorrência, muitas palavras que mencionamos no parágrafo anterior, categorizadas como “lixo”, acabam sendo incluídas. Mas, na medida em que aumenta a frequência, essa possibilidade diminui. Este também é outro ponto que mereceu nossa atenção nos ensaios definidos para a Etapa 5 do método de pesquisa (estudo de caso explanatório).

4.3.2 Comparando os conceitos da transcrição com os da ontologia

Durante o estudo de viabilidade foi elaborada uma ontologia do domínio do projeto que serviu para o estudo de caso exploratório. Esta ontologia foi elaborada por uma pesquisadora com experiência neste tipo de atividade e que participou do estudo de caso exploratório da Etapa 1. Ela utilizou o mesmo documento de onde foram retiradas informações para a elaboração dos roteiros das reuniões do estudo de viabilidade. Este documento apresenta a visão geral do projeto, com quarenta e oito requisitos funcionais e quatro requisitos não funcionais da aplicação.

Este ensaio consistiu em analisar separadamente os conceitos extraídos das transcrições de cada reunião e verificar a similaridade destes, com os conceitos da ontologia. O resultado deste ensaio é a indicação se os conceitos que estão na transcrição ocorrem na ontologia.

Para realizar o ensaio utilizamos a aplicação ONTrace, já comentada na seção 4.3.1, etapa 7. Esta aplicação verificou a similaridade entre os termos da transcrição e da ontologia. Na Figura 11 temos a definição de como é feita esta análise de similaridade pela ferramenta ONTrace [Nol09]. Um dos propósitos desta ferramenta é estabelecer correlações entre os artefatos utilizados em um processo de desenvolvimento de software através da ontologia do domínio.

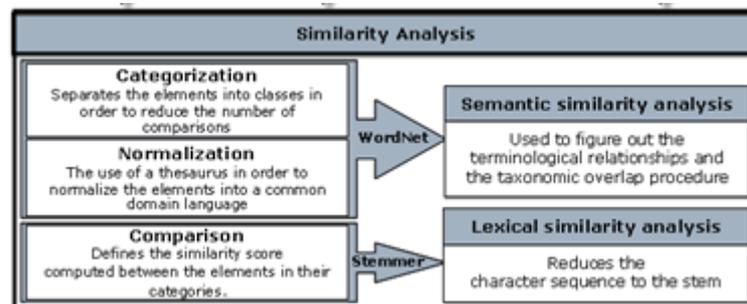


Figura 11 - Análise de similaridade realizada pelo ONTrace

Os resultados destes ensaios são apresentados na Tabela 7, onde a “Frequência” é o número de vezes em que um conceito aparece no texto transcrito da conversação; “Conceitos Transcritos” é a quantidade de conceitos detectados na transcrição para determinada frequência

e; “Identificados na ontologia” é a percentagem destes conceitos que foram encontrados similares na ontologia.

Este ensaio não foi realizado para a primeira reunião do estudo de viabilidade, somente para as três últimas, pois na primeira ainda não se tinha uma definição clara de qual domínio seria utilizado e, portanto, omitiram-se muitos aspectos do domínio do negócio no roteiro na reunião.

Tabela 7 - Resultados das transcrições comparados com a ontologia

Frequência	Reunião 1		Reunião 2		Reunião 3		Média
	Conceitos Transcritos	Identificados na ontologia	Conceitos Transcritos	Identificados na ontologia	Conceitos Transcritos	Identificados na ontologia	
>=2	57	64,3%	50	62,1%	36	60,0%	61,4%
>=3	40	68,8%	40	62,8%	21	60,9%	64,1%
>=4	24	73,3%	31	67,6%	12	61,5%	67,5%
>=5	16	70,0%	16	70,6%	10	63,6%	68,1%

Analisando os resultados, podemos observar que na medida em que aumenta a frequência em que um conceito aparece na transcrição, também aumenta a probabilidade de haver um conceito similar na ontologia. Como em todas as análises fizemos medições com a frequência até o valor 5, para uma análise mais aprofundada, foram realizados ensaios com frequências maiores no estudo de caso explanatório. Desta forma, poderemos confirmar esta tendência, indicando que conceitos mais frequentes nas reuniões são mais qualificados, sob o ponto de vista da correspondência com os conceitos da ontologia do domínio.

4.3.3 Análise do *survey* aplicado ao estudo de viabilidade

Após os oitos participantes do *survey* preencherem seus formulários, todos os dados foram tabulados e então foram realizadas as análises das informações. A Tabela 8 apresenta um exemplo de como os dados de uma reunião foram tabulados. A coluna “%” apresenta o percentual de participantes que apontaram aquele termo como relevante para aquela reunião.

Depois de tabulados, estes dados foram comparados com os dados que foram obtidos das etapas anteriores deste estudo de viabilidade, onde os termos foram extraídos dos áudios das reuniões e comparados à ontologia do domínio.

Tabela 8 - Exemplo tabulação de dados do *survey*

Termos da Ontologia	Avaliador 1	Avaliador 2	Avaliador 3	Avaliador 4	Avaliador 5	Avaliador 6	Avaliador 7	Avaliador 8	%
Brazilian_Law				x	x	x	x	x	63%
Brazilian_Tax				x	x	x	x	x	63%
Chassis									0%
COFINS					x				13%
Commodity	x	x	x	x	x	x	x	x	100%
Company									0%
CPU									0%
D3									0%
Database									0%
Datawarehouse									0%
Desktop							x		13%
FDL									0%
Fiscal				x					13%
Functional_Requirement		x	x	x	x		x	x	75%
Glovia		x					x		25%
GRM									0%
Hard_Drive									0%
IMC									0%
Interface									0%
IPI				x					13%
IT_Admin									0%
LADOMS									0%
Manufactory									0%
Memory									0%
Modem									0%
Monoprocessor									0%
Motherboard							x		13%
Multiprocessor									0%
Non-Functional_Requirement									0%
Nota_Fiscal	x	x	x		x	x	x	x	88%
...									

Várias análises foram realizadas no intuito de observar o comportamento das informações derivadas do cruzamento dos dados da geração automática dos termos das reuniões, com o *survey* onde os participantes apontaram os termos relevantes associados às reuniões. Podemos apontar como síntese destas observações, as informações apresentadas nas Tabela 9, Tabela 10 e Tabela 11.

Na Tabela 9 pode-se observar uma das análises realizadas, que foi verificar os termos apontados pelos humanos de acordo com o número de participantes que escolheram o termo e se este não foi detectado automaticamente. Na coluna “100% (todos)” estão os termos que todos os participantes do *survey* apontaram como um termo relevante para a reunião 1, neste caso somente “commodity”; na coluna “50-99% (maioria)” estão os termos que no mínimo a metade dos participantes apontaram como relevantes e; na coluna “1-49% (minoria)” estão os termos que menos da metade dos participantes apontaram como relevantes. Esta análise foi feita para as três reuniões utilizadas no *survey*. Convém destacar que os termos gerados pela extração automática

Tabela 10 - Termos relevantes gerados automaticamente na reunião 1

Termos gerados automaticamente - Reunião 1		
Brazilian_Law		
Commodity		
Company		
Functional_Requirement		
Glovia		
Interface		
IPI		
PPB		
PPB_User		
Product		
Production_Order		
Project		
Quota		
RECOF		
RECOF_User		
RecofSys		
Report		
Requirement		
Server		
User		
xxxxxxxxxxxxxx		
xxxxxxxxxxxxxx		
xxxxxxxxxxxxxx		
Total	23	
Detectado	15	65%
Não detectado	8	35%

	Termo gerado automaticamente e também detectado por humanos
xxxxxxx	Termo ocultado por estar associado à identidade das empresas participantes

Observa-se que os participantes apontaram 18 termos diferentes como relevantes para a reunião 1 (sendo 15 detectados automaticamente), e se considerarmos o que a maioria apontou, teremos então 10 termos. Enquanto que de forma automática foram gerados 23 termos. Portanto, o método automático gerou mais termos que os apontados pelos participantes. Isso foi um comportamento observado em todas as reuniões, o número de conceitos gerados automaticamente sendo maior que o apontado pelos participantes. No estudo de caso explanatório são apresentados experimentos em que variamos a frequência da ocorrência dos termos selecionados automaticamente para reduzir o número de conceitos gerados e atribuir um grau de relevância aos mesmos. Estes resultados são apresentados no capítulo 6, onde temos os resultados do estudo de caso explanatório.

Na Tabela 11 temos um resumo das três reuniões abordadas no *survey*, onde é comparada a quantidade de termos detectados por humanos e quantos destes foram detectados também de

forma automática. A coluna “% Acerto” indica o percentual de termos que foram apontados por humanos e também foram gerados automaticamente.

Tabela 11 - Resumo da análise dos termos detectados nas reuniões

Reunião 1	Qtde	Detectado automaticamente	
		Qtde	% Acerto
Termos apontados por todos humanos	1	1	100,0%
Termos apontados pela metade ou maioria dos humanos	9	9	100,0%
Termos apontados pela minoría dos humanos	8	5	62,5%
Reunião 2			
Reunião 2	Qtde	Detectado automaticamente	
		Qtde	% Acerto
Termos apontados por todos humanos	2	2	100,0%
Termos apontados pela metade ou maioria dos humanos	10	9	90,0%
Termos apontados pela minoría dos humanos	7	3	42,9%
Reunião 3			
Reunião 3	Qtde	Detectado automaticamente	
		Qtde	% Acerto
Termos apontados por todos humanos	1	1	100,0%
Termos apontados pela metade ou maioria dos humanos	5	3	60,0%
Termos apontados pela minoría dos humanos	5	1	20,0%

Observa-se na Tabela 11, que nas três reuniões, quando um termo foi apontado por todos os humanos, ele foi detectado pelo método automático. Quando ele foi apontado pela metade ou maioria dos participantes este índice de acerto já diminui um pouco, ficando na média das três reuniões em 83,8%. Já quando é apontado pela minoria dos participantes, a média de acerto nas três reuniões ficou em 41,8%.

Com os resultados do *survey* percebemos que o método que gera os termos automaticamente melhora a taxa de acerto (% Acerto) na medida em que aumenta o “consenso” entre os participantes em apontar um termo relevante. Também observamos que uma melhoria para o método seria diminuir o número de termos gerados de forma automática focando nos termos em que são apontados pela maioria dos participantes. Ensaio para ajustar este funcionamento do método foram realizados no estudo de caso explanatório, capítulo 6.

4.3.4 Considerações sobre o estudo de viabilidade

Durante o estudo de viabilidade foi possível avaliar a correlação entre os conceitos que foram extraídos de um texto, que foi usado como roteiro para a fala dos participantes, e os

conceitos que foram extraídos da transcrição da fala dos mesmos. Nestes ensaios percebemos que para os termos com frequência de ocorrência no texto maior ou igual a três, a probabilidade de serem transcritos pelo método automático proposto está acima de 90%.

Após este primeiro ensaio, foi verificada a probabilidade destes termos que foram transcritos das reuniões ocorrerem na ontologia do domínio. Constatou-se que na medida em que aumenta a frequência de ocorrência do termo na reunião, aumenta a probabilidade deste termo ser um conceito da ontologia do domínio. Um termo que ocorre três vezes na reunião tem menor probabilidade de ser um conceito da ontologia do domínio do que um termo que ocorreu dez vezes na reunião.

E por último, foi realizado um *survey*, para analisar se os termos que são gerados automaticamente são os mesmos que foram apontados pelos humanos, ao indicar os termos relevantes das reuniões. Os resultados foram satisfatórios, pois no conjunto das três reuniões mais de 70% dos termos apontados pelos humanos foram gerados automaticamente (Tabela 11). Contudo, percebeu-se a oportunidade de aprimorar o método para gerar os termos relevantes automaticamente em função da variação da frequência da ocorrência na reunião.

4.4 Considerações sobre o capítulo

Este capítulo apresentou resultados dos experimentos das três etapas iniciais que compõem o método de pesquisa para esta tese, que são:

- o reconhecimento inicial do problema;
- o estudo em profundidade sobre gestão do conhecimento em projetos de software ágeis e;
- o estudo de viabilidade.

Os resultados dos experimentos realizados nestas etapas apresentaram dados relevantes e evidências, baseados em estudos com resultados empíricos. Estes resultados serviram de ancoradouro para a proposição da metodologia de aquisição do conhecimento em reuniões de projeto de desenvolvimento de software, apresentada no capítulo 5.

O reconhecimento inicial do problema, através do estudo de caso exploratório, possibilitou-nos identificar uma área de pesquisa específica com questões em aberto onde poderíamos apresentar contribuições para o avanço da área. O Estudo em profundidade, através da revisão sistemática, permitiu-nos mapear na literatura o panorama preciso da área de pesquisa, possibilitando certificar-nos das contribuições já realizadas e das questões em aberto. O

estudo de viabilidade permitiu-nos a possibilidade de aferir, técnica e instrumentalmente, pontos que sugerimos para a proposta metodológica, que endereçam as questões em aberto e relevantes, observadas no reconhecimento inicial do problema e no estudo em profundidade da área.

Os resultados obtidos nos ensaios apontam a viabilidade, apesar de algumas restrições tecnológicas já comentadas, de que é possível extrair automaticamente conhecimento das conversações das reuniões de projeto de desenvolvimento de software. Este conhecimento é apresentado na forma de conceitos que podem ser utilizados como indexadores das reuniões.

5 Metodologia Proposta

Como etapa do método de pesquisa adotado, foi proposta uma metodologia para aquisição de conhecimento em reuniões de projetos de desenvolvimento de software, que objetiva indexar os principais conceitos associados às reuniões através da ontologia do domínio. Esta metodologia está dividida em sete etapas, elencadas a seguir e que serão detalhadas nas próximas seções:

- realizar a reunião seguindo o protocolo;
- transcrever o áudio da reunião;
- etiquetar as palavras da reunião;
- ordenar palavras da reunião por frequência de ocorrência no texto;
- tratar similaridade das palavras classificadas;
- classificar palavras da reunião de acordo com frequência no texto;
- associar termos extraídos da reunião com a ontologia do domínio.

Estas etapas foram planejadas e elaboradas com base nos ensaios realizados no estudo de viabilidade, que apresentaram resultados satisfatórios, e das contribuições do estudo de caso exploratório, assim como da revisão sistemática, conforme apresentado no Capítulo 4.

Além do estudo de viabilidade, que avaliou as atividades da metodologia proposta, também foi realizado um estudo de caso explanatório, detalhado no Capítulo 6, que apresenta ensaios para todas as etapas da metodologia, aplicado em uma empresa, que se prestou como uma instância adequada para a realização dos experimentos, de forma a avaliar todas as atividades propostas para a metodologia.

A Figura 12 apresenta uma visão geral da metodologia, elencando:

- as atividades associadas a cada etapa;
- os recursos necessários para execução das atividades e;
- os subprodutos gerados em cada etapa.



Figura 12 - Metodologia proposta

Para uma melhor compreensão da metodologia proposta, nas próximas seções são apresentadas as suas etapas de forma detalhada, conforme apresentada na Figura 12.

5.1 Etapa 1 - Realização da reunião seguindo o protocolo

Para o estudo de viabilidade, descrito na seção 3.3, foi elaborado um protocolo para as reuniões, com o intuito de facilitar o processo de transcrição das falas visando melhor extrair do arquivo de áudio os conceitos abordados nas reuniões. Este protocolo passou por definir entre os participantes da reunião o papel de um condutor, ou moderador, da reunião.

Para este estudo de viabilidade, também foram definidas “tags” ou marcadores para criar separadores de partes da reunião de acordo com os assuntos tratados, procurando facilitar a localização de em qual parte da reunião determinado conceito foi abordado. Isso serviria para evitar que fosse escutada toda uma reunião para recuperar alguma informação. Também se pode atribuir graus de relevância diferenciados para assuntos tratados em diferentes partes da reunião. No estudo apresentado em [Bru08], similarmente, são utilizadas algumas palavras chaves comuns em reuniões de times que seguem a metodologia SCRUM para facilitar a atualização de artefatos usados no projeto. Além destes pontos, algumas questões referentes ao comportamento de participantes durante as reuniões também são importantes, como pode ser observado em [Jay09] e [Ver09]. É importante que os participantes das reuniões sigam pontos de um protocolo, mas sabe-se que isso é um desafio. Por exemplo, é difícil evitar que mais de uma pessoa fale ao mesmo tempo durante uma reunião. Várias pessoas falando ao mesmo tempo podem prejudicar o desempenho do software de reconhecimento de fala. Por isso, é importante o conhecimento do protocolo pelos participantes, mas também que este altere o mínimo possível o andamento natural da reunião.

Convém destacar que existem algumas aplicações de áudio *mining* ou áudio *indexing*, que indexam um arquivo de áudio e possibilitam localizar determinada palavra no arquivo de áudio. Além de indicar se determinada palavra foi encontrada, estas aplicações também podem indicar em que posição do áudio encontra-se tal palavra, facilitando desta forma a localização de determinada informação no arquivo de áudio. Dentre as aplicações encontradas para este propósito destacamos as seguintes soluções: Nuance – Dragon¹², Aurix¹³ e Nexidia¹⁴. Estas aplicações não são gratuitas e não possuem versões para demonstração ou teste. Destacamos

¹² http://www.nuance.com/naturallyspeaking/products/sdk/sdk_audiomining.asp

¹³ <http://www.aurix.com/>

¹⁴ <http://www.nexidia.com/>

também as pesquisas feitas pela Microsoft nesta área, através do projeto MAVIS¹⁵, onde já podemos encontrar aplicações para este fim em produtos como o OneNote 2007 e Exchange Server 2010. A empresa Google também possui pesquisas nesta área, onde destacamos o projeto Gaudi, que surgiu no ano de 2008, mas que foi retirado da lista do labs.google e no momento concentra suas pesquisas de *speech recognition* em aplicações focadas para o Youtube¹⁶.

Desta forma, quando da realização do estudo de caso explanatório, para as reuniões foi mantida no protocolo a figura do condutor da reunião e uma organização na condução para evitar que mais de uma pessoa fale ao mesmo tempo. O uso de “tags” para marcar parte do texto não foi utilizado no estudo de caso, na perspectiva de usar futuramente alguma aplicação de áudio *indexing*, facilitando a localização das palavras no arquivo de áudio. Também se percebeu no estudo de viabilidade que o uso de “tags” não apresentou influência na definição da relevância de um conceito por este estar em determinada seção da reunião. Portanto, o uso de “tags” nas reuniões, que havia sido usado no estudo de viabilidade, foi retirado do protocolo.

Desta forma, o protocolo proposto para a realização das reuniões, em adequação à metodologia proposta, está baseado na definição da figura de um condutor para a reunião, que irá realizar as falas introdutórias, mencionando o tema da reunião, participantes e data/hora da realização e realizar o encerramento da mesma. Assim como irá conduzir os participantes para que eles falem em tom alto e claro, de forma a facilitar a captura do áudio pelos equipamentos e também evitar, na medida do possível, que mais de um participante fale ao mesmo tempo.

Podemos apontar como um requisito a qualidade da acústica do local da reunião e equipamentos apropriados para captura do áudio das falas para a realização das reuniões, como descrito no estudo de viabilidade, apresentado no Capítulo 4.

5.2 Etapa 2 - Transcrição do áudio da reunião

Após a realização da reunião que deve seguir o protocolo e os requisitos já apontados, o arquivo de áudio com o conteúdo da reunião será submetido a um software de reconhecimento automático de fala, para que seja realizada a transcrição da reunião. Os principais softwares de reconhecimento de fala exigem um treinamento do usuário para um melhor desempenho, portanto, inclui-se como requisito também que o condutor da reunião faça este treinamento no software, para que o reconhecimento da sua fala tenha um desempenho melhor. Os softwares

¹⁵ <http://research.microsoft.com/en-us/projects/mavis/>

¹⁶ <http://www.youtube.com/testtube>

para reconhecimento automático de fala apresentam desempenho melhor na língua inglesa, do que na língua portuguesa. Aliás, os principais produtos pesquisados, apresentados na Seção 2.4 não apresentam suporte para a língua portuguesa.

O resultado desta etapa é um arquivo com a transcrição textual da reunião, gerado pelo software de reconhecimento automático de fala.

5.3 Etapa 3 - Etiquetar as palavras da reunião

Para realizar esta etapa é necessário utilizar o arquivo texto resultante da transcrição da reunião e aplicar um software etiquetador para classificar todas as palavras do texto. Desta forma teremos como extrair os substantivos do texto, que é o tipo de palavra mais relevante para o processo de indexação dos termos das reuniões. No estudo de viabilidade e no estudo de caso explanatório, consideramos os conceitos extraídos de uma reunião todos os substantivos encontrados na transcrição e mais alguns outros tipos de palavras, como numerais e palavras estrangeiras. Os tipos de palavras retornados, de acordo com a tabela de classificação do *tagger*¹⁷ (Tabela 5) foram: NN (substantivo singular ou coletivo), NNS (substantivo plural), NNP (nome próprio singular), NNPS (nome próprio plural), CD (números cardinais) e FW (palavras estrangeiras). Na Figura 13, temos um exemplo de como é apresentado um arquivo de texto que foi etiquetado.

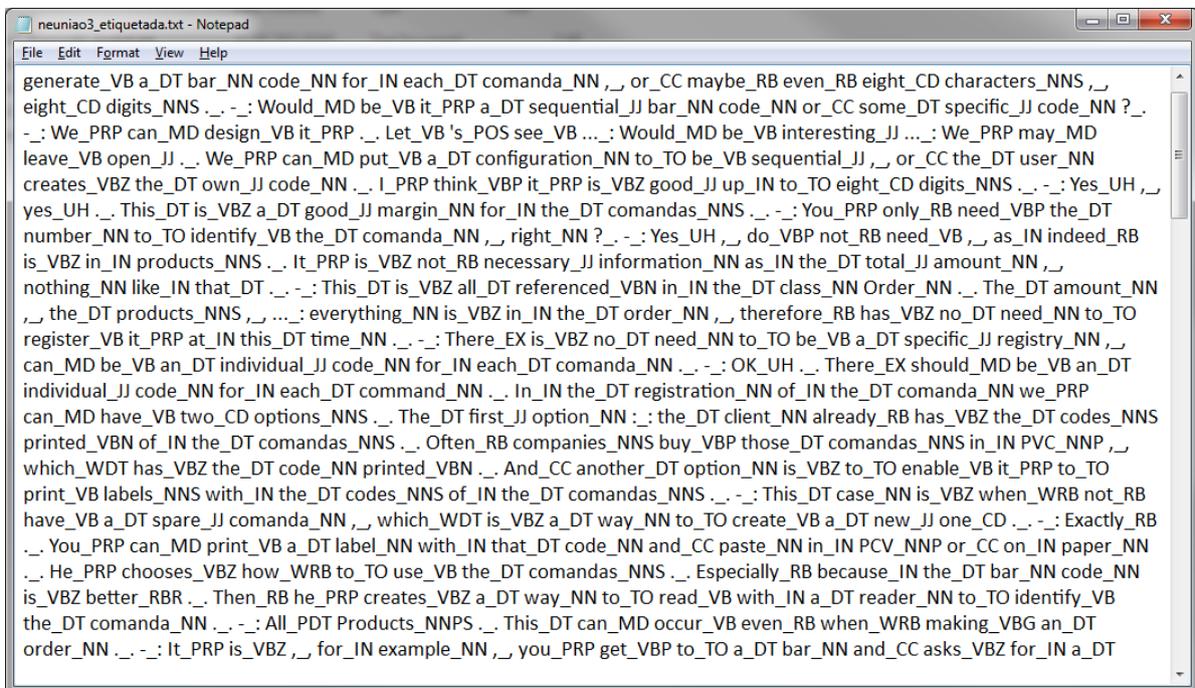


Figura 13 - Texto etiquetado

¹⁷ <http://nlp.stanford.edu/software/tagger.shtml>

A Etapa 3 é então concluída com a geração de um arquivo com o texto etiquetado, conforme a Figura 13.

5.4 Etapa 4 - Ordenar palavras da reunião por frequência de ocorrência no texto

Após etiquetar as palavras, que são selecionadas de acordo com a classificação NN, NNS, NNP, NNPS, CD e FW, é realizada uma análise no arquivo de texto etiquetado, de forma a gerar uma matriz com as palavras ordenadas pela frequência com que ocorrem no texto, como exemplificado na Tabela 12.

Tabela 12 - Palavras ordenadas pela frequência

Palavra	Frequência
comanda	45
code	15
comandas	14
Order	13
table	10
tables	10
bar	6
paper	6
data	5
example	5
number	5
option	5
product	5
Products	5
3	4
class	4
client	4
operation	4
...	

Esta etapa consiste, portanto, na aplicação de um algoritmo para a geração de uma matriz da frequência de ocorrência das palavras no texto, a partir do arquivo texto etiquetado.

5.5 Etapa 5 - Tratar similaridade das palavras classificadas

Nesta etapa utiliza-se um algoritmo do tipo *stemmer*, para agrupar palavras com o mesmo núcleo, evitando, por exemplo, que na lista de palavras apareçam separadamente “comanda” e “comandas”, “table” e “tables”, sendo que nestes casos as palavras são agrupadas na sua forma singular. Por razões gramaticais, os documentos usam diferentes formas de uma palavra, como podemos observar em “democracy”, “democratic” e “democratization”.

O objetivo de um algoritmo de *stemmer* é reduzir as formas flexionadas a um mesmo núcleo chamado *stem*. A pesquisa sobre ferramentas automáticas para reduzir uma palavra ao seu *stem* remonta a 1960, quando derivações iniciais foram propostas e experiências foram relatadas para avaliar seu impacto sobre a eficácia da recuperação do núcleo das palavras [Mel07].

Nós utilizamos em nosso projeto o algoritmo mais comum para a língua Inglesa, e que tem demonstrado ser muito eficaz empiricamente - o algoritmo de *Porter* [Por06]. No site do *NLP Group - The Stanford Natural Language Processing Group*¹⁸ pode-se realizar teste com o algoritmo *stemmer*, são apresentados exemplos de uso e pode-se baixar o software.

Pode-se ainda utilizar uma lista de *stop words* baseadas em ontologias sobre engenharia de software para refinar a lista dos conceitos válidos extraídos das reuniões [Cal06], [Won08]. Por exemplo, durante uma reunião de um projeto de desenvolvimento de software muitas vezes a palavra “software” será mencionada, mas não é de interesse que seja apontada como um conceito relevante. Por isso, a retirada de conceitos do domínio engenharia de software, é uma forma interessante de refinar a lista de conceitos relevantes das reuniões.

Ao final desta etapa é gerada a matriz de frequência de ocorrência das palavras no texto, refinada pelo uso de um algoritmo do tipo *stemmer*. A Tabela 13 ilustra como ficaria a matriz de frequência das palavras após ter sido utilizado o algoritmo *stemmer*.

Tabela 13 - Palavras ordenadas após aplicar *stemmer*

Palavra	Frequência		Palavra	Frequência
comanda	45	➔	comanda	59
code	15		code	15
comandas	14		Order	13
Order	13		table	20
table	10		bar	6
tables	10		paper	6
bar	6		data	5
paper	6		example	5
data	5		number	5
example	5		option	5
number	5		product	10
option	5		3	4
product	5		class	4
Products	5		client	4
3	4		operation	4
class	4			
client	4			
operation	4			
...				

5.6 Etapa 6 - Classificar palavras da reunião de acordo com frequência no texto

Durante o estudo de viabilidade realizou-se ensaios em que os conceitos retirados das reuniões eram comparados com os conceitos presentes na ontologia do domínio. Podemos destacar dentre os resultados apresentados, que os termos que possuem frequência inferior a três

¹⁸ <http://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html>

deveriam ser descartados e que, na medida em que aumenta a frequência de ocorrência de um termo no texto, maior é a probabilidade deste termo ser um conceito da ontologia do domínio. Os resultados detalhados destes ensaios estão na seção 4.3 Estudo de viabilidade.

Também percebemos no estudo de viabilidade, a necessidade de ensaios que apontassem uma heurística que indicasse mais precisamente, em que frequência, os termos apontados tornar-se-iam relevantes para serem usados como indexadores das reuniões. Estes ensaios foram propostos para serem realizados no estudo de caso explanatório, apresentado no Capítulo 6.

Para propor uma forma de calcular a faixa de frequência dos termos a serem selecionados, analisamos a curva de distribuição normal [Oat06]. Na Figura 14, percebemos que 68,2% dos valores estão entre a média e o valor de uma vez o desvio padrão (área azul), já 95,4% dos valores estão entre a média e duas vezes o valor do desvio padrão (área verde) e 99,7% dos valores estão entre a média e três vezes o valor do desvio padrão (área cinza). Considerando que nos ensaios anteriores foi percebido que quanto maior a frequência de um termo na reunião, maior a probabilidade deste estar associado à ontologia do domínio, procurou-se então uma forma de selecionarmos os termos que estiverem mais próximos à “cauda” da curva de distribuição normal.

Desta forma, durante o estudo de caso explanatório, os experimentos indicaram que palavras na faixa entre o valor do desvio padrão e duas vezes o valor do desvio padrão da frequência de ocorrência das palavras no texto, é que deveriam ser apontadas como termos relevantes para indexar uma reunião.

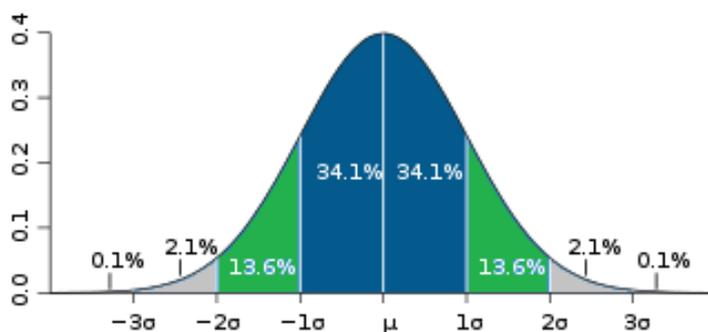


Figura 14 - Distribuição Normal [Fonte: Wikipedia]

Exemplificando, para melhor demonstrar a explanação anterior, tomemos o caso da reunião utilizada como exemplo na Tabela 12, onde foram extraídas 112 palavras classificadas como NN, NNS, NNP, NNPS, CD e FW e a frequência média em que elas ocorreram no texto foi 2,72; o desvio padrão foi 5,93 e; o valor de duas vezes o desvio padrão foi 11,86 (Figura 15). Então se definiu duas faixas de termos relevantes: (i) a primeira faixa começa no valor da soma da média da frequência mais o valor do desvio padrão, neste caso, “2,72 + 5,93”, que é igual a 8,65, que

arredondando ficou 9; (ii) a segunda faixa começa no valor da soma da média da frequência mais duas vezes o valor do desvio padrão, neste caso, “ $2,72 + 11,86$ ”, que é igual a 14,58, que arredondando ficou 15.

Desta forma, para este caso definimos então duas faixas: uma que vai da frequência de 9 a 14, e outra que vai de valores iguais e acima de 15. Na Figura 15 temos uma ilustração que exemplifica como são definidas estas duas faixas, que concentram os termos mais relevantes de uma das reuniões do estudo de caso explanatório.

Quantidade de palavras selecionadas na reunião	112		
Média de frequência das palavras selecionadas	2,72		
Desvio Padrão	5,93		
Desvio Padrão x 2	11,86		
Faixa 2 \geq (Média + Desvio Padrão x 2)	15		
Faixa 1 \geq (Média + Desvio padrão)	9		

Reunião 3		
Palavra	Freq.	
comanda	58	Faixa 2
table	20	
code	15	
Order	13	Faixa 1
product	10	
client	7	
bar	6	
paper	6	
data	5	
example	5	
number	5	
option	5	
3	4	
class	4	
operation	4	
1	3	
5	3	
quantity	3	
attributes	3	
command	3	
consumption	3	
eight	3	
reader	3	
registration	3	
scanner	3	
type	3	

Figura 15- Exemplo de classificação das palavras em uma reunião

A escolha pelo uso do desvio padrão no cálculo para apontarmos a faixa de frequência dos termos relevantes manifestou-se adequada, pois desta forma conseguimos perceber se nas reuniões alguns poucos termos aparecem com uma frequência bem acima dos demais, que é um indicativo para ser um termo indexador da reunião. Em reuniões que isso acontece o desvio padrão tende a aumentar. Já em reuniões onde não temos poucos termos em que a frequência destoa dos demais, o desvio padrão tende a diminuir e isso aumentará o número de termos na faixa de frequência dos termos relevantes. Na Tabela 14 podemos observar este comportamento através do exemplo entre duas reuniões.

Tabela 14 - Análise do desvio padrão

Reunião 3		Reunião 4	
Quantidade de Termos	112	Quantidade de Termos	129
Média de frequência	2,70	Média de frequência	2,36
Valor Desvio Padrão	5,89	Valor Desvio Padrão	2,64
Qtde termos selecionados na Faixa 2	3	Qtde termos selecionados na Faixa 2	6
Qtde termos selecionados na Faixa 1	2	Qtde termos selecionados na Faixa 1	9

Percebemos nesta análise entre estas duas reuniões, que na reunião 3, onde o desvio padrão foi maior, a quantidade de termos selecionados foi menor, já na reunião 4, onde o desvio padrão foi menor, a quantidade de termos selecionados foi maior. Na Figura 16 podemos perceber esta tendência no comparativo entre as reuniões do estudo de caso explanatório.

Essa análise utilizando a quantidade de termos em função do desvio padrão poderia auxiliar na indicação de que uma reunião é mais focada em determinados tópicos, ou se ela possui um perfil mais abrangente, considerando a discussão de tópicos associados ao projeto no qual a reunião está inserida. Quanto menor a quantidade de termos usados como indexadores de uma reunião, mais focada ela é; quanto maior a quantidade de termos; menos focada ela é.

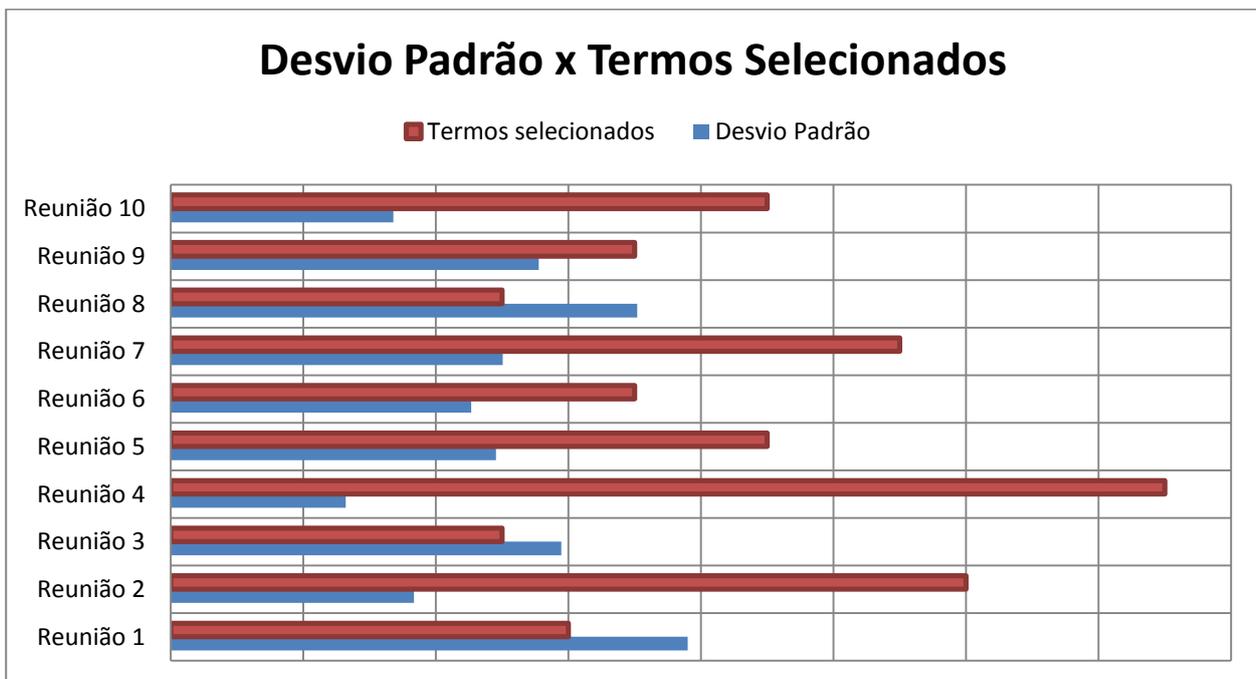


Figura 16 - Desvio padrão comparado ao número de termos selecionados

Portanto, ao final desta etapa da metodologia, que prevê classificar os termos abordados na reunião, o resultado é uma matriz com os conceitos mais “relevantes” e a frequência em que os mesmos ocorreram na reunião. Nesta matriz são apontados, com base no cálculo do desvio padrão, os termos que aparecem na “Faixa 1” e “Faixa 2”, conforme mostrado na Figura 15. A “Faixa 1” representa os conceitos que possuem uma relação de média relevância com a reunião, pois estão na faixa entre uma vez o desvio padrão e duas vezes o desvio padrão, enquanto que a “Faixa 2” representa os conceitos com uma relação de alta relevância, pois estão na faixa acima de duas vezes o desvio padrão.

5.7 Etapa 7 - Associar conceitos extraídos da reunião com ontologia do domínio

Após elencados em uma matriz, os termos mais relevantes associados à reunião, torna-se possível então, associar estes com os conceitos da ontologia do domínio do projeto. Para fazer esta associação é criada uma classe na ontologia do domínio que represente as informações de identificação de uma reunião e os termos relevantes associados a esta. A metodologia proposta não prevê a elaboração de uma ontologia e sim, que ela já tenha sido previamente concebida.

A Figura 17 apresenta um exemplo de como são estruturadas as classe “Meeting” e “Tag”, que representam os conceitos associados às reuniões. Como o estudo de viabilidade e os estudos de caso tiveram as reuniões gravadas na língua inglesa, a ontologia e os exemplos também serão apresentados nesta língua. Desta forma, a classe “Meeting” está estruturada da seguinte forma:

- um *datatype property* “Date_time”, onde será inserido a data e hora da reunião;
- um *object property* “Have_many”, que associa a classe “Meeting” com a classe “Tag”.

A classe “Tag” possui os seguintes elementos:

- um *datatype property* “Term”, onde será inserido um termo extraído da reunião;
- um *datatype property* “Frequency”, onde será inserido a frequência no texto deste termo extraído da reunião;
- um *object property* “References_Strongly”, usado para associar um elemento classificado na “Faixa 2” (Figura 15) da classe “Tags” com um elemento da classe “Restaurant”;
- um *object property* “References_Weakly”, usado para associar um elemento classificado na “Faixa 1” (Figura 15) da classe “Tags” com um elemento da classe “Restaurant”.

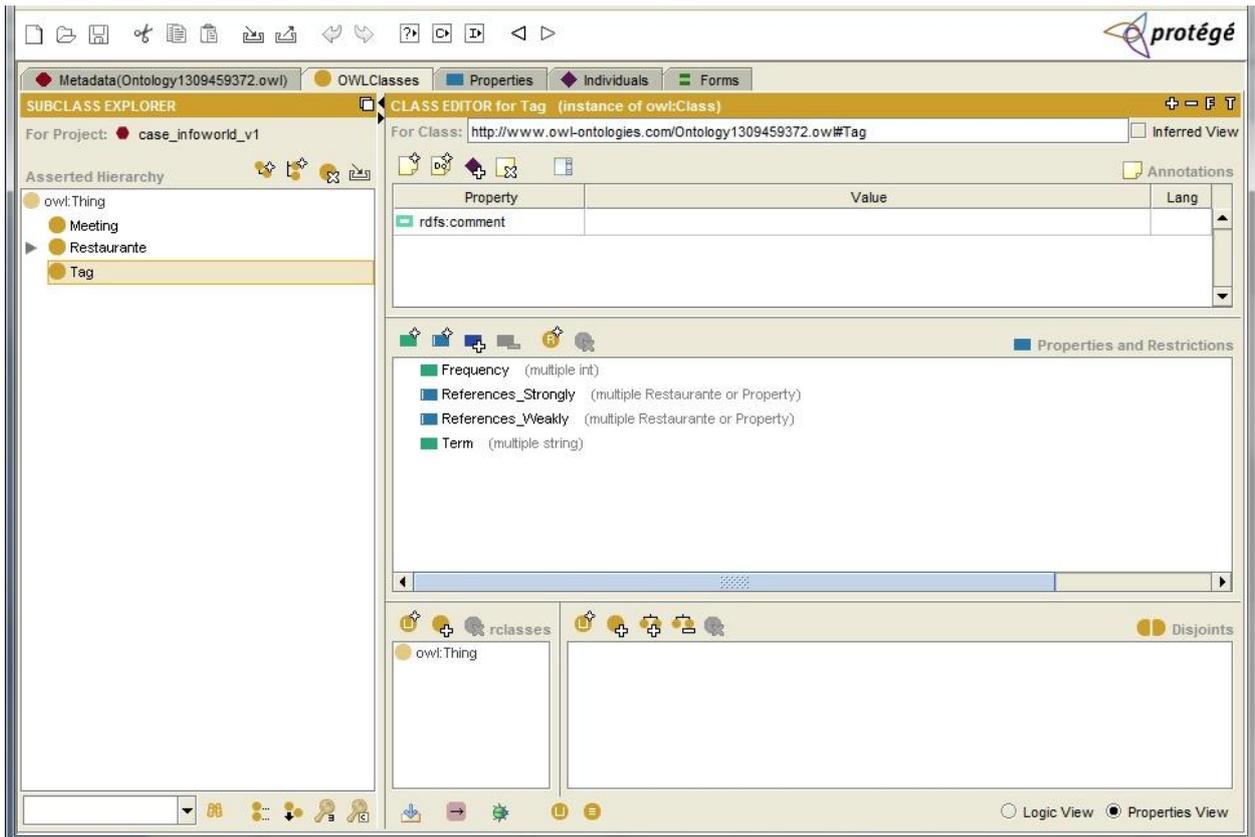


Figura 17- Ontologia de reunião

Depois de criada a estrutura de “Meeting” na ontologia do domínio pode-se incluir os indivíduos (instâncias) desta classe, como pode ser observado na Figura 18, onde foram inseridas dez reuniões. Para cada reunião foi informado data e horário de início e os termos extraídos associados a esta reunião. Temos no exemplo da Figura 18 as informações da “meeting_3”, que são a data e hora da reunião, inseridas através do *datatype property* “Date_time” e os termos associados a ela são indivíduos da classe “Tag” associados através do *object property* “Have_many”.

Para associar uma reunião aos seus termos, temos de primeiramente cadastrar estes termos através de instâncias da classe “Tag”, como está ilustrado na Figura 19. Neste exemplo, temos uma instância do termo “Comanda” que ocorreu em “Meeting_3”, onde foram inseridos os *datatype properties* “Frequency = 58” e “Term = comanda”. Depois de inseridos os termos da reunião, como instâncias da classe “Tag”, estes podem ser associados a uma instância da classe “Meeting” através do *object property* “Have_many”.

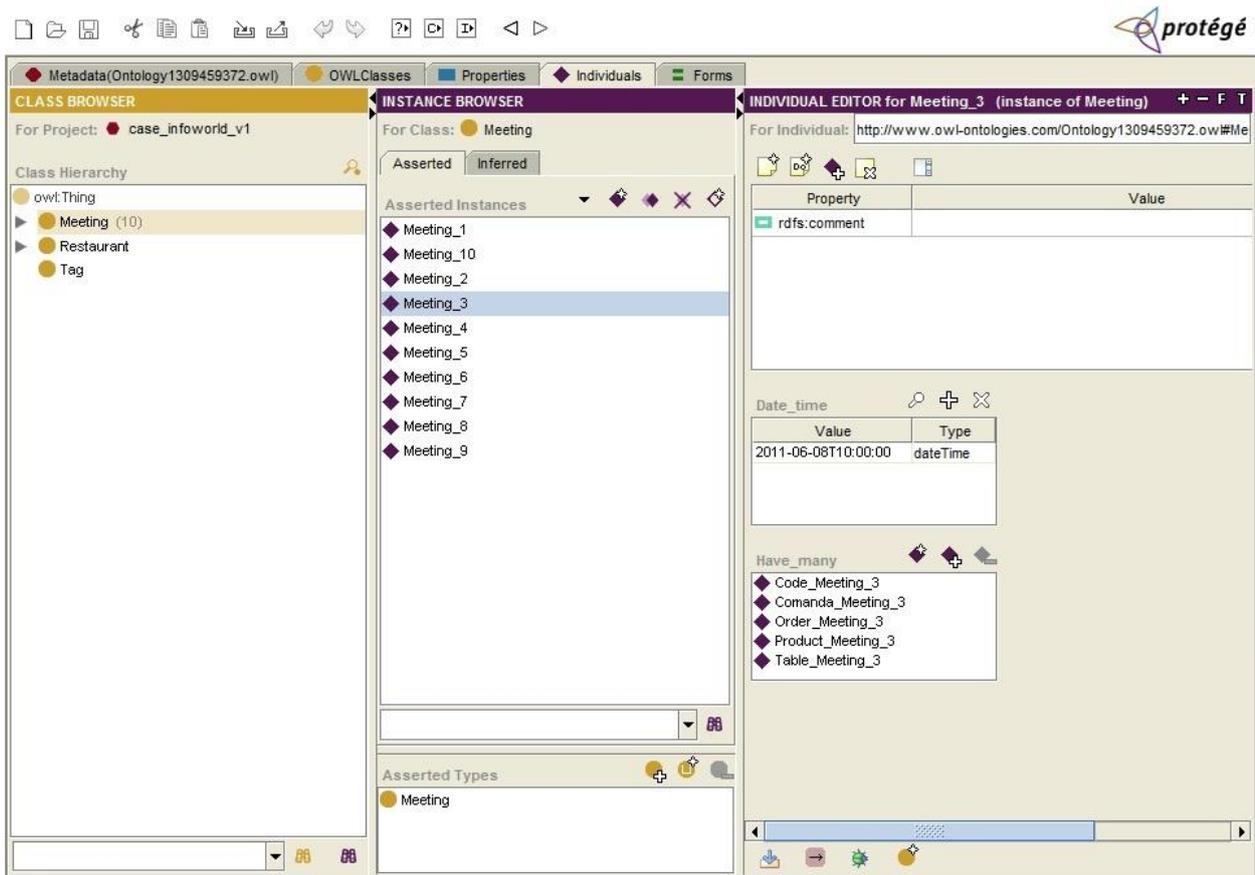


Figura 18- Indivíduos da classe “Meeting”

No exemplo apresentado, que pode ser observado através da Figura 18 e da Figura 19, a “Meeting_3” tem seus termos “Code, Comanda, Order, Product e Table” associados através do *object property* “Have_many”.

Como último passo para que os conceitos extraídos das reuniões estejam associados com a ontologia do domínio, os conceitos equivalentes, neste caso, de “Restaurant” e de “Tag” devem ser associados. Contudo, para cada conceito associado, deve ser informado se ele está classificado na Faixa 1 ou 2 (conforme o desvio padrão), de acordo como foi descrito na Etapa 6 desta metodologia.

Tomemos como exemplo novamente a Figura 19, onde o conceito “comanda” da “Meeting_3” foi associado à instância da classe “Comanda” (subclasse de “Restaurant”) através do *object property* “References_Strongly”. Esta associação indica que o conceito “comanda” extraído da “Meeting_3”, por ser um termo classificado na Faixa 2 desta reunião possui uma “forte referência” ao conceito “Comanda” da classe “Restaurant”.

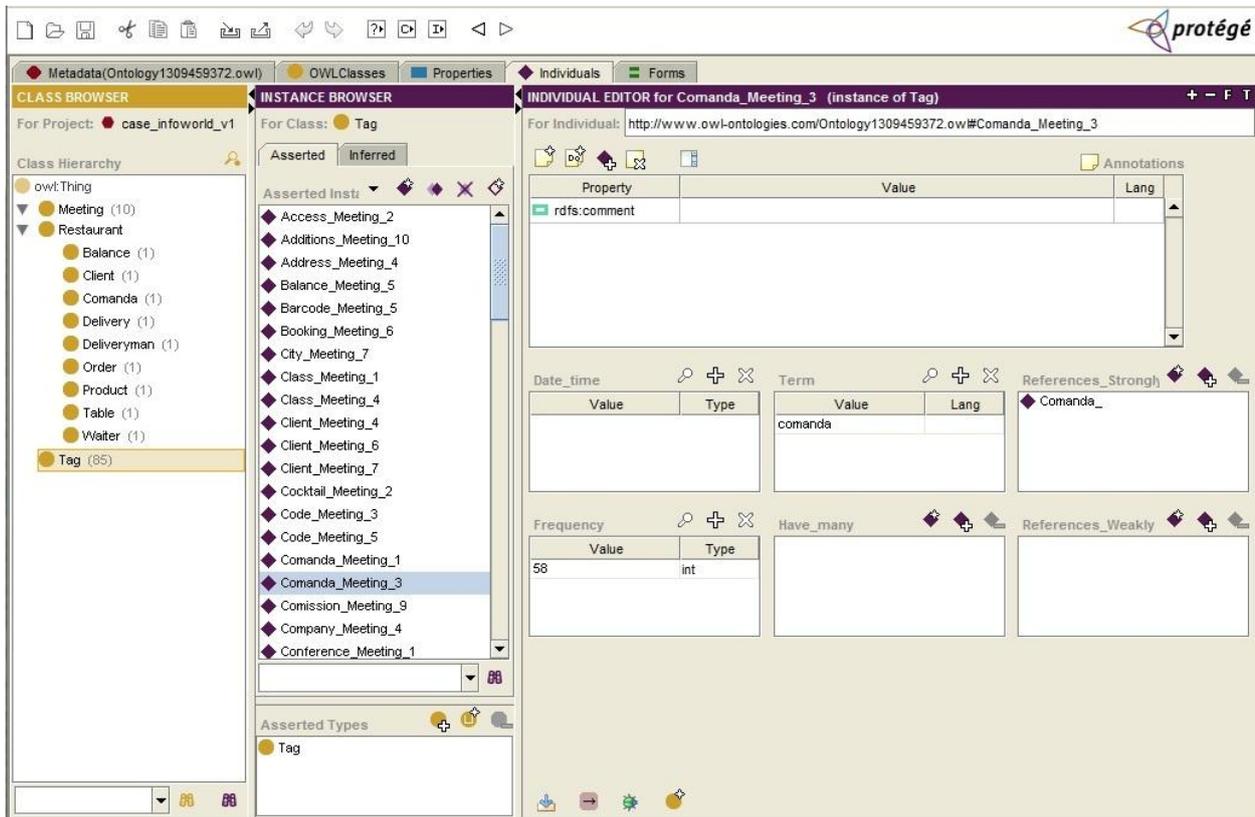


Figura 19 - Indivíduos da classe "Tag"

Desta forma, para concretizar esta última etapa da metodologia, todos os conceitos das reuniões que possuem conceitos correspondentes na ontologia do domínio, devem ser associados a estes em uma das seguintes formas:

- usando o *object property* "References_Strongly" caso este conceito da reunião tenha sido classificado na Faixa 2 de frequência (média + desvio padrão x2), ou;
- usando o *object property* "References_Weakly" caso este conceito da reunião tenha sido classificado na Faixa 1 de frequência (média + desvio padrão).

Convém destacar que a metodologia proposta não apresenta como usar o relacionamento entre os conceitos para realização de inferências. Mas este trabalho faz parte de um projeto de pesquisa com um escopo maior, onde outros projetos proporcionam mecanismos para que o conhecimento e os relacionamentos entre os termos da ontologia sejam usados para executar inferências.

Na Figura 20 é apresentado um diagrama que proporciona uma visão geral do processo que contempla as etapas da metodologia, onde podemos observar os participantes, as tarefas, os fluxos e os artefatos que fazem parte do processo. No diagrama apresentado, temos três participantes que executam as atividades: a equipe, o condutor e o engenheiro de conhecimento.

O papel da equipe é participar da reunião, na qual serão gravados os diálogos, e ela pode representar os desenvolvedores, analistas, testadores, clientes, fornecedores, enfim os *stakeholders* de uma forma geral.

O condutor é um membro da equipe e o seu papel é conduzir a reunião, dentro de um protocolo estabelecido, de forma a otimizar a qualidade do áudio a ser gerado com as conversações. Isso envolve operar os equipamentos para gravação dos arquivos de áudio e conduzir os participantes de forma a melhorar a qualidade do áudio a ser gerado, evitando que vários participantes falem ao mesmo tempo e que se manifestem de forma adequada para a gravação dos diálogos.

O engenheiro de conhecimento realiza várias atividades dentro do processo, que compreendem todas as tarefas que vão, desde a gravação do áudio gerado nas reuniões, até a associação dos termos extraídos das reuniões com a ontologia do domínio. O engenheiro de conhecimento manipula e trata todas as ferramentas e artefatos definidos no processo.

O diagrama apresentado na Figura 20, em relação ao apresentado na Figura 12, proporciona uma perspectiva mais voltada para uma visão das tarefas a serem realizadas pelos participantes. O diagrama da Figura 20, também oferece uma percepção do fluxo das atividades e os artefatos gerados e utilizados durante o processo.

Convém destacar que o uso de uma ontologia do domínio é um requisito para a metodologia apresentada, e que este ainda não é um artefato amplamente usado em projetos de desenvolvimento de software, principalmente em projetos que seguem metodologias ágeis. Salientamos novamente, que a metodologia aqui apresentada é para projetos de desenvolvimento de software, independente de serem adotadas práticas tradicionais ou ágeis.

Para uma organização adotar a metodologia proposta é necessário, portanto, que a mesma faça de ontologias de domínio dos seus projetos, de um software de reconhecimento automático de fala; de um software etiquetador (*tagger*) e que utilize dispositivos de gravação de áudio adequados para as reuniões. Também é necessário que alguém da equipe desempenhe o papel de condutor nas reuniões e de um engenheiro do conhecimento, para a realização das tarefas apontadas na Figura 20.

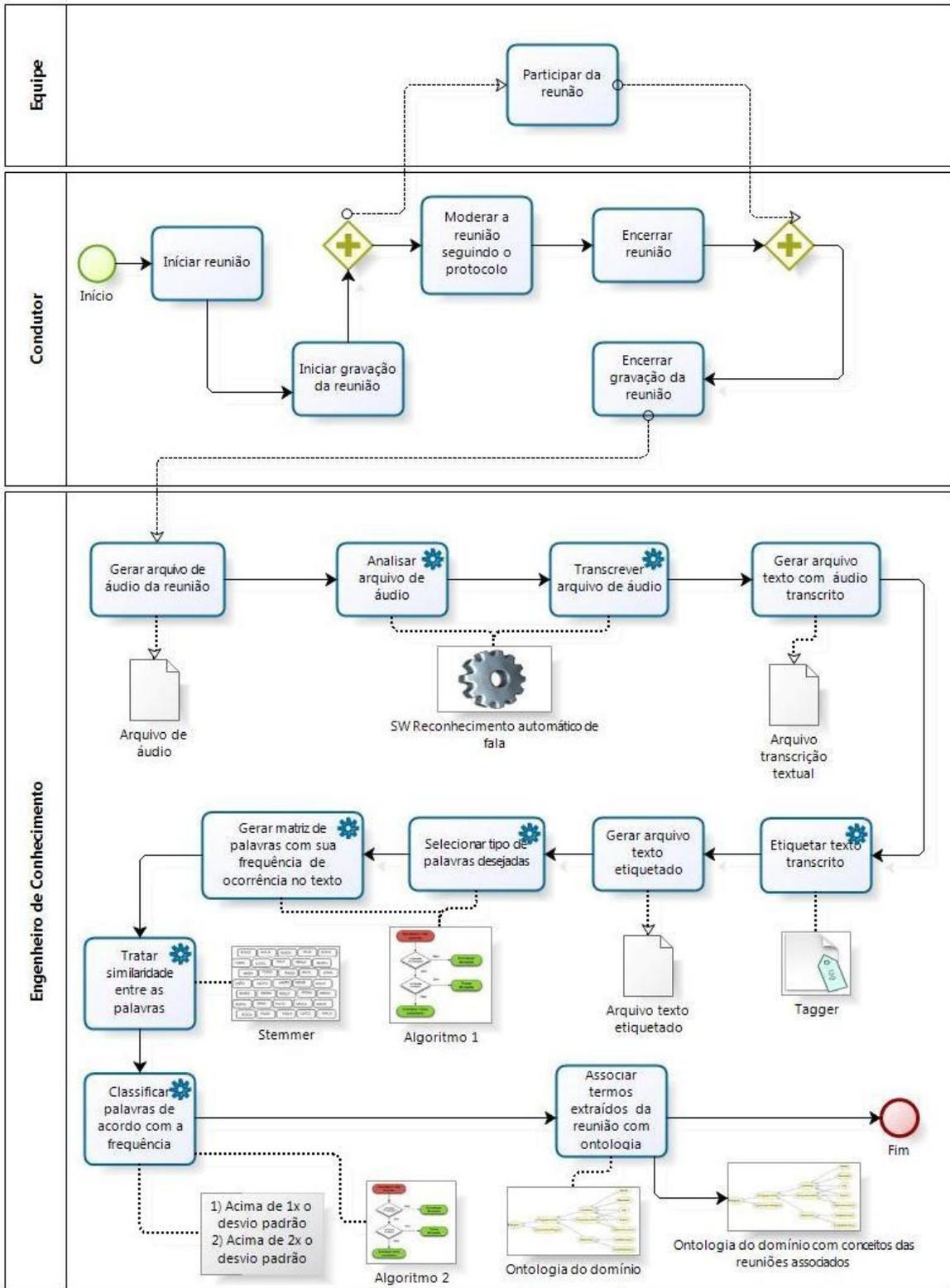


Figura 20 - Metodologia proposta utilizando notação BPMN

5.8 Considerações sobre o capítulo

Realizando então todas estas etapas da metodologia proposta, temos uma estrutura de indexação dos termos mais relevantes das reuniões associados à ontologia do domínio, que permitirá consultas para verificar, por exemplo, em que reuniões, um ou mais conceitos da

ontologia foram tratados, ou quais conceitos da ontologia foram tratados em uma reunião e, até associar as reuniões, a outros artefatos através da ontologia do domínio. No estudo de caso explanatório (Capítulo 6) é apresentado um protótipo que permite a realização de consultas à ontologia.

Convém destacar que a metodologia apresentada neste trabalho segue em linhas gerais como pressuposto conceitual e teórico para um modelo de gestão do conhecimento, o modelo de cinco fases para criação do conhecimento proposto por Nonaka e Takeuchi [Non95]:

- compartilhamento do conhecimento tácito: o conhecimento tácito de uma organização está nos indivíduos que a compõe. E esse conhecimento individual deve ser compartilhado com a organização;
- criação de conceitos: o conhecimento tácito compartilhado anteriormente deve ser transformado em conhecimento explícito;
- justificação de conceitos: a organização aplica um filtro sobre os conceitos criados, a fim de verificar se “os conceitos recém-criados valem realmente a pena para a organização e para a sociedade”;
- construção de um arquétipo: o conhecimento justificado é transformado em algo tangível ou concreto, ou seja, construir algo com o conhecimento justificado, que pode ser um protótipo;
- difusão interativa do conhecimento: o conhecimento criado, justificado e aplicado se transforma em base para a criação de novos conhecimentos.

No próximo capítulo será apresentado um estudo de caso explanatório, onde foi possível aplicar a metodologia proposta e avaliar seus resultados através de um *survey*.

6 Resultados do estudo de caso explanatório

Entre junho e julho de 2011 foi realizado um estudo de caso explanatório, conforme previsto no método de pesquisa, em uma empresa desenvolvedora de software que possui sua sede instalada na cidade de Torres - RS. Para este estudo de caso foi realizado o acompanhamento de um projeto de atualização de um módulo do principal produto produzido pela empresa, que é uma aplicação para automação comercial, que atende o comércio varejista em geral.

Para apresentar os tópicos que compõem o estudo de caso, o capítulo será dividido nas seguintes seções: (i) o contexto da organização; (ii) as atividades que foram analisadas durante o desenvolvimento do projeto; (iii) os instrumentos utilizados para coleta de dados; (iv) as considerações obtidas a partir da análise de dados e; (v) as discussões sobre os resultados obtidos.

6.1 Contexto da Organização

A empresa onde foi desenvolvido o estudo de caso possui dezesseis anos de atuação no mercado, conta com vinte funcionários na área de desenvolvimento e suporte aos seus produtos e possui clientes espalhados em quase todos os estados brasileiros. O carro chefe da empresa é uma aplicação para automação comercial que atende os seguintes seguimentos: supermercados, comércio em geral, confecção e calçados, materiais de construção, ópticas e relojarias, livrarias e papelarias, *fastfoods*, bares e restaurantes, autopeças e serviços, e postos de combustíveis.

A empresa nos últimos dois anos vem implantando gradualmente práticas oriundas de metodologias ágeis ao seu processo de desenvolvimento de software. A gerente de projetos da empresa define que “o processo atual é híbrido, partindo de um modelo em cascata e adotando gradualmente práticas ágeis, dentro de um processo de amadurecimento da equipe e colaboradores”.

O estudo de caso foi focado em uma alteração na aplicação para bares e restaurantes, onde alguns relatórios e regras de negócio foram atualizados e foi implantado um módulo para conexão com uma balança para pesagem de produtos. O sistema não fazia a leitura do peso e valor de um produto diretamente da balança, era necessário anotar as informações em uma comanda e depois digitá-los em um terminal. Com a alteração proposta, será possível na aplicação para bares e restaurantes, a leitura dos dados diretamente da balança. Estas modificações no sistema são o resultado da análise de um conjunto de melhorias apontadas pela base de clientes que utiliza a aplicação, desta forma não foi desenvolvido para um cliente em específico.

O projeto teve a duração de vinte e um dias e quatro funcionários participaram das atividades, sendo um analista e três desenvolvedores. As reuniões que foram o objeto do estudo de caso, contaram com a participação destes quatro funcionários, sendo que nem sempre todos participaram das reuniões.

6.2 Atividades analisadas

Foram utilizadas para o nosso estudo de caso as reuniões realizadas pela equipe de análise e desenvolvimento responsável pelas alterações no sistema para bares e restaurantes. As reuniões eram realizadas com o intuito de que a equipe equalizasse o entendimento dos requisitos associados às mudanças propostas e informasse o andamento das tarefas. Estas reuniões ocorreram dez vezes durante a duração do projeto e foram realizadas em português.

As reuniões foram realizadas nas dependências da sede da empresa, em uma sala reservada para reuniões. O analista de sistemas do projeto conduziu as reuniões que foram gravadas em um gravador digital portátil. Não foi requisito uma excelente qualidade das condições acústicas para a gravação das reuniões, pois estas tiveram de ser transcritas para depois serem gravadas na língua inglesa. O autor desta tese acompanhou a primeira reunião, orientando e evitando principalmente que mais de uma pessoa falasse ao mesmo tempo e também para verificar se a qualidade do áudio, considerando os equipamentos e as condições acústicas, seria apropriada para a posterior transcrição.

Após a gravação das dez reuniões, cada uma delas foi, de forma manual, transcrita em arquivo texto contendo todas as falas realizadas durante o andamento da reunião. Depois, baseado no texto da transcrição, todas as reuniões foram gravadas na língua inglesa, por uma professora de inglês, que também participou de estudo de viabilidade. Esta tradução realizada para que o áudio fosse submetido ao software de reconhecimento automático de fala pode gerar um viés, sendo uma ameaça ao processo.

Para a gravação do áudio durante esta etapa foi utilizado um microfone direcional sensível com sistema eliminador de ruídos. Da mesma forma como foi realizada no estudo de viabilidade, a qualidade do áudio gerado foi averiguada pelo próprio software de reconhecimento automático de fala utilizado.

A gravação das reuniões em inglês gerou um arquivo de áudio em formato “mp3” para cada uma das dez reuniões, que foram submetidos ao processo descrito pela metodologia proposta, conforme apresentado no capítulo 5, e melhor visualizado na Figura 12. O resultado da

aplicação da metodologia resultou em uma série de informações extraídas das reuniões, que serão analisados na seção 6.4 Análise dos dados.

Antes da realização das reuniões, o autor desta tese juntamente com o analista responsável pelo projeto, elaborou uma ontologia do domínio do projeto, apresentado na Figura 21, que serviu para o estudo de caso.

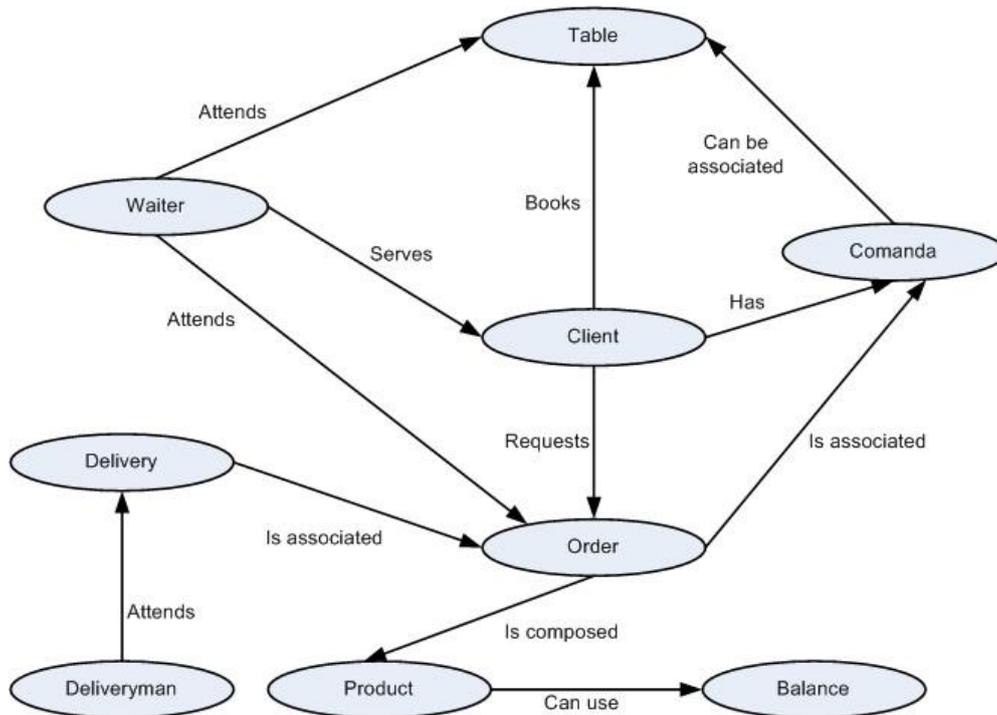


Figura 21 - Ontologia do domínio abordado no estudo de caso

Durante o estudo de caso foi elaborada e testada uma heurística para a classificação dos termos selecionados em duas faixas, ou níveis, de acordo com a frequência em que o termo ocorre na reunião. Estas duas faixas estão baseadas no cálculo do desvio padrão da frequência dos termos na reunião e indicam um grau de associação destes termos com a reunião através do uso da ontologia do domínio. Todo o processo para a definição destas duas faixas e como os termos são selecionados foram detalhadamente explicados na seção 5.6 Etapa 6 - Classificar palavras da reunião de acordo com frequência no texto. Nas próximas seções também são apresentadas algumas discussões sobre o método de classificação dos termos das reuniões.

6.3 Instrumento de pesquisa

Neste estudo de caso, para as dez reuniões realizadas foi aplicada a metodologia proposta neste trabalho, que de forma automática, a partir do áudio das reuniões gerou os conceitos mais relevantes associados às reuniões. A realização de um *survey*, com os participantes das reuniões,

foi uma forma encontrada de avaliar se os conceitos mais relevantes gerados através da nossa metodologia estariam de acordo com a percepção humana, de quais seriam os conceitos mais relevantes.

Para a realização do *survey* foi elaborado um instrumento, que foi submetido à avaliação dos participantes das reuniões de projeto do estudo de caso explanatório. De uma forma geral, este instrumento apresentou aos respondentes uma lista de termos, que são os conceitos da ontologia do domínio, para que estes analisassem as reuniões e apontassem os termos que seriam relevantes para os tópicos tratados em cada reunião. O *survey* foi composto pelos seguintes passos: (i) o autor desta tese fez uma reunião individual com cada respondente do *survey* explicando o contexto da realização do mesmo; (ii) cada respondente revisou o conteúdo de cada uma das reuniões realizadas durante o projeto, sendo que foi disponibilizado no instrumento a transcrição completa de todas as reuniões; (iii) cada respondente apontou na planilha de avaliação os termos relevantes de cada reunião. No Apêndice E é apresentado o instrumento utilizado neste *survey*, com mais detalhes sobre a aplicação do mesmo.

6.4 Análise dos dados

A análise dos dados gerados no estudo de caso explanatório é composta de duas etapas: (i) análise dos termos gerados de forma automática para refinamento do método proposto e; (ii) comparação entre os dados que foram gerados de forma automática e os dados obtidos através do *survey*.

Em um primeiro momento, foram realizadas análises sobre os resultados gerados com a aplicação da metodologia proposta às reuniões que compuseram o estudo de caso, partindo da observação dos termos que foram gerados de forma automática. De acordo com os resultados do estudo de viabilidade, a “linha de corte” para selecionar os termos relevantes era a frequência de ocorrência na reunião sendo igual ou superior a três (seção 4.3 Estudo de viabilidade). Ao aplicar este método às reuniões do estudo de caso explanatório percebemos que a quantidade de termos selecionada como “relevante” era mais extensa do que nos ensaios do estudo de viabilidade. Como o objetivo era capturar os termos mais relevantes que poderiam servir de indexadores para uma reunião, o foco era apontar um número pequeno de termos, mas que realmente servissem para identificar e diferenciar os assuntos tratados nas reuniões. Como apontado nos *surveys* realizados (estudo de viabilidade e estudo de caso explanatório) estes termos relevantes geralmente são os termos que mais ocorrem na reunião.

Depois de alguns ensaios para tentar selecionar os termos que se diferenciavam dos demais pela frequência em que ocorriam no texto, surgiu uma proposta de usar o cálculo do desvio padrão para separar da lista de termos de uma reunião, aqueles que se destacam dos demais pela maior frequência. A seção 5.6 Etapa 6 - Classificar palavras da reunião de acordo com frequência no texto, apresentou como foi elaborada a heurística para o uso do desvio padrão na classificação dos termos relevantes de uma reunião.

Para exemplificar a situação percebida no estudo de caso explanatório, podemos observar a Figura 22 onde temos a relação dos termos de cada uma das dez reuniões. Nesta figura foi feito um corte para aparecerem os termos com frequência igual ou superior a três, pois as reuniões tinham em média cem termos.

Podemos verificar, por exemplo, que na “Reunião 1” se mantivéssemos a linha de corte na frequência igual ou maior a três, seriam selecionados como termos relevantes para esta reunião trinta e uma palavras. Ao aplicarmos o método proposto, conforme descrito na metodologia (seção 5.6), usando os valores de desvio padrão das frequências, teremos então seis termos selecionados como relevantes. O método proposto ainda divide os termos selecionados em duas faixas:

- Faixa 1: começa no valor da soma da média da frequência mais o valor do desvio padrão, que neste caso seria $3,69$ (média) + $7,80$ (desvio padrão) = $11,49$, arredondado para 11. Isto indica que, a Faixa 1 começaria nos termos com frequência igual ou maior a 11 e iria até a Faixa 2, descrita a seguir.
- Faixa 2: começa no valor da soma da média da frequência mais duas vezes o valor do desvio padrão, que neste caso seria $3,69$ (média) + $15,60$ (duas vezes o desvio padrão) = $19,29$, arredondado para 19. Neste caso, a Faixa 2 começaria nos termos com frequência igual ou maior a 19.

Convém destacar que neste exemplo da Reunião 1 (Figura 22), de acordo com a metodologia proposta, ainda faltaria aplicar aos termos selecionados o tratamento de similaridade através do algoritmo *stemmer* e verificar se o termo é um conceito existente na ontologia do domínio, ou seja, esta lista de termos ainda não é a final e poderá diminuir.

Reunião 1		Reunião 2		Reunião 3		Reunião 4		Reunião 5		Reunião 6		Reunião 7		Reunião 8		Reunião 9		Reunião 10	
Termo	Freq.	Termo	Freq.	Termo	Freq.	Termo	Freq.	Termo	Freq.	Termo	Freq.	Termo	Freq.	Termo	Freq.	Termo	Freq.	Termo	Freq.
table	64	tab	27	comanda	58	delivery	18	balance	33	table	33	delivery	34	table	52	waiter	31	product	27
report	38	table	14	table	20	client	17	product	22	booking	21	report	24	waiter	46	commission	25	report	19
conference	20	order	13	code	15	address	10	order	15	client	16	time	19	time	20	report	19	period	10
Comanda	16	screen	12	Order	13	deliveryman	10	quantity	12	day	15	client	18	user	10	sales	15	example	9
class	11	delivery	10	product	10	screen	9	code	11	time	12	filter	16	identification	10	period	14	discount	8
Order	11	terminal	10	bar	6	neighborhood	9	system	11	one	7	deliveryman	12	restaurant	7	month	9	Additions	7
product	10	system	8	paper	6	report	7	value	10	system	7	city	10	service	7	date	9	quantity	7
system	10	example	7	data	5	example	6	barcode	10	border	6	neighborhood	10	configuration	6	filter	7	sales	7
information	6	field	7	example	5	information	6	label	8	example	6	system	9	option	6	week	7	total	7
transfer	6	access	6	number	5	class	5	class	6	hours	6	date	8	comanda	6	day	6	customer	5
10	5	cocktail	6	option	5	company	5	comanda	6	color	5	example	8	commission	5	fortnight	4	date	5
coupon	5	information	6	3	4	default	5	information	6	date	5	products	7	delivery	5	one	4	end	5
status	5	default	5	class	4	field	5	check	5	user	5	reports	7	supervisor	5	option	4	filter	5
PAF	5	counter	4	client	4	kitchen	5	table	4	attendant	4	commission	6	password	4	calculation	3	code	4
classes	4	detail	4	operation	4	order	5	user	4	hour	4	period	6	setting	4	manager	3	unit	4
code	4	number	4	1	3	Central	4	customer	3	three	4	day	5	value	4	right	3	comanda	3
date	4	observation	4	5	3	code	4	delivery	3	two	4	gourmet	5	access	3	service	3	delivery	3
example	4	waiter	4	amount	3	data	4	reading	3	way	4	option	5	case	3	two	3	description	3
fiscal	4	configuration	3	attributes	3	place	4	time	3	configuration	3	data	4	example	3	user	3	gourmet	3
items	4	customer	3	command	3	register	4	way	3	person	3	localities	4	gourmet	3	configuration	2	SESC	3
One	4	settings	3	consumption	3	search	4	etc.	2	restaurant	3	One	4	ID	3	definition	2	system	3
time	4	user	3	customer	3	system	4	example	2	care	2	something	4	one	3	dollars	2	100	2
value	4	way	3	eight	3	employees	3	form	2	center	2	user	4	product	3	example	2	150	2
waiter	4	attendant	2	reader	3	InfoWorld	3	IP	2	course	2	value	4	screen	3	reports	2	50	2
client	3	change	2	registration	3	need	3	models	2	deadline	2	ADM	3	Torres	3	thousand	2	ADM	2
description	3	changes	2	scanner	3	observation	3	network	2	dinner	2	ADM.	3	ah	2	three	2	B	2
message	3	class	2	type	3	time	3	products	2	layout	2	case	3	attendance	2	waiters	1	client	2
number	3	discounts	2	cancellation	2	tip	3	screen	2	limit	2	columns	3	counter	2	months	1	cost	2
Products	3	extras	2	card	2	10.00	2	settings	2	name	2	configuration	3	five	2	weeks	1	counter	2
reading	3	features	2	case	2	2.00	2	support	2	Ok	2	decision	3	login	2	commissions	1	coupon	2
total	3	gourmet	2	codes	2	addresses	2	three	2	option	2	Torres	3	lot	2	accordance	1	day	2
case	2	lemonade	2	commands	2	amount	2	two	2	part	2	ah	2	movement	2	amount	1	department	2
consideration	2	note	2	configuration	2	CEP	2	units	2	pattern	2	cafeteria	2	need	2	comandas	1	drink	2
...		

Figura 22 - Lista de termos por reunião

Para resumir a análise feita nas dez reuniões, na Figura 23 é apresentada para cada reunião uma série de informações que auxiliam a compreensão dos principais dados extraídos das análises, que foram realizadas pela aplicação da metodologia proposta. A seguir detalharemos as informações apresentadas na Figura 23:

- qtde de termos da reunião: indica quantos termos classificados como substantivos, números e palavras estrangeiras foram extraídos do texto;
- média de frequência dos termos: indica a média da frequência de ocorrência dos termos classificados na reunião;
- valor do desvio padrão: valor calculado para o desvio padrão considerando as frequências dos termos da reunião;
- valor do desvio padrão x 2: o valor do desvio padrão multiplicado por dois;
- Faixa 1: média + desvio padrão: aponta o valor da frequência em que inicia a Faixa 1;
- Faixa 2: média + desvio padrão x2: aponta o valor da frequência em que inicia a Faixa 2;
- qtde termos Faixa 1: aponta a quantidade de termos que foram classificados para a Faixa 1 e o percentual que representa esta quantidade em relação ao total de termos selecionados da reunião;
- qtde termos Faixa 2: aponta a quantidade de termos que foram classificados para a Faixa 2 e o percentual que representa esta quantidade em relação ao total de termos selecionados da reunião;
- Faixa 1 + Faixa 2: é a soma da quantidade de termos classificados nas duas faixas e o percentual que representa esta quantidade em relação ao total de termos selecionados da reunião;
- Qtde termos Freq \geq 3: indica a quantidade de termos que possuem frequência igual ou maior que três e o percentual que representa esta quantidade em relação ao total de termos selecionados da reunião.

Reunião 1		Reunião 2		Reunião 3		Reunião 4		Reunião 5	
Qtde Termos da reunião	96	Qtde Termos da reunião	97	Qtde Termos da reunião	113	Qtde Termos da reunião	129	Qtde Termos da reunião	82
Média de frequência dos termos	3,7	Média de frequência dos termos	2,6	Média de frequência dos termos	2,7	Média de frequência dos termos	2,4	Média de frequência dos termos	3,1
Valor do Desvio Padrão	7,8	Valor do Desvio Padrão	3,7	Valor do Desvio Padrão	5,9	Valor do Desvio Padrão	2,6	Valor do Desvio Padrão	4,9
Valor do Desvio Padrão x 2	15,6	Valor do Desvio Padrão x 2	7,3	Valor do Desvio Padrão x 2	11,8	Valor do Desvio Padrão x 2	5,3	Valor do Desvio Padrão x 2	9,8
Faixa 1: Média + Desvio Padrão	11	Faixa 1: Média + Desvio Padrão	6	Faixa 1: Média + Desvio Padrão	9	Faixa 1: Média + Desvio Padrão	5	Faixa 1: Média + Desvio Padrão	8
Faixa 2: Média + Desvio Padrão x 2	19	Faixa 2: Média + Desvio Padrão x 2	10	Faixa 2: Média + Desvio Padrão x 2	14	Faixa 2: Média + Desvio Padrão x 2	8	Faixa 2: Média + Desvio Padrão x 2	13
Qtde termos Faixa 1	3	Qtde termos Faixa 1	6	Qtde termos Faixa 1	2	Qtde termos Faixa 1	9	Qtde termos Faixa 1	6
%	3,1%	%	6,3%	%	2,1%	%	9,4%	%	6,3%
Qtde termos Faixa 2	3	Qtde termos Faixa 2	6	Qtde termos Faixa 2	3	Qtde termos Faixa 2	6	Qtde termos Faixa 2	3
%	3,1%	%	6,3%	%	3,1%	%	6,3%	%	3,1%
Faixa 1 + Faixa 2	6	Faixa 1 + Faixa 2	12	Faixa 1 + Faixa 2	5	Faixa 1 + Faixa 2	15	Faixa 1 + Faixa 2	9
%	6,3%	%	12,5%	%	5,2%	%	15,6%	%	9,4%
Qtde termos Freq >=3	31	Qtde termos Freq >=3	23	Qtde termos Freq >=3	27	Qtde termos Freq >=3	28	Qtde termos Freq >=3	20
%	32,3%	%	24,0%	%	28,1%	%	29,2%	%	20,8%

Reunião 6		Reunião 7		Reunião 8		Reunião 9		Reunião 10		Item	Média
Qtde Termos da reunião	94	Qtde Termos da reunião	104	Qtde Termos da reunião	100	Qtde Termos da reunião	67	Qtde Termos da reunião	113	Qtde Termos da reunião	99,5
Média de frequência dos termos	2,8	Média de frequência dos termos	3,3	Média de frequência dos termos	3,2	Média de frequência dos termos	3,4	Média de frequência dos termos	2,3	Média de frequência dos termos	2,9
Valor do Desvio Padrão	4,5	Valor do Desvio Padrão	5,0	Valor do Desvio Padrão	7,0	Valor do Desvio Padrão	5,6	Valor do Desvio Padrão	3,4	Valor do Desvio Padrão	5,0
Valor do Desvio Padrão x 2	9,1	Valor do Desvio Padrão x 2	10,0	Valor do Desvio Padrão x 2	14,1	Valor do Desvio Padrão x 2	11,1	Valor do Desvio Padrão x 2	6,7	Valor do Desvio Padrão x 2	10,1
Faixa 1: Média + Desvio Padrão	7	Faixa 1: Média + Desvio Padrão	8	Faixa 1: Média + Desvio Padrão	10	Faixa 1: Média + Desvio Padrão	9	Faixa 1: Média + Desvio Padrão	6	Faixa 1: Média + Desvio Padrão	8,0
Faixa 2: Média + Desvio Padrão x 2	12	Faixa 2: Média + Desvio Padrão x 2	13	Faixa 2: Média + Desvio Padrão x 2	17	Faixa 2: Média + Desvio Padrão x 2	14	Faixa 2: Média + Desvio Padrão x 2	9	Faixa 2: Média + Desvio Padrão x 2	13,0
Qtde termos Faixa 1	2	Qtde termos Faixa 1	6	Qtde termos Faixa 1	2	Qtde termos Faixa 1	2	Qtde termos Faixa 1	5	Qtde termos Faixa 1	4,3
%	2,1%	%	6,3%	%	2,1%	%	2,1%	%	5,2%	%	4,5%
Qtde termos Faixa 2	5	Qtde termos Faixa 2	5	Qtde termos Faixa 2	3	Qtde termos Faixa 2	5	Qtde termos Faixa 2	4	Qtde termos Faixa 2	4,3
%	5,2%	%	5,2%	%	3,1%	%	5,2%	%	4,2%	%	4,5%
Faixa 1 + Faixa 2	7	Faixa 1 + Faixa 2	11	Faixa 1 + Faixa 2	5	Faixa 1 + Faixa 2	7	Faixa 1 + Faixa 2	9	Faixa 1 + Faixa 2	8,6
%	7,3%	%	11,5%	%	5,2%	%	7,3%	%	9,4%	%	9,0%
Qtde termos Freq >=3	21	Qtde termos Freq >=3	31	Qtde termos Freq >=3	25	Qtde termos Freq >=3	19	Qtde termos Freq >=3	21	Qtde termos Freq >=3	24,6
%	21,9%	%	32,3%	%	26,0%	%	19,8%	%	21,9%	%	25,6%

Figura 23 - Resumo dos dados das reuniões

Diante das informações apresentadas na Figura 23 podemos destacar que em média, a quantidade de termos selecionados na Faixa 1 + Faixa 2 (9%) foi quase um terço do valor da quantidade de termos selecionados com $Freq \geq 3$ (25,6%). Isso significa que considerando uma reunião em que fosse extraída a quantidade de cem termos, se fosse usado o critério de $Freq \geq 3$, vinte e seis termos seriam considerados relevantes, enquanto que usando a heurística baseada nos valores do desvio padrão, teríamos nove termos considerados relevantes, e na média, metade em cada faixa.

Esta análise reforçou a percepção, já apontada no estudo de viabilidade, da necessidade de apresentar uma heurística para aperfeiçoar o processo de seleção dos termos relevantes, que foi alcançada com o cálculo utilizado para definir as duas faixas de termos selecionados, baseado no desvio padrão.

Na primeira etapa desta seção, apresentamos as análises realizadas sobre a geração automática dos termos relevantes baseado na metodologia proposta, para as dez reuniões do estudo de caso. A seguir, apresentaremos as análises dos dados focando na comparação dos termos gerados automaticamente, com os termos apontados pelos humanos no *survey*, para as mesmas dez reuniões do estudo de caso. O principal objetivo desta comparação foi avaliar a precisão da metodologia proposta, para elencar os termos relevantes de uma reunião, comparados à percepção dos participantes das reuniões.

Convém destacar que os termos que foram gerados automaticamente para as dez reuniões do estudo de caso, antes de serem comparados com os termos apontados pelos participantes das reuniões, passaram pelo tratamento do algoritmo *stemmer* e também os termos foram comparados e limitados aos conceitos da ontologia do domínio, conforme a metodologia proposta (capítulo 5).

Na Tabela 15 temos um exemplo de comparação dos termos gerados automaticamente para uma reunião e os termos que foram apontados pelos participantes da reunião no *survey*. Esta mesma análise foi realizada para todas as dez reuniões e os resultados estão sinteticamente apresentados na Tabela 16.

Tabela 15 - Comparação de termos gerados automaticamente com avaliação humana

	Qtde	% Acerto*	% Acerto**				
Faixa 2 (média + DP x2)	3	92%	100%				
Faixa 1 (média + DP)	2	75%	100%				
Termos não detectados automaticamente	5	50%	80%				
Termos Ontologia	Reunião 3						
Classes	Automático	H1	H2	H3	H4	HT*	HT**
Balance						-	-
Client			x			-25%	0%
Comanda	Faixa 2	x	x	x	x	100%	100%
Delivery						-	-
Deliveryman						-	-
Meeting						-	-
Order	Faixa 1	x	x	x	x	100%	100%
Product	Faixa 1	x		x		50%	100%
Restaurant		x				-25%	0%
Table	Faixa 2	x	x	x	x	100%	100%
Tag						-	-
Waiter						-	-
Datatype properties						-	-
Address						-	-
Barcode						-	-
Baud_Rate						-	-
Birthday						-	-
CEP_Address						-	-
City						-	-
CNPJ						-	-
Code_Product	Faixa 2	x		x	x	75%	100%
Comanda_Number		x	x		x	-75%	100%
Commission						-	-
Commission_type						-	-
CPF						-	-
Date_Time						-	-
Delivery_number						-	-
Description						-	-
Frequency						-	-
ID						-	-
IP_of_Communication						-	-
Model						-	-
Name						-	-
Neighborhood						-	-
Observation						-	-
Order_Number		x		x		-50%	100%
Order_Type						-	-
Phone						-	-
Product_Type						-	-
Quantity						-	-
RG						-	-
Status_Order						-	-
Table_Number						-	-
Table_Status						-	-
Term						-	-
Transferred			x			-25%	0%
Type_of_Communication						-	-
UF_Address						-	-
Value_Total						-	-
Value						-	-
Vehicle						-	-
Object properties						-	-
Attend						-	-
...						-	-

A Tabela 15 apresenta todos os termos da ontologia divididos em categorias: *classes*, *datatype properties* e *object properties* e, na coluna “Automático”, quais destes conceitos foram detectados automaticamente e qual a classificação atribuída: “Faixa 1” ou “Faixa 2”. Também

temos as colunas “H1”, “H2”, “H3” e “H4”, que representam os apontamentos feitos pelos participantes do *survey*, indicando os conceitos considerados por eles relevantes. A coluna “HT*” totaliza as marcações feitas pelos participantes, indicando em valores percentuais a proporção de humanos que marcaram aquele conceito. Quando o valor é negativo, significa que o conceito foi apontado pelos participantes, mas não foi detectado automaticamente. A coluna “HT**” também totaliza as marcações feitas pelos participantes, mas indica se a metade ou maioria dos participantes apontaram o termo como relevante. Se a metade ou maioria apontar como relevante é apontado 100%, caso contrário 0%.

Essa divisão em duas formas de analisar a totalização das marcações dos participantes foi feita pelo fato de, ser esta, uma avaliação subjetiva, difícil de ser quantificada. Desta forma apresentamos duas visões, uma que apresenta uma média aritmética das marcações dos avaliadores para os termos (HT*); e outra que considera se um termo é relevante (100%) ou não (0%), considerando se a metade ou maioria dos participantes o considerou relevante (HT**).

Na parte superior da Tabela 15 temos um resumo dos dados apresentados. O item “Faixa 2 (média + DP x2)” indica quantos conceitos nesta faixa foram apontados automaticamente, sendo informado na coluna “Qtde”. Para a margem de acerto em relação à avaliação humana, temos as colunas “% Acerto*”, baseada nos valores apontados em “HT*” e a coluna “% Acerto**”, baseada nos valores apontados em “HT**”. Abordando primeiramente a coluna “% Acerto*”, esta margem de acerto significa o percentual de marcações feitas pelos participantes considerando “HT*”. Por exemplo, para a Faixa 2 tivemos três termos detectados automaticamente: “Comanda”, onde todos os participantes também o identificaram como um conceito relevante, ou seja 100%; “Table”, todos também apontaram como relevante, ou seja 100% e; “Code_Product”, onde três dos quatro participantes apontaram como relevante, ou seja, 75%. Desta forma, para a Faixa 2, houve na média uma margem de acerto de 92%. Da mesma forma foi feito o cálculo para a Faixa 1, onde foi encontrado o valor de 75%. Diferentemente, a coluna “% Acerto**”, utiliza o percentual de marcações feitas pelos participantes considerando o valor de “HT**”. Por exemplo, para a Faixa 2 tivemos três termos detectados automaticamente: “Comanda”, onde todos os participantes também identificaram como um conceito relevante, ou seja 100%; “Table”, todos também apontaram como relevante, ou seja 100% e; “Code_Product”, onde três dos quatro participantes apontaram como relevante, ou seja, a maioria, portanto também será 100%. Desta forma, para a Faixa 2, houve uma margem de acerto de 100%. Da mesma forma foi feito o cálculo para a Faixa 1, onde foi apontado o valor de 100%.

Para o item “Termos não detectados automaticamente”, temos na coluna “Qtde” quantos termos foram apontados por humanos no *survey* e não foram gerados automaticamente através da metodologia proposta. O cálculo para apontar os valores das colunas “% Acerto*” e “% Acerto**” é diferente do que foi usado para Faixa 2 e Faixa 1. Neste caso, a coluna “% Acerto*” indica o percentual de quantos termos foram apontados por humanos e também detectados automaticamente. Para o exemplo da Tabela 15, temos 10 termos apontados por humanos e cinco não foram detectados pelo método automático, portanto o valor é 50%. Na coluna “% Acerto**”, seguimos usando o conceito de “metade ou maioria”, portanto, um termo só é considerado relevante se foi apontado pela metade ou maioria dos participantes. Neste caso então, os termos “Client”, “Restaurant” e “Transferred” foram desconsiderados, pois nestas três situações somente um participante considerou o termo como relevante. Desta forma o valor da coluna “% Acerto**” é 71,4%, ou seja, de sete termos apontados pelos humanos, dois não foram detectados, pois três foram descartados.

Estas análises apresentadas para a reunião usada como exemplo na Tabela 15, também foram realizadas para todas as dez reuniões do estudo de caso e os dados estão sinteticamente apresentados na Tabela 16.

Tabela 16 - Resumo dos dados das reuniões

	Reunião 1			Reunião 2			Reunião 3		
	qnte	Acerto*	Acerto **	qnte	Acerto*	Acerto **	qnte	Acerto*	Acerto **
Faixa 2 (média + DP x2)	1	100%	100%	3	83%	100%	3	92%	100%
Faixa 1 (média + DP)	2	63%	100%	0	-	-	2	75%	100%
Termos não detectados automat.	4	43%	86%	4	43%	100%	3	33%	83%
	Reunião 4			Reunião 5			Reunião 6		
	qnte	Acerto*	Acerto **	qnte	Acerto*	Acerto **	qnte	Acerto*	Acerto **
Faixa 2 (média + DP x2)	5	100%	100%	3	89%	100%	4	100%	100%
Faixa 1 (média + DP)	1	100%	100%	4	67%	80%	0	-	-
Termos não detectados automat.	6	50%	60%	5	58%	88%	2	67%	100%
	Reunião 7			Reunião 8			Reunião 9		
	qnte	Acerto*	Acerto **	qnte	Acerto*	Acerto **	qnte	Acerto*	Acerto **
Faixa 2 (média + DP x2)	3	100%	100%	3	67%	67%	2	100%	100%
Faixa 1 (média + DP)	3	92%	100%	1	25%	0%	1	75%	100%
Termos não detectados automat.	5	55%	86%	5	44%	57%	5	38%	60%
	Reunião 10			Média					
	qnte	Acerto*	Acerto **	Qtde	Acerto*	Acerto **			
Faixa 2 (média + DP x2)	1	100%	100%	2,8	93%	97%			
Faixa 1 (média + DP)	3	67%	100%	1,7	70%	85%			
Termos não detectados automat.	2	67%	80%	4,1	50%	80%			

Podemos observar na Tabela 16 que para a Faixa 2 a média de “Acerto*” foi de 93%. Isso significa que dos termos gerados automaticamente, 93% estavam de acordo com a percepção

humana dos participantes do *survey*. Este cálculo considera a média aritmética para verificar se os respondentes do *survey* apontam o termo como relevante. Por exemplo, se um termo é considerado relevante por três dos quatro respondentes, apontamos que 75% consideram o termo relevante. Já a média “Acerto**” foi de 97%, onde não usamos a média aritmética, e sim, é apontado se a metade ou maioria considera o termo relevante, caso contrário (minoridade) ele não é considerado relevante.

Já para os termos não detectados automaticamente, temos o índice de 50% de acerto (Acerto*), indicando que foi detectado na média 50% dos termos apontados pelos respondentes do *survey*. Neste cálculo foi considerado que se somente um respondente apontou determinado termo como relevante e os demais respondentes não, e este termo não foi gerado automaticamente, ele foi considerado não detectado. Já o índice de Acerto**, que foi de 80%, significa que 80% dos termos apontados pelos respondentes foram gerados automaticamente. Este cálculo considera que se um termo foi apontado como relevante por somente um dos quatro respondentes, ou seja, não representa a metade nem a maioria, este termo foi descartado, portanto, retirado do índice dos termos não detectados automaticamente.

Outra observação, sobre a análise dos dados das reuniões do estudo de caso, diz respeito à associação da ocorrência dos termos classificados como mais relevantes a um determinado tipo de conceito da ontologia. Podemos observar este tipo de associação na Tabela 17, onde, por exemplo, é apontado que a maioria dos termos classificados na Faixa 2, onde estão os conceitos com maior frequência, estão associados majoritariamente com conceitos de tipo *class* na ontologia. Já a maioria dos termos classificados como Faixa 1, que são de menor frequência que os da Faixa 2, ocorreram mais vezes associados a conceitos do tipo *datatype property* da ontologia. Os conceitos do tipo *object property* apresentaram somente um termo como relevante.

Baseado nos dados da Tabela 17, percebemos que embora somente doze conceitos da ontologia sejam do tipo *class*, 60% dos termos classificados como relevantes (Faixa 2 e Faixa 1) estão associados a estes conceitos e se considerarmos somente a Faixa 2 (mais relevante) 71% são do tipo *class*.

Tabela 17 - Análise dos termos por tipos de conceitos da ontologia

Tipo de conceito da ontologia	Qtde
Class	12
Qtde termos classificado como Faixa 2	20
Qtde termos classificado como Faixa 1	7
Qtde não detectado automaticamente	13
Datatype property	38
Qtde termos classificado como Faixa 2	7
Qtde termos classificado como Faixa 1	10
Qtde não detectado automaticamente	28
Object property	11
Qtde termos classificado como Faixa 2	1
Qtde termos classificado como Faixa 1	0
Qtde não detectado automaticamente	1

Observamos então, que os conceitos do tipo *class* são os mais relevantes do domínio e que os *datatype properties* são conceitos subordinados às *classes*. Como o objetivo da metodologia é indexar os conceitos mais relevantes das reuniões de projeto, as *classes* poderiam representar um nível mais representativo e os *datatype* e *object properties* um nível secundário. Essa diferenciação pode ser usada, por exemplo, nas consultas, pois num primeiro momento pode-se apresentar para uma consulta as classes e não todos os termos da ontologia. Caso a consulta não atenda às expectativas, apresenta-se um segundo nível, composto pelos demais conceitos da ontologia.

Mesmo com todos os resultados satisfatórios já apresentados quando comparados os conceitos gerados automaticamente e os apontados pelos humanos, como se pode observar na Tabela 16, ainda percebemos a necessidade de realizar uma análise da correlação entre os conceitos gerados automaticamente e os conceitos apontados pelos respondentes no *survey*. O motivo para a análise da correlação entre estes dois conjuntos de valores deveu-se, ao fato de que, não havíamos feito uma análise que incluísse todos os conceitos gerados automaticamente, independente da faixa, e os conceitos não detectados.

Desta forma, criamos para cada reunião uma análise do coeficiente de correlação de Pearson [Oat06] entre a matriz dos conceitos gerados automaticamente e a matriz dos conceitos apontados pelos humanos. Em cada matriz, que é baseada nos termos da ontologia, um conceito

gerado, ou apontado, é marcado com “1”, enquanto que o conceito não gerado é marcado com “0”, como pode ser observado na Tabela 18.

Essa análise serviu para observarmos o comportamento da correlação entre o conjunto total dos conceitos gerados automaticamente, assim como os não gerados, com o conjunto total dos conceitos apontados, e os não apontados, pelos respondentes do *survey*.

Na Tabela 18 apresentamos um exemplo do cálculo do coeficiente de correlação de termos gerados automaticamente para a Reunião 3 e os termos relevantes apontados pelos quatro respondentes do *survey*. Na primeira coluna, “Termos”, é apresentada a listagem dos conceitos da ontologia do domínio. Na segunda coluna, “Aut”, temos em valor “1” os termos que foram gerados automaticamente tanto para a Faixa 2, como para a Faixa 1. As colunas “H1”, “H2”, “H3 e “H4”, indicam em valor “1” o que foi marcado pelos respondentes como conceito relevante. Na coluna “HT*”, temos a média aritmética das marcações feitas pelos respondentes para cada conceito e na coluna “HT**”, temos a indicação, através do valor “1”, se a metade ou maioria dos respondentes considerou aquele conceito relevante. Na última linha da tabela temos os coeficientes de correlação entre os valores gerados automaticamente e os valores gerados a partir da avaliação dos respondentes.

Na Tabela 19 temos o conjunto de todos os coeficientes de correlação para as dez reuniões do estudo de caso. Cabe destacar que foram feitas análises de duas formas: considerando o uso de todos os tipos de conceitos da ontologia (*class*, *datatype* e *object property*); e usando somente os conceitos do tipo *class* da ontologia. Foi realizada essa diferenciação para esta análise, pois conforme apresentado na Tabela 17, a maior parte dos termos gerados automaticamente, assim como os que foram apontados por humanos, são conceitos do tipo *class* da ontologia. E podemos perceber que realmente quando usamos somente os conceitos do tipo *class* existe uma maior correlação entre os termos gerados automaticamente e os apontados por humanos.

Tabela 18 - Exemplo de correlação entre termos gerados automaticamente e os apontados pelos humanos

Ontologia	Reunião 3						
Termos	Aut.	H1	H2	H3	H4	HT*	HT**
Balance	0	0	0	0	0	0,00	0
Client	0	0	1	0	0	0,25	0
Comanda	1	1	1	1	1	1,00	1
Delivery	0	0	0	0	0	0,00	0
Deliveryman	0	0	0	0	0	0,00	0
Meeting	0	0	0	0	0	0,00	0
Order	1	1	1	1	1	1,00	1
Product	1	1	0	1	0	0,50	1
Restaurante	0	1	0	0	0	0,25	0
Table	1	1	1	1	1	1,00	1
Tag	0	0	0	0	0	0,00	0
Waiter	0	0	0	0	0	0,00	0
Address	0	0	0	0	0	0,00	0
Barcode	0	0	0	0	0	0,00	0
Baud_Rate	0	0	0	0	0	0,00	0
...
CEP_Address	0	0	0	0	0	0,00	0
City	0	0	0	0	0	0,00	0
CNPJ	0	0	0	0	0	0,00	0
Code_Product	1	1	0	1	1	0,75	1
Comanda_Number	0	1	1	0	1	0,75	1
Commission	0	0	0	0	0	0,00	0
Commission_type	0	0	0	0	0	0,00	0
Neighborhood	0	0	0	0	0	0,00	0
Observation	0	0	0	0	0	0,00	0
Order_Number	0	1	0	1	0	0,50	1
Order_Type	0	0	0	0	0	0,00	0
Phone	0	0	0	0	0	0,00	0
Product_Type	0	0	0	0	0	0,00	0
Quantity	0	0	0	0	0	0,00	0
RG	0	0	0	0	0	0,00	0
Status_Order	0	0	0	0	0	0,00	0
Table_Number	0	0	0	0	0	0,00	0
Table_Status	0	0	0	0	0	0,00	0
Term	0	0	0	0	0	0,00	0
Transferred	0	0	1	0	0	0,25	0
Type_of_Communication	0	0	0	0	0	0,00	0
UF_Address	0	0	0	0	0	0,00	0
Value_Total	0	0	0	0	0	0,00	0
Request	0	0	0	0	0	0,00	0
Serve	0	0	0	0	0	0,00	0
Correlação		0,77	0,50	0,90	0,78	0,85	0,83

Também podemos observar que o coeficiente de correlação na média sempre ficou acima de 0,7. Isso indica uma forte correlação entre as duas variáveis [Oat06]. A análise que considerou todos os tipos de conceitos da ontologia apresentou entre as dez reuniões, três com coeficientes abaixo de 0,7. No entanto, a análise que considerou somente os conceitos do tipo *class*, apresentou somente uma reunião com coeficiente abaixo de 0,7.

Tabela 19 - Resumo dos coeficientes de correlação entre os termos gerados automaticamente e os apontados por humanos

	Todos os tipos de conceitos da ontologia		Somente o tipo Class da ontologia	
	HT*	HT**	HT*	HT**
Reunião 1	0,63	0,63	0,85	0,82
Reunião 2	0,79	0,76	0,84	0,82
Reunião 3	0,85	0,83	0,93	1,00
Reunião 4	0,85	0,70	0,99	1,00
Reunião 5	0,82	0,87	0,83	0,82
Reunião 6	0,92	0,89	0,89	0,77
Reunião 7	0,94	0,92	0,92	0,82
Reunião 8	0,53	0,32	0,91	0,77
Reunião 9	0,81	0,76	0,79	0,52
Reunião 10	0,61	0,68	0,70	0,77
Média	0,78	0,74	0,86	0,81

Estes resultados reforçam os argumentos que apontam a viabilidade da metodologia proposta para a aquisição de conhecimento, pela geração automática de conceitos extraídos das reuniões.

6.5 Uso de um protótipo para realizar consultas na ontologia

Durante o estudo de caso explanatório foi elaborada uma ontologia do domínio e conforme a metodologia proposta, após serem gerados e classificados os conceitos relevantes das reuniões de projeto, estes devem ser inseridos na ontologia através de uma classe “Meeting” (descrito na seção 5.7 Etapa 7 - Associar conceitos extraídos da reunião com ontologia do domínio).

Após a ontologia do domínio ser enriquecida com as informações extraídas das reuniões, poderíamos realizar consultas na ontologia para verificar, por exemplo, em quais reuniões determinado conceito foi abordado, ou quais os conceitos estão associados a uma determinada reunião, entre outras consultas que explorem a relação entre os conceitos da ontologia do domínio e a sua associação aos conceitos tratados nas reuniões do projeto. Cabe destacar, que o que chamamos de ontologia do domínio é referente à etapa anterior ao enriquecimento da ontologia com as informações das reuniões, sendo depois esta atualização, uma ontologia do projeto, pois não reflete mais somente o domínio.

Para facilitar o processo de elaboração das consultas, foi construído um protótipo que carrega a ontologia do domínio atualizada com as informações das reuniões e através de uma máquina de inferência, que neste caso integrada ao *framework* Jena¹⁹, possibilita que sejam realizadas consultas utilizando os conceitos da ontologia.

A Figura 24 apresenta a tela do protótipo que foi elaborado para a realização das consultas sobre os conceitos da ontologia.

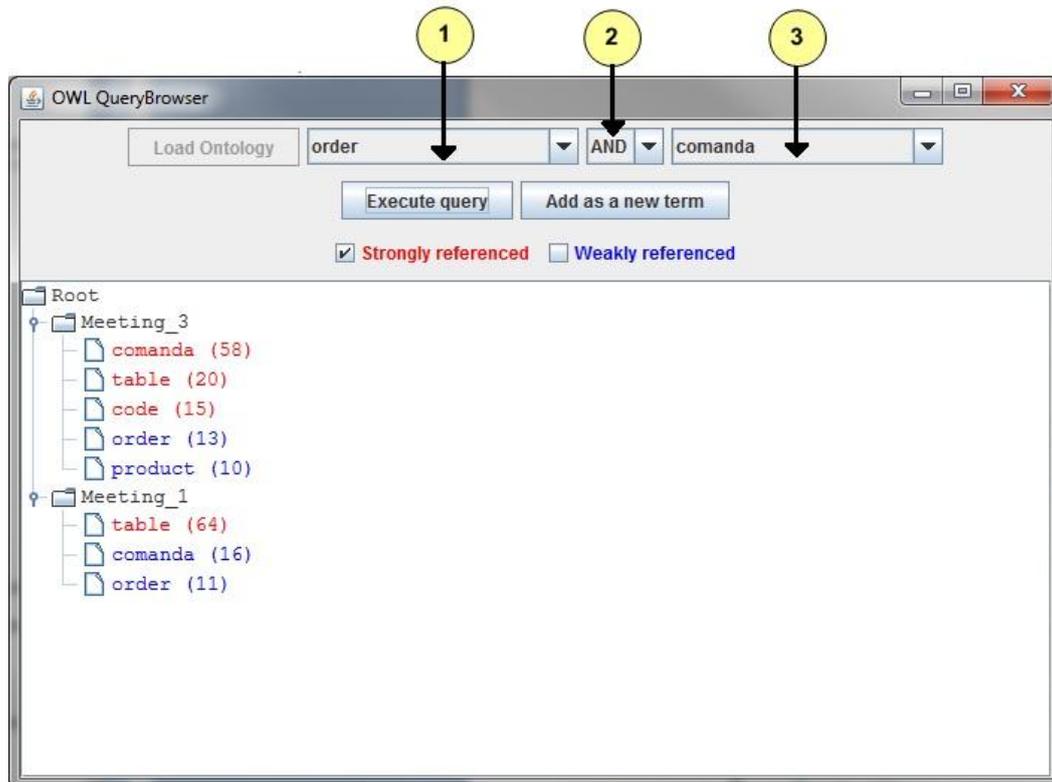


Figura 24 - Tela do protótipo

Neste protótipo a primeira tarefa a ser realizada é a carga da ontologia a ser analisada, que é feita através do botão “Load Ontology”. Depois de carregada a ontologia, podemos realizar as consultas utilizando os *comboboxes* 1 e 3, que listam todos os conceitos da ontologia e o *combobox* 2 que possibilita usar os operadores lógicos “E” e “OU”, no caso da consulta utilizar mais de um termo. Temos duas *checkboxes* para definir o nível de associação do conceito com a reunião, conforme foi apresentado na seção 5.7. Caso queira-se incluir mais uma cláusula à consulta, pode-se utilizar o botão “Add as a new term”. Depois de escolhido o conceito, ou os conceitos, juntamente com os operadores lógicos que compõem a expressão para a consulta, deve-se clicar no botão “Execute query”, para que sejam listadas as reuniões que atendem às especificações da consulta.

¹⁹ <http://jena.sourceforge.net/>

Na Figura 24, temos o exemplo de uma consulta onde foi verificado em quais reuniões os conceitos “Order” e “Comanda” são termos relevantes, tanto para associações do tipo “Strongly referenced”, que equivalem a conceitos da Faixa 2, como associações do tipo “Weakly referenced”, que equivalem a conceitos da Faixa 1. O resultado aponta que as reuniões “Meeting_1” e “Meeting_3” satisfazem os critérios da consulta. A Reunião é apresentada em uma estrutura de árvore, que permite exibir ou ocultar os demais conceitos associados à reunião, juntamente com a frequência e qual o tipo de associação, que é diferenciada pela cor da fonte. Ao clicar-se duas vezes sobre o rótulo da reunião é aberta uma caixa de diálogo que permite ao usuário visualizar o texto com a transcrição textual da reunião, ou que execute o áudio da gravação da reunião, conforme mostrado na Figura 25.

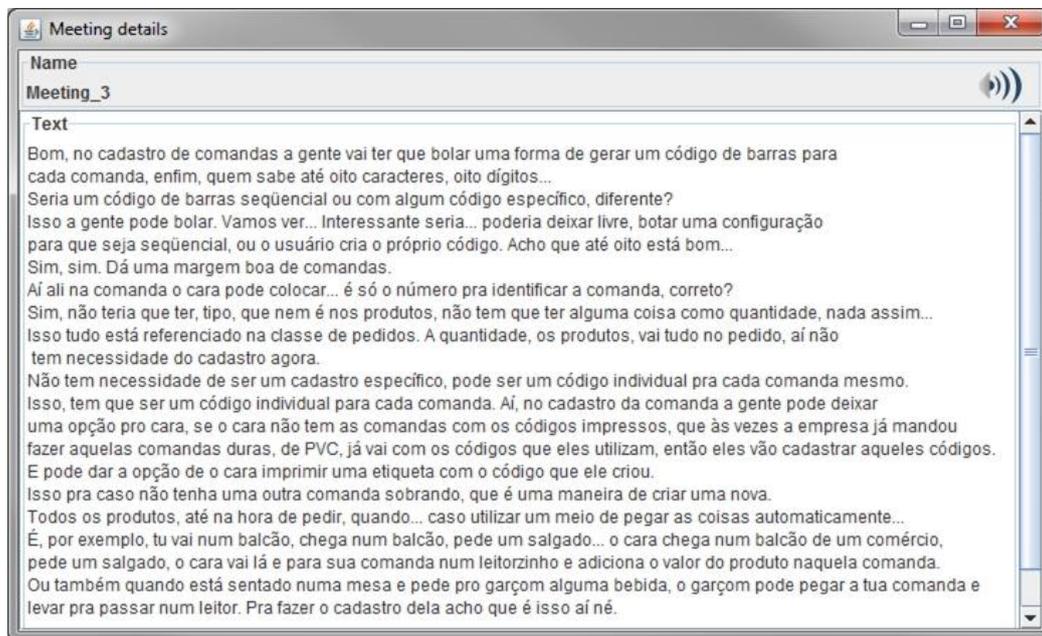


Figura 25 - Tela protótipo com texto e áudio da reunião

Estas consultas poderiam ser realizadas em uma aplicação para edição de ontologias, como o Protégé²⁰, mas o uso do protótipo apresentado facilita a elaboração das consultas e da visualização das informações de forma mais adequada aos objetivos desta pesquisa.

6.6 Contribuições deste capítulo

Este capítulo apresentou uma detalhada descrição de como foi conduzido o estudo de caso explanatório, que representou uma etapa importante no método de pesquisa desta tese, pois se prestou como uma forma de avaliação baseada em resultados empíricos para a metodologia proposta de aquisição de conhecimento em reuniões de projeto de desenvolvimento de software.

²⁰ <http://protege.stanford.edu/>

Além da descrição do contexto da organização, das atividades analisadas durante a execução do estudo de caso e do instrumento de pesquisa utilizado, foi apresentada uma detalhada análise dos dados coletados. Na análise de dados apresentada, destacamos as seguintes contribuições:

- a avaliação da proposta do uso do valor do desvio padrão da frequência de ocorrência dos termos na reunião como forma de extrair os conceitos relevantes de uma reunião. Os resultados foram positivos e confirmaram as expectativas, já apontadas no estudo de viabilidade;
- o uso de duas faixas para a classificação de termos relevantes, pois conforme verificado nos ensaios, os termos mais frequentes nas reuniões tendem a serem os conceitos mais relevantes. Essa divisão apresenta-se útil principalmente para execução de consultas;
- os resultados satisfatórios da comparação dos conceitos gerados automaticamente quando confrontados com os conceitos apontados por humanos para as mesmas reuniões. Na Tabela 16 podemos observar os resultados sobre a margem de acerto nestas comparações e na Tabela 19 podemos observar os coeficientes de correlação, indicando na média uma forte correlação, entre os conceitos gerados automaticamente e aqueles apontados por humanos;
- a percepção que a maioria dos conceitos extraídos das reuniões, assim como aqueles com maior frequência, estão associados a conceitos do tipo *class* na ontologia do domínio. Isso pode simplificar o processo de consultas, pois reduz consideravelmente a quantidade de conceitos utilizados e também pode ser avaliada uma diferenciação de relevância de conceitos do tipo *class*, para os demais tipos.

Além das contribuições provenientes dos resultados da análise de dados, também elencamos como uma contribuição a apresentação de um protótipo que facilita a realização de consultas à ontologia do domínio, atualizada com os conceitos extraídos das reuniões.

Com os resultados apresentados neste estudo de caso exploratório percebe-se que a metodologia proposta é viável, pois leva a resultados consistentes na aquisição de conhecimento de reuniões de projeto de desenvolvimento de software, gerando conceitos capazes de servir como indexadores destas reuniões.

7 Considerações Finais

A maioria das empresas despende horas em reuniões entre funcionários, clientes e colaboradores. Muitas informações importantes são abordadas e quase não se utilizam mecanismos para fazer a gestão do conhecimento que circulam nestas reuniões. Avanços tecnológicos têm permitido gravar e armazenar estas conversações. Meios automáticos para fazer a transcrição e indexação do conhecimento destas reuniões incrementariam em vários aspectos a produtividade dos participantes das reuniões, assim como dos não participantes direta ou indiretamente envolvidos [Tur08].

Para empresas de desenvolvimento de software temos uma situação alinhada ao contexto descrito anteriormente, onde algumas tendências, como as associadas ao surgimento e avanço das metodologias ágeis, têm intensificado ainda mais este tipo de situação. A redução da externalização do conhecimento durante o processo de desenvolvimento de software tem intensificado a comunicação direta entre os envolvidos nos projetos [Cha03].

Algumas iniciativas como o projeto CALO apresentado em [Tur10], indicam a preocupação e os investimentos que estão sendo realizados em várias áreas, no intuito de melhor aproveitar o conhecimento que circula nas reuniões e outros tipos de conversações realizadas entre os diversos envolvidos nas atividades das organizações.

Diante deste quadro, e ratificando como objetivo geral deste projeto “a proposta de uma metodologia para aquisição de conhecimento em reuniões de projetos de desenvolvimento de software”, concluímos que a metodologia proposta nesta tese é adequada para o contexto de desenvolvimento de software e consideramos positivos os resultados apresentados através dos experimentos realizados. Os resultados iniciais, apresentados no capítulo 4, foram satisfatórios e ancoraram a metodologia proposta, com subsídios baseados em resultados empíricos. Estes resultados iniciais foram compostos por: (i) um estudo de caso exploratório, que possibilitou o reconhecimento inicial do problema, identificando questões em aberto onde poderiam ser apresentadas contribuições para o avanço da área; (ii) uma revisão sistemática que permitiu realizar um estudo em profundidade, identificando na literatura um quadro preciso da área de pesquisa e; (iii) um estudo de viabilidade, que apresentou resultados de experimentos que corroboraram com a proposta metodológica apresentada.

Com o subsídio dos resultados do estudo de viabilidade e com todas as etapas da metodologia definidas, realizou-se um estudo de caso explanatório para uma avaliação completa

da metodologia, incluindo todas as suas etapas, em uma instância adequada ao contexto deste projeto. O estudo de caso explanatório, como apresentado na análise de dados, apresentou resultados favoráveis às propostas apresentadas para este projeto.

Salientamos que a metodologia apresentada não é direcionada para ser aplicada em projetos que seguem métodos ágeis. A metodologia proposta pode ser aplicada tanto em projetos que seguem métodos tradicionais, como os que métodos ágeis. Embora os estudos iniciais tenham sido realizados em projetos que aplicavam métodos ágeis, estando então as questões iniciais de pesquisa neste contexto, a evolução do projeto apontou para questões que levaram à especificação de uma metodologia na qual, é indiferente se o projeto segue práticas ágeis ou tradicionais.

Com estes resultados, percebeu-se que a metodologia proposta é viável e que leva a resultados consistentes, resultando em um método capaz de extrair e indexar conhecimento das reuniões de projeto de desenvolvimento de software e que a metodologia apresentada é passível de automação, desta forma, atendendo plenamente a questão de pesquisa geral desta tese.

Apontamos como as principais contribuições desta tese:

- uma metodologia para aquisição do conhecimento de reuniões de projetos de desenvolvimento de software, elaborada e avaliada a partir de um método de pesquisa ancorado em resultados empíricos e satisfatórios ao contexto proposto;
- uma revisão sistemática planejada e executada de acordo com o rigor científico exigido para este tipo de estudo, que serviu para mapear a área de pesquisa deste projeto apontando os principais resultados de trabalhos já publicados;
- uso das questões em aberto mapeados na revisão sistemática para subsidiar e ancorar a metodologia proposta;
- dois estudos de caso com aplicação de *surveys*: o primeiro, um estudo de caso exploratório, aplicado em uma empresa de atuação global, que serviu para o reconhecimento inicial do problema e apontou questões em aberto sobre gestão do conhecimento quando do uso preferencial do conhecimento tácito, ao invés de explícito; o segundo, um estudo de caso explanatório, aplicado em uma empresa que prestou-se com uma instância adequada ao contexto desta tese e, apresentou através da análise de dados, resultados consistentes que justificaram a metodologia proposta;

- um protótipo que permite realizar consultas à ontologia resultante, que é a ontologia do domínio enriquecida com as informações das reuniões, que possibilita verificar, por exemplo, em quais reuniões determinado conceito, ou conjunto de conceitos, foi abordado; ou quais os conceitos estão associados a uma determinada reunião. Também permite visualizar a transcrição do texto e ouvir o áudio das reuniões resultantes das consultas.

Este projeto também apresentou algumas limitações e, que devidos a estas, alguns experimentos não puderam ser realizados, sendo que estes poderiam influenciar nas análises apresentadas. Dentre as limitações, estas são apontadas como as mais relevantes:

- o acesso durante a realização dos experimentos à ferramentas do tipo “audio mining”. Estas ferramentas possibilitam pesquisar por palavras (em modo texto) diretamente no arquivo de áudio, apontando, por exemplo, em que posição (*time stamp*) no arquivo está localizada determinada palavra. Não pudemos avaliar o desempenho destas ferramentas, que poderiam complementar algumas das etapas da metodologia proposta e auxiliar nas consultas, como as realizadas pelo protótipo apresentado. Na seção “5.1 Etapa 1 - Realização da reunião seguindo o protocolo” são apresentadas algumas destas ferramentas;
- a utilização de outras instâncias para o estudo de caso explanatório. Apontamos como pontos restritivos ao uso de outras instâncias durante o projeto: o custo em se aplicar um estudo de caso; limitações de tempo de um projeto de doutoramento e; a dificuldade em encontrar empresas dispostas a colaborar gratuitamente com projetos de pesquisa. Acreditamos que avaliar outras instâncias com diferentes características enriqueceria a análise. Por exemplo, uma instância que aplicasse de modo mais “purista” práticas ágeis, comparada a outra que seguisse um método tradicional. Desta forma poderíamos avaliar outros aspectos relacionados à cultura organizacional, assim como fatores humanos e sociais. Também poderíamos considerar o tamanho da empresa, se utiliza desenvolvimento distribuído, o seu nível de maturidade em gestão do conhecimento, entre outras comparações que poderiam enriquecer a análise.

No intuito de dar continuidade aos resultados já alcançados nesta tese, apresentamos como sugestão para trabalhos futuros, as seguintes propostas:

- um estudo de caso explanatório aplicado a um projeto de desenvolvimento de software, onde além de seguir todas as etapas da metodologia, como o já apresentado nesta tese, o uso do protótipo também possa ser avaliado. Poder-se-ia utilizar a mesma instância usada para esta tese e acrescentar a avaliação do uso do protótipo;
- a utilização de termos freqüentes da reunião, que não foram associados à ontologia do domínio, como uma forma de enriquecer a ontologia com novos conceitos. Por exemplo, um termo que ocorreu muitas vezes em uma reunião e não faz parte da ontologia do domínio, pode ser avaliado para ser tornar um novo conceito da ontologia;
- a automatização de todas as etapas da metodologia, que são realizadas em diferentes ferramentas e que poderiam ser integradas em uma única, facilitando a sua aplicação;
- a avaliação da incorporação da metodologia proposta, como um processo de gestão do conhecimento inserido em um processo de desenvolvimento de software. Portanto, seria avaliado qual o impacto desta prática de gestão do conhecimento, proposta pela metodologia nesta tese, em um ciclo de desenvolvimento de software já estruturado e definido;
- a avaliação dos níveis de relevância dos conceitos considerando a diferenciação dos conceitos do tipo *class* para os demais tipos de conceitos da ontologia;
- o uso dos conceitos extraídos das reuniões como uma forma de avaliar, ou acompanhar, a evolução do conhecimento no contexto de um projeto ou da organização. Com a metodologia apresentada neste trabalho pode-se mapear o uso de determinados conceitos no contexto de um projeto, ou de um conjunto de projetos.

Desta forma, com a implementação dos trabalhos futuros aqui propostos, acreditamos que novos resultados venham a contribuir e somar-se aos resultados já apresentados nesta tese.

Referências Bibliográficas

- [Abe99] ABECKER, A.; DECKER, S. "Organizational Memory: Knowledge Acquisition, Integration, and Retrieval Issues". In: 5th Biannual German Conference on Knowledge-Based Systems, 1999, pp. 113-124.
- [Ada10] ADAMI, A. G. "Automatic Speech Recognition: From the Beginning to the Portuguese Language". In: 9th International Conference on Computational Processing of the Portuguese Language, 2010, 73p.
- [Ala99] ALANI, H.; DASMAHAPATRA, S.; O'HARA, K.; SHADBOLT, N. "Identifying communities of practice through ontology network analysis". IEEE Intelligent Systems, v. 18, n. 2, Mar-Abr 2003, pp. 18-25.
- [Aur03] AURUM, A.; JEFFREY, R.; WOHLIN, C.; HANDZIC, M. "Managing Software Engineering Knowledge". Berlin: Springer Verlag, 2003, 380p.
- [Bab09] BABAR, M. A.; DINGSOYR, T.; LAGO, P.; VLIET, H. VAN. "Software Architecture Knowledge Management: Theory and Practice". New York: Springer, 2009, 279p.
- [Bas94] BASILI, V. R.; CALDIERA, G.; ROMBACH, H. D. "Experience Factory". Encyclopedia of Software Engineering, n. 1, 1994, pp. 469-476.
- [Bj08] BJØRNSON, F. O.; DINGSØYR, T. "Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used". Information and Software Technology, v. 50, n. 11, Out 2008, pp. 1055-1068.
- [Bre07] BRERETON, P.; KITCHENHAM, B.; BUDGEN, D.; TURNER, M.; KHALIL, M. "Lessons from applying the systematic literature review process within the software engineering domain". Journal of Systems and Software, v. 80, n. 4, Abr 2007, pp. 571-583.
- [Bru08] BRUEGGE, B.; SCHILLER, J. "Word Spotting in Scrum Meetings". In: 19th International Conference on Database and Expert Systems Applications, set 2008, pp. 125-129.
- [Buo05] BUONO, A.; POULFELT, F. "Challenges and issues in knowledge management". Greenwich: Information Age Publishing, 2005, 382p.
- [Cal06] CALERO, C.; RUIZ, F.; PIATTINI, M. "Ontologies for Software Engineering and Software Technology". New York: Springer, 2006, 340p.
- [Cañ06] CAÑAS, A.; NOVAK, J. D. "Re-Examining The Foundations for Effective Use of Concept Maps. Concept Maps: Theory, Methodology, Technology". In: Second International Conference on Concept Mapping, 2006, pp.494-502.
- [Cha03] CHAU, T.; MAURER, F.; MELNIK, G. "Knowledge sharing: agile methods vs. Tayloristic methods". In: 12th IEEE International Workshops on Enabling Technologies, 2003, pp 302-307.
- [Cha04] CHAU, T.; MAURER, F. "Knowledge sharing in agile software teams. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)", v. 3075, 2004, pp. 173-183.
- [Cho05] CHOO, C. W. "The Knowing Organization: How Organizations Use Information to Construct Meaning, Create Knowledge, and Make Decisions". 2. ed. Oxford: Oxford University Press, 2005, 368p.

- [Dav98] DAVENPORT, T. H.; PRUSAK, L. "Working knowledge: how organizations manage what they know". Boston: Harvard Business School, 1998, 199p.
- [Des03] DESOUZA, K. C. "Barriers to effective use of knowledge management systems in software engineering". *Communications of the ACM*, v. 46, n. 1, Jan 2003, pp. 99-101.
- [Din09] DINGSØYR, T.; BJØRNSON, F. O.; SHULL, F. "What Do We Know about Knowledge Management? Practical Implications for Software Engineering". *IEEE Software*, v. 26, n. 3, Mai-Jun 2009, pp. 100-103.
- [Din02] DINGSØYR, T.; CONRADI, R. "A Survey of Case Studies of the Use of Knowledge Management in Software Engineering". *International Journal of Software Engineering and Knowledge Engineering*, v. 12, n. 4, Ago 2002, pp. 391-414.
- [Dyb02] DYBÅ, T. "Enabling Software Process Improvement: An Investigation of the Importance of Organizational Issues". *Empirical Software Engineering*, v. 7, n. 4, Dez 2002, pp. 387-390.
- [Dyb08] DYBÅ, T.; DINGSØYR, T. "Empirical studies of agile software development: A systematic review". *Information and Software Technology*, v. 50, n. 9-10, Ago 2008, pp. 833-859.
- [Dyb08a] DYBÅ, T.; DINGSØYR, T. "Strength of evidence in systematic reviews in software engineering". In: *Second ACM-IEEE international symposium on Empirical software engineering and measurement*, 2008, pp. 178-187.
- [Ear01] EARL, M. "Knowledge Management Strategies: Toward a Taxonomy". *Journal of Management Information Systems*, v. 18, n. 1, 2001, pp. 215-242.
- [Fel01] FELDMANN, R.; ALTHOFF, K. "Advances in Learning Software Organizations". Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 2-6.
- [Fis01] FISCHER, G.; OTSWALD, J. "Knowledge management: problems, promises, realities, and challenges". *IEEE Intelligent Systems*, v. 16, n. 1, Jan 2001, pp. 60-72.
- [Fri09] FRIEDLAND, G.; HUNG, H.; YEO, C. "Multi-modal speaker diarization of real-world meetings using compressed-domain video features". In: *IEEE International Conference on Acoustics, Speech and Signal Processing*, Abr 2009, pp. 4069-4072.
- [Gru93] GRUBER, T. R. "A Translation Approach to Portable Ontology Specifications". *Knowledge Creation Diffusion Utilization*, Abr 1993, pp. 199-220.
- [Hap01] HAPPEL, H.; SEEDORF, S. "Applications of Ontologies in Software Engineering". *Engineering*, 2001, pp. 1-14.
- [Jay09] JAYAGOPI, D. B.; HUNG, H.; YEO, C.; GATICA-PEREZ, D. "Modeling Dominance in Group Conversations Using Nonverbal Activity Cues". *IEEE Transactions on Audio, Speech, and Language Processing*, v. 17, n. 3, Mar 2009, pp. 501-513.
- [Joh05] JOHN, M.; MAURER, F.; TESSEM, B. "Human and social factors of software engineering: workshop summary". *ACM SIGSOFT Software Engineering Notes*, v. 30, n. 4, 2005, 6p.
- [Kit04] KITCHENHAM, B. "Procedures for Performing Systematic Reviews". Technical Report TR/SE0401, Keele University, Australia, 2004, 28p.
- [Kom02] KOMI-SIRVIO, S.; MANTYNIEMI, A; SEPPANEN, V. "Toward a practical solution for capturing knowledge for software projects". *IEEE Software*, v. 19, n. 3, Mai 2002, pp. 60-62.

- [Kor07] KORKALA, M.; ABRAHAMSSON, P. "Communication in Distributed Agile Development: A Case Study". In: 3rd EUROMICRO Conference on Software Engineering and Advanced Applications, Jul 2007, pp. 203-210.
- [Kun05] KUNIAVSKY, M.; RAGHAVAN, S. "Guidelines are a tool: building a design knowledge management system for programmers". In: Conference on Designing for User eXperience, 2005, 8p.
- [Lev09] LEVY, M.; HAZZAN, O. "Knowledge management in practice: The case of agile software development". In: ICSE Workshop on Cooperative and Human Aspects on Software Engineering, 2009, pp. 60-65.
- [Lie02] LIEBOWITZ, J. "A Look at NASA Goddard Space Flight Center's Knowledge Management Initiatives". IEEE Software, v. 19, n. 3, Mai 2002, pp. 40-42.
- [Mag07] MAGANTI, H. K.; GATICA-PEREZ, D.; MCCOWAN, I. "Speech Enhancement and Recognition in Meetings With an Audio-Visual Sensor Array". IEEE Transactions on Audio, Speech and Language Processing, v. 15, n. 8, Nov 2007, pp. 2257-2269.
- [Mag07a] MAGANTI, H. K.; MOTLICEK, P.; GATICA-PEREZ, D. "Unsupervised Speech/Non-Speech Detection for Automatic Speech Recognition in Meeting Rooms". In: IEEE International Conference on Acoustics, Speech, and Signal Processing, 2007, pp. 1037-1040.
- [Mel04] MELNIK, G.; MAURER, F. "Direct verbal communication as a catalyst of agile knowledge sharing". In: Agile Development Conference, 2004, pp. 21-31.
- [Mel07] MELUCCI, M.; ORIO, N. "Design, Implementation, and Evaluation of a Methodology for Automatic Stemmer Generation". Journal of the American Society for Information Science, v. 58, n. 5, Mar 2007, pp. 673-686.
- [Mil99] MILTON, N.; SHADBOLT, N.; COTTAM, H.; HAMMERSLEY, M. "Towards a Knowledge Technology for Knowledge Management Applicability of KA to KM". Journal of Human-Computer Studies, v. 51, 1999, 6p.
- [Mon03] MONTONI, M. A. "Aquisição de Conhecimento: Uma Aplicação no Processo de Desenvolvimento de Software". COPPE/UFRJ, 2003, 107p.
- [Nob88] NOBLIT, G. W.; HARE, R. D. "Meta-ethnography: synthesizing qualitative studies". London: Sage Publications, 1988, 88p.
- [Noll 09] NOLL, R. P.; RIBEIRO, M. B. "Concepts-Based Traceability: Using Experiments to Evaluate Traceability Techniques". In: 11th International Conference – ICEIS 2009. Milan: Springer, 2009, pp. 539-550.
- [Non95] NONAKA, I.; TAKEUCHI, H. "The Knowledge – Creating Company: How Japanese Companies Create the Dynamics of Innovation". Oxford: Oxford University Press, 1995, 284p.
- [Noy01] NOY, N. F.; MCGUINNESS, D. L. "Ontology Development 101 : A Guide to Creating Your First Ontology". Stanford Knowledge Systems Laboratory Technical Report, 2001, pp. 1-25.
- [Oat06] OATES, B. J. "Researching Information Systems and Computing". London: Sage Publications, 2006, 341p.
- [O'le98] O'LEARY, D. "Using AI in knowledge management: knowledge bases and ontologies". IEEE Intelligent Systems, v. 13, n. 3, 1998, pp. 34-39.

- [Pik08] PIKKARAINEN, M.; HAIKARA, J.; SALO, O.; ABRAHAMSSON, P.; STILL, J. "The impact of agile practices on communication in software development". *Empirical Software Engineering*, v. 13, n. 3, 2008, pp. 303-307.
- [Por06] PORTER, M. F. "An algorithm for suffix stripping". *Program: electronic library and information systems*, v. 40, n. 3, 2006, p. 211-218.
- [Qum08] QUMER, A.; HENDERSON-SELLERS, B. "An evaluation of the degree of agility in six agile methods and its applicability for method engineering". *Information and Software Technology*, v. 50, n. 4, 2008, pp. 280-295.
- [Rot07] ROTOVNIK, T. "Large vocabulary continuous speech recognition of an inflected language using stems and endings". *Speech Communication*, v. 49, n. 6, Jun 2007, pp. 437-452.
- [Rub04] RUBIN, S. H.; DAI, W. "Knowledge management: overcoming the knowledge acquisition bottleneck". In: *IEEE International Conference on Information Reuse and Integration*, 2004, pp. 420-425.
- [Rup03] RUPING, A. "Agile Documentation: A Pattern Guide to Producing Lightweight Documents for Software Projects". New York: John Wiley & Sons, Inc., 2003, 226p.
- [Rus02] RUS, I.; LINDVALL, M. "Guest Editors' Introduction: Knowledge Management in Software Engineering". *IEEE Software*, v. 19, n. 3, St-Out 2002, pp. 17-19.
- [Sal08] SALAZAR-TORRES, G.; COLOMBO, E.; CORREA DA SILVA, F. S. ; NORIEGA, C. A.; BANDINI, S. "Design issues for knowledge artifacts". *Knowledge-Based Systems*, v. 21, Dez 2008, pp. 856-867.
- [Sch09] SCHNEIDER, K. "Experience and Knowledge Management in Software Engineering". 1. ed. Berlin: Springer Publishing Company, 2009, 235p.
- [Sen06] SENGE, P. "The Fifth Discipline: The Art & Practise of the Learning Organisation". Revised ed. London: Broadway Business, 2006, 445p.
- [Sow00] SOWA, J. F. "Knowledge Representation: logical, philosophical and computational foundations". Brooks/Cole, 2000, 594p.
- [Sta01] STAAB, S.; STUDER, R.; SCHNURR, H.; SURE, Y. "Knowledge processes and ontologies". *IEEE Intelligent Systems*, v. 16, n. 1, Jan-Feb 2001, pp. 26-34.
- [Sto10] STOLCKE, A.; FRIEDLAND, G.; IMSENG, D. "Leveraging speaker diarization for meeting recognition from distant microphones" In: *IEEE International Conference on Acoustics, Speech and Signal Processing*, Mar 2010, pp. 4390-4393.
- [Ter02] TERRA, J. C. C.; GORDON, C. "Portais Corporativos: A revolução na gestão do conhecimento". São Paulo: Negócio Editora, 2002, 480p.
- [Tur04] TURBAN, E.; McLEAN, E.; WETHERBE, J. "Tecnologia da Informação para Gestão". Porto Alegre: Bookman, 2004, 660p.
- [Tur08] TUR, G. ; STOLCKE, A; VOSS, L. "The CALO meeting speech recognition and understanding system". In: *IEEE Spoken Language Technology Workshop*, Dez 2008, pp. 69-72.
- [Tur10] TUR, G.; STOLCKE, A.; VOSS, L.; PETERS, S.; HAKKANI-TÜR, D.; DOWDING, J.; FAVRE, B.; FERNANDEZ, R.; FRAMPTON, M.; FRANDSEN, M.; FREDERICKSON, C.; GRACIARENA, M.; KINTZING, D.; LEVEQUE, K.; MASON, S.; NIEKRASZ, J.; PURVER, M.; RIEDHAMMER, K.;

SHRIBERG, E.; TIEN, J.; VERGYRI, D.; YANG, F. "The CALO meeting Assistant System". IEEE Transactions on Audio, Speech, and Language Processing, Ago 2010, p. 1-10.

- [Ver09] VERGYRI, D.; STOLCKE, A.; TUR, G. "Exploiting user feedback for language model adaptation in meeting recognition". In: IEEE International Conference on Acoustics, Speech, and Signal Processing, Abr 2009, pp. 4737- 4740.
- [Won08] WONGTHONGTHAM, P.; KASISOPHA, N.; CHANG, E.; DILLON, T. "A Software Engineering Ontology as Software Engineering Knowledge Representation". In: Third International Conference on Convergence and Hybrid Information Technology, Nov 2008, pp. 668-675.
- [Yan09] YANZER CABRAL, A. R.; BLOIS RIBEIRO, M.; LEMKE, A. P; SILVA, M. T.; CRISTAL, M.; FRANCO, C. "A case study of knowledge management usage in agile software projects". In: 11th International Conference – ICEIS, Mai 2009, pp.627-638.

Apêndice A - Questionário do Estudo de Caso Exploratório

PUCRS, Computer Science School
 Empresa XXXXX
 ISEG – Intelligent Systems Engineering Group
 September / 2008

Questionnaire: Team Overview

A. INTRODUCTORY NOTE

The present questionnaire is part of a research which has the aim of verifying the use of documents and artefacts under the light of Knowledge Management, as well as the communication standards and cooperation of the team during the software development process based on agile methodologies. It is necessary to obtain information about the stakeholder's profile, used technologies, team and company involved in the research, among some other information that characterize the to-be-observed environment. This questionnaire was elaborated to understand the project's context (PPB/RECOF).

All the information you provide will be kept confidential. Data collected from this questionnaire will be used for research purposes only.

Please read instructions carefully and answer all questions. On average, it should take 15 minutes to complete the questionnaire. We thank you for your collaboration!

B. VOCABULARY

Traditional software development approaches: traditional approaches include Waterfall model and its variances. They are often referred to as “plan-driven”, “task-based” or “Tayloristic approaches”.

Agile software development approaches: refers to a group of software development methodologies that promote development iterations, open collaboration, and process adaptability throughout the life-cycle of the project. Agile approaches include, for instance, Extreme Programming and Scrum.

Hybrid software development approach: it combines practices from both agile and traditional approaches.

Knowledge Management (KM): KM can be defined as the identification and analysis of available and required knowledge assets and knowledge asset related processes, and the subsequent planning and control of actions to develop both the assets and the processes so as to fulfill organizational objectives.

C. RESEARCH GROUP

- Anderson Yanzer (PhD student - PUCRS, Brazil)
- Ana Paula Lemke (PhD student - PUCRS, Brazil)
- Marcos Tadeu (Master student - PUCRS, Brazil)
- xxxxxxxxxxxx (Project Manager, xxxxxxxxxxxx)
- xxxxxxxxxxxx (Development Manager, xxxxxxxxxxxx)

- Marcelo Blois Ribeiro (Supervisor - PUCRS, Brazil)
- Rafael Prikladnicki (Co-Supervisor - PUCRS, Brazil)

D. QUESTIONS

- Answer all questions based on your experience within the company. You have the option to decline to answer any particular question.

General questions

1. Name:

2. Age:

- Less than 20 years old 31 to 35 years old
 20 to 25 years old 36 to 40 years old
 26 to 30 years old Over 41 years old

3. Education level:

- PhD Master/MBA Graduate Undergraduate High School

4. English language expertise:

- Very High High Medium Low Very Low

5. Years of professional experience in Information Technology area:

- Over 5 years 3 to 5 years 1 to 3 years Less than 1 year No experience

6. Company:

- XXXXX XXXXXXXX Other:

7. Your current working relationship with the Company:

- Employee Contractor Trainee/Intern Other:

8. When did you start to work for the Company (Month/Year)?

_____ / _____

Experience in software development methodologies

Traditional Methodologies

Complete the questions 9 to 11 if you **had some experience** in traditional methodologies otherwise go to question 12.

9. Years of professional experience working with traditional software development.

- Over 5 years 3 to 5 years 1 to 3 years Less than 1 year No experience

10. Number of projects working with traditional software development.

- a. I worked in, approximately, _____ distinct projects with traditional software development.

11. Is the documentation generated during traditional software development processes useful in the project tasks?

- Yes, always. Yes, most of the time. Yes, sometimes. No.

Agile Methodologies

Complete the questions 12 to 18 if you **had some experience** in agile methodologies otherwise go to question 19.

12. Years of professional experience working with agile software development.

- Over 5 years 3 to 5 years 1 to 3 years Less than 1 year No experience

13. Number of projects working with agile software development.

- a. I worked in, approximately, _____ distinct projects with agile software development.

14. Which agile methodologies were used on the projects? You may check more than one box.

- Agile RUP
- Adaptive Software Development (ASD)
- Crystal Methods
- Extreme Programming (XP)
- Dynamic Systems Development Methodology (DSDM)
- Feature Driven Development (FDD)
- Scrum
- Other:

15. Did you have any formal training in agile development methodologies in your organization?

- No
- Yes, approximately, how many hours?

16. Do you have any certification on agile methodologies?

- No
- Yes, which?
 - Certified Scrum Master
 - DSDM Foundation
 - DSDM Practitioner
 - Certified Scrum Practitioner
 - Other:

17. Is the documentation generated during agile software development processes useful in the project tasks?

- Yes, always.
- Yes, most of the time.
- Yes, sometimes.
- No.

18. In your opinion, were the following characteristics present in the agile processes used in the projects you had worked?

- (a) Active user involvement. Yes. No.
- (b) Close collaboration of business people and developers. Yes. No.
- (c) Managing requirements throughout the development. Yes. No.
- (d) Testing integrated throughout the lifecycle. Yes. No.
- (e) Frequent delivery of products. Yes. No.
- (h) Self-organizing teams. Yes. No.

Hybrid Methodologies

Complete the questions 19 and 20 if you **had some experience** in hybrid methodologies otherwise go to question 21.

19. Number of projects working with hybrid software development.

a. I worked in, approximately, distinct projects with hybrid software development.

20. If you could select the methodology for a new project, which methodology would you select?

- A traditional methodology.
- An agile methodology.
- A hybrid methodology.

Why do you prefer this methodology?

PPB Control Project related questions

Complete the questions 21 to 26 if you **are working at the PPB Control project** otherwise go to question 27.

21. What is/are your role/roles at the PPB Control project?
 Manager Developer Tester Stakeholder DBA Other:

22. What is/are your SCRUM role/roles at the PPB Control project?
 Scrum Master Product Owner Developer
 Other:

23. PPB domain expertise.
 Very High (I worked in different projects related to PPB domain).
 High (I worked in a project related to PPB domain).
 Medium (I worked in parts of different projects related to PPB domain).
 Low (I worked in parts of another project related to PPB domain).
 Very Low (It is my first project related to PPB domain).

24. What is your allocation at the PPB Control project?
 100% 90 to 70% 60 to 40% 40 to 20% less than 20%

25. Where are you located geographically while working at the PPB Control project (city and country)? _____

26. The PPB Control project has been developed using a hybrid development process. In your opinion, the defined process is mainly a/an:

Traditional methodology Agile methodology
 Why?

27. Is the documentation generated during the process useful in the project tasks?
 Yes, always. Yes, most of the time. Yes, sometimes. No.

RECOF TE project related questions

Complete the questions 28 to 33 if you are working at the RECOF TE project.

28. What is/are your role/roles at the RECOF TE project?
 Manager Developer Tester Stakeholder DBA Other:

29. What is/are your SCRUM role/roles at the RECOF TE project?
 Scrum Master Product Owner Developer
 Other:

30. RECOF domain expertise.
 Very High (I worked in different projects related to RECOF domain).
 High (I worked in a project related to RECOF domain).
 Medium (I worked in parts of different projects related to RECOF domain).
 Low (I worked in parts of another project related to RECOF domain).
 Very Low (It is my first project related to RECOF domain).

31. What is your allocation at the RECOF TE project?
 100% 90 to 70% 60 to 40% 40 to 20% less than 20%

32. Where are you located geographically while working at the RECOF TE project (city and country)? _____

Apêndice B – Sessão *Results* da Revisão Sistemática

Results

We selected 20 studies, of which we extracted the “synthesis of findings” to structure the results in order to answer the focal questions of this review. The results are structured in a topical way, which were the main subjects found in primary studies. We performed the data extraction and the synthesis of findings (section 3.7) to achieve this outline. Table 1 presents the main topics found in selected studies; the Number column represents the number of articles and References column represents which articles were analyzed.

Table 1: main topics of selected studies.

Topic	Number	References
Problems and specific aspects of project documentation in agile methods	6	[S1] [S2] [S3] [S4] [S5] [S6]
Features that involve the transfer and collaboration of tacit knowledge during software development	5	[S1] [S2] [S6] [S7] [S8]
Adoption of a knowledge management methodology in agile projects	5	[S9] [S10] [S11] [S12] [S13]
Tools used in some phase of knowledge management in the agile projects context	5	[S2] [S3] [S14] [S15] [S16]
The influence of human and social factors inside the team and between the team and clients	3	[S1] [S17] [S18]
Use of communities of practice as a way to improve knowledge sharing	3	[S2] [S6] [S19]
Knowledge artifacts and experience knowledge and how they are used in the software development process	2	[S11] [S20]

Before detailing the topics that were selected, along with an analysis of the articles, we will present the strategy adopted for assessing the quality of the selected articles.

Quality assessment of the studies

In Section 3.5 we detailed how the quality criteria for evaluation of the studies were defined. Appendix A and appendix B present the forms used by the authors to evaluate the studies with empirical results and the studies without empirical results. The two authors read all the articles and after that they applied the questionnaires separately. Assessments that showed different results were reevaluated by both authors together in order to come to a consensus.

Table 2 – Quality assessment for the empirical studies

Study	Research Q1	Objectives Q2	R.Design Q3	Sampling Q4	Data C. Q5	Reflexivity Q6	Data A. Q7	Findings Q8	Value Q9	Total
[S1]	1,0	1,0	1,0	1,0	0,5	0,0	0,5	1,0	1,0	7,0
[S4]	1,0	1,0	1,0	1,0	0,5	0,0	0,5	1,0	1,0	7,0
[S6]	1,0	1,0	1,0	1,0	1,0	0,0	1,0	0,5	1,0	7,5
[S7]	1,0	1,0	1,0	1,0	1,0	0,5	1,0	1,0	1,0	8,5
[S8]	1,0	1,0	1,0	1,0	0,5	0,0	0,5	1,0	1,0	7,0
[S10]	1,0	0,5	0,5	1,0	0,5	0,0	0,5	1,0	0,5	5,5
[S12]	1,0	1,0	1,0	0,5	1,0	0,0	1,0	0,5	1,0	7,0
[S14]	1,0	0,5	1,0	0,0	0,5	0,0	0,5	1,0	1,0	5,5
[S17]	1,0	1,0	1,0	1,0	1,0	0,0	0,5	1,0	1,0	7,5
[S18]	1,0	1,0	1,0	1,0	1,0	0,5	1,0	1,0	1,0	8,5
[S20]	1,0	1,0	0,0	0,0	0,5	0,0	0,5	1,0	1,0	5,0
Total	11,0	10,0	9,5	8,5	8,0	1,0	7,5	10,0	10,5	6,9

In these 11 studies with empirical results, the appraisal instrument applied was based on the CASP (appendix A). The main observations of this evaluation were: i) there was a major concern by the authors with the definition of objectives (Q2), justification and description for the Research Design (Q3), a clear statement of Findings (Q8) and a discussion about the Value of Research (Q9), ii) only two studies addressed the question Reflexivity (Q6) and in a superficial way, iii) the average score was 6.9, and the highest score possible was 9.0.

Table 3 – Quality assessment for the non-empirical studies

study	Research Q1	Objectives Q2	Awareness Q3	Suggestion Q4	Developm. Q5	Evaluation Q6	Conclusions Q7	Value Q8	Total
[S2]	1,0	1,0	1,0	1,0	1,0	1,0	1,0	0,5	7,5
[S3]	1,0	1,0	1,0	1,0	1,0	1,0	0,5	0,5	7,0
[S5]	1,0	0,5	1,0	1,0	0,5	0,0	0,5	1,0	5,5
[S9]	1,0	1,0	1,0	1,0	1,0	0,5	1,0	1,0	7,5
[S11]	1,0	1,0	1,0	1,0	1,0	1,0	0,5	1,0	7,5
[S13]	1,0	1,0	1,0	1,0	1,0	1,0	0,5	1,0	7,5
[S15]	1,0	1,0	0,5	1,0	1,0	0,5	0,5	0,5	6,0
[S16]	1,0	1,0	1,0	1,0	1,0	1,0	0,5	0,5	7,0
[S19]	1,0	1,0	1,0	1,0	1,0	0,5	1,0	0,5	7,0
Total	9,0	8,5	8,5	9,0	8,5	6,5	6,0	6,5	6,9

In the 9 studies without empirical results, the applied instrument is presented in appendix B, and it was designed to evaluate research of the "design and creation" type. The main observations of this evaluation were: i) there was a major concern by the authors with the definition of objectives (Q2), Awareness - recognition and articulation of the problem or focus issue - (Q3), Suggestion - how the problem can be addressed - (Q8) and Development – implementation of the tentative design idea -(Q9); ii) the conclusion was the worst rated item (Q7); iii) the average score was 6.9, and the highest score possible was 8.0, iv) studies that did not score in Question 1 (Research) and had no empirical results were discarded.

Studies with empirical results that obtained the degree "zero" in any of the following items: Research (Q1), Objectives (Q2), Findings (Q8) or Value (Q9) were excluded. Studies without empirical results and that obtained the degree "zero" in any of the following items: Research (Q1), Objectives (Q2), Conclusion (Q7) or Value (Q8) were excluded. Studies, with or without empirical results, that obtained the degree of less than 5.0 were excluded.

Problems and specific aspects of project documentation in agile methods

Among the primary studies, six addressed the documentation issue, to some extent. This is an issue that has been discussed for a long time by the software engineering community and has recently gained added attention with the increase of agile methodologies use. In Table 4, we can see the summary of the studies that addressed this topic.

Table 4 – Studies that present the problems and specific aspects of project documentation in agile methods.

Paper/ Year	Topic addressed	Empirical evaluation / Research method (RM)
[S1] 2004	Two visions of knowledge sharing: Knowledge-as-object, a concept associated with higher use of documentation (tayloristic) and knowledge-as-relationship, a concept associated with the use of face-to-face conversation (agile).	Yes: 28 computer professionals and students RM: mixed
[S2] 2003	The knowledge sharing issue, counterpointing the documentation-centric approaches and the face-to-face approaches proposing an approach lying between these two, in a virtual agile team context.	No
[S3] 2003	The knowledge sharing problem among members of development teams by presenting problems on the documentation intensive use, as well as problems arising from the lack of documentation.	No
[S4] 2003	The use of the postmortem technique in agile projects (specifically XP), most suitable for small and medium organizations.	Yes: 9 software development companies RM: single-case
[S5] 2006	The use of knowledge management techniques in agile projects to improve the software reuse for designing and implementing optimization algorithms and metaheuristics.	No
[S6] 2009	The team behavior in the use of the available documentation and communication and interaction aspects among team members and customers.	Yes: 2 virtual agile teams; 22 professionals RM: single-case

In all of these studies on use of documentation we can see that the central issue addressed by the authors is the comparison of the use of documentation in projects that follow traditional methods and projects that follow agile methodologies. In traditional methodologies, the use of documentation is more intense. On the other hand, in agile projects face-to-face communication is preferred, instead of the use of documentation. How much documentation must be reduced in agile projects is still under debate and depends on project features.

Melnik and Maurer [S1] made a comparison of the use of face-to-face communication instead of documentation in certain situations. They observed that there are types of knowledge that can be efficiently shared in documentation such as facts that need to be reported, however there are also skills that are shared more effectively by verbal channels such as concepts, ideas, and desires, which need faster feedback. Another advantage of verbal communication, besides sharing knowledge is that it allows its own transformation.

Thinking along the same line, in which advantages and disadvantages of documentation-centric approaches and face-to-face communication are compared, Holz and Maurer [S2] suggest a mixed approach to capturing and sharing knowledge among members of developer virtual teams. The authors suggest that even though the project is agile, if the team is distributed, there should be a greater focus on knowledge sharing through documents. These authors also cite that the cost to create and maintain updated documentation is one of the factors for reducing documentation in agile methodologies.

As noted, the trend is that documentation in agile projects should be reduced although Holz and Schafer [S3] point out some disadvantages of the lack of documentation in projects:

- experts in particular subjects waste a lot of time answering the same questions;
- the situation in which they know there has been a certain problem before, but they cannot remember its solution;
- the direct knowledge exchange between members of different teams is hard or improbable if they are not co-located, or do not belong to the same community;
- important knowledge is lost as soon as experienced developers leave the project or company.

It should be emphasized that none of those authors discourage the use of documentation. Nevertheless, its use should be minimized as much as possible because there is a high cost to produce and maintain it, but certainly it has its importance even in agile projects. This documentation minimum depends on each project and involves the context of teams, organizations, customers, requirements, contracts, etc.

Dingsøy and Hanssen [S4] present a proposal for capturing externalized knowledge during projects, which is a more simplified version of the postmortem technique to be used in agile projects (specifically XP), most suitable for small and medium organizations. The technique takes into consideration the short time that the team has to make knowledge explicit in documents, but also considers that many details are important to be recorded in documents. Therefore, the postmortem technique, which is a review done at the end of the project to sum up positive and negative experiences, is presented as an alternative in agile projects that have difficulties in documenting important project information.

Another problem that may occur in agile projects is that by concentrating knowledge in the tacit format and sharing it through discussions, the reuse of some knowledge types is hampered. Crawford *et al.* [S5] address the reuse issue in software engineering and state that this is not a technological problem, but a knowledge management problem. Reuse in software engineering has always been associated with the packaging of objects, patterns, templates, and components that are a type of explicit knowledge. But "innovation, problem solving, creativity, intuitive design, good analysis, and effective project management involves more tacit knowledge, which is harder to transfer". Thus, knowledge management, considering tacit knowledge use, plays an important role in reuse. The combination of documents (explicit knowledge) and conversation (tacit knowledge) offers important insights about reusability and software engineering, especially in agile projects.

In Yanzer Cabral *et al.* [S6] a case study was applied to a team that used agile practice. Practitioners mentioned in interviews that the documentation normally available in the project was outdated. However, it was also detected in the interviews that they usually consulted available documentation before consulting colleagues to address questions and to solve problems. Although the documentation is reduced and outdated, the team used it as a source of knowledge to extract the context of the area and reduce the direct communication. At first, we imagined that the respondents would not use the documentation because they considered it outdated and inadequate, but surprisingly they did. This case study corroborates statements in the earlier studies cited, showing that even with the availability of direct conversation channels, the team used outdated project documentation.

In general, none of the studies, with or without empirical results, supports the non-use of documentation. However, due to agile assumptions, we perceive the reduction of documentation in agile projects and from this emerges the need for alternatives to knowledge acquisition and management in other ways besides the traditional methods of documentation.

Features that involve the transfer and collaboration of tacit knowledge during software development

Considering the priority of working with tacit knowledge in agile projects, there are some studies that deal with aspects of knowledge transfer and collaboration in tacit form. Analyzing traditional methods, we can see that the focus is more on transforming tacit knowledge into explicit knowledge and then sharing it among various levels of organization. Regarding agile methodologies, the focus is to create mechanisms for sharing knowledge in a tacit manner, without undergoing externalization processes. In Table 5, we can observe a summary of studies that address this topic.

Table 5 – Studies of features that involve the transfer and collaboration of tacit knowledge during software development.

Paper/ Year	Topic addressed	Empirical evaluation / Research method (RM)
[S1] 2004	The importance and influence of face-to-face communication in tacit knowledge sharing in agile projects.	Yes: 28 computer professionals and students RM: mixed
[S2] 2003	The use of techniques and tools to facilitate the distribution and capture of knowledge among the developers team considering the use of knowledge in both tacit and explicit forms.	No
[S6] 2009	Problems that arise in knowledge sharing and in communication among teams in projects that adopt agile practices.	Yes: 2 virtual agile teams; 22 professionals RM: single-case
[S7] 2005	The pair programming technique as a form of sharing tacit knowledge and contextualizing the difficulties inherent in the process of knowledge management when knowledge is treated in a tacit way.	Yes: students and Masters of technologies of software RM: Experiment
[S8] 2006	Common challenges in agile methodologies and distributed software development, and new challenges that arise when projects adopt these two approaches.	Yes: 3 software development companies; 18 professionals RM: multiple-case

Considering that knowledge should be shared and socialized, we face issues that are associated with the ways of communication used among those involved. As knowledge will be handled preferentially in its tacit form in projects using agile methodologies, the way of sharing it will focus on conversations between everyone involved in the process (team, customer, sponsors, etc.). When knowledge sharing is prioritized by talking or face-to-face communication among the ones who are involved, many benefits can be seen, but also many problems [S1,S7]. These problems are usually associated with human and social factors that influence the communication effectiveness and also with acquisition, storage, and indexing of tacit knowledge.

Melnik and Maurer [S1] highlight the fact that conversation goes beyond the exchange of knowledge, it also provides an environment to transform knowledge. In this study, the authors point out several aspects that must be observed in a face-to-face interaction and the kind of knowledge that is better shared in this communication type. An experiment was conducted with students and professionals to simulate situations of information loss during communication.

Although the most effective way to share tacit knowledge during agile projects is face-to-face communication [S1], the use of tools and techniques can assist the process of capturing, storing, and distributing knowledge in these environments. Holz and Maurer [S2] present the PRIME, a tool that provides a technical infrastructure for the task-specific capture of information needs and their distribution to developers, as well as feedback communication. In the same study, these authors point out communities of practice as an approach that can increase the effectiveness of tacit knowledge sharing circulating in projects.

The effectiveness and quality of the communication channel inside the team and between the team and the customers depends on the participants' initiative and speaking abilities. Canfora *et al.* [S7] address this issue concerning the knowledge exchange effectiveness in pair programming. These authors conducted experiments to verify if the educational background of each individual influences knowledge creation and sharing.

Another issue that arises in agile methods is their use in distributed software development, in which face-to-face conversation must be replaced by non face-to-face. This is a problem because agile practices suggest the use of intense face-to-face communication. Nonetheless, a mix of technology and best practices can come very close to eliminating the inefficiencies of non face-to-face communication.

To address this issue Ramesh *et al.* [S8] performed three case studies that pointed out challenges in agile distributed development which were: communication problems, lack of control, and lack of trust. There are two groups of challenges, the first being distributed development challenges which include: difficulty to initiate communication; miscommunication; decreased frequency of communication; increased communication costs; and time difference. The second being agile communication challenges such as: lack of formal communication and increased demand for informal communication. They cite suggestions to address these problems, which include: people work in the same time zone; formal channels for informal communication; and constant communication through technological mechanisms.

Yanzer Cabral *et al.* [S6] also address these issues and add the cultural difference factor as an aspect that hinders tacit knowledge sharing. These authors mention in their case study that communication problems are intensified in agile projects with distributed teams and problems of knowledge acquisition and storage in agile projects by intense use of informal communication. They highlight the need for a new perspective to a knowledge management methodology to be used in agile development processes. This methodology should be centered on the tacit knowledge that circulates in the projects.

As observed in these primary studies, tacit knowledge sharing brings many advantages in terms of knowledge management, but also brings some challenges. We can highlight two trends to address these challenges: i) the use of approaches such as communities of practice and technological tools to assist communication and knowledge sharing; ii) the development of personal initiative, the ability to communicate, and team knowledge sharing through training courses and capacity building.

Adoption of a knowledge management methodology in agile projects

One way to systematize knowledge management is to adopt a methodology that is defined and organized in phases, which may include creation, acquisition, identification, validation, storage, and sharing of knowledge. Authors such as Nonaka and Takeuchi [8] and Tuomi [52], among others present knowledge management theories and address, for example, the development and transformation of tacit-explicit. In the studies presented in Table 6, we have some examples in which companies have developed frameworks to adopt agile methodologies based on the phases of the creation, processing, and sharing of knowledge.

Table 6 – Studies that present the adoption of a knowledge management methodology in agile projects.

Paper/ Year	Topic addressed	Empirical evaluation / Research method (RM)
[S9] 2003	Knowledge management theories used to explain how extreme programming practices can improve the agility of a organization	No
[S10] 2007	Software development practices adapted to the model of five stages for organizational knowledge creation proposed by Nonaka and Takeuchi	Yes: agile software development team RM: single-case
[S11] 2008	Knowledge creation phases of Nonaka and Takeuchi associated with phases of experience knowledge in the extreme programming context.	No
[S12] 2005	The knowledge creation phase, following Tuomi's model, to extend the technology acceptance model (TAM) for the use of extreme programming.	Yes: 2 projects, an agile and another waterfall; 14 professionals RM: single-case
[S13] 2009	A conceptual framework for agile methodologies acceptance based on a knowledge management perspective.	No

Knowledge creation is presented as the central focus in the agile methodologies implementation by Kähkönen and Abrahamsson [S9], who state that "agility is a result of a more effective knowledge creation process involving both the customer and development team". They use the knowledge creation model proposed by Tuomi to analyze XP practices and state that "these practices enhance knowledge creation through immediate (or frequent) and mutual articulation and appropriation", which are stages of Tuomi's conceptual framework. This framework was chosen because of its ability to clarify the creation of knowledge where several communities of practices are involved.

The study of Ken and Ilio [S10] also focuses on the knowledge creation phase, but using Nonaka and Takeuchi's theory. This was used to improve the creation process in teams of agile software development. One of the goals of the company in which the case study was conducted was to increase the team's creativity, focusing on the use of tacit knowledge within agile methodologies, such as the Scrum / XP. In this case study, the authors suggest a more

favorable environment for innovation in agile development teams that adopt the knowledge creation model of five stages of Nonaka and Takeuchi.

Kokkonniemi [S11] presents an iterative software development cycle based on XP, which provides opportunities to use knowledge in the form of experiences. This iterative cycle is based on activities of knowledge acquisition, creation, transformation, and transfer. This author observed three different phases relating to experience knowledge: recognition, preservation, and exploitation. He identified artifacts used in XP based on experience: user stories, new versions, metaphors, shared stories, test case codes, coding standards, and own rules. After that, the author combines the four phases of knowledge creation of Nonaka and Takeuchi with the experience knowledge phases. The conclusion argues why it is profitable to link extreme programming and experience knowledge through peer reviews.

Articles by Bahli and Zeid [S12] and Chan and Thong [S13] presented studies about technology acceptance model (TAM) in the adoption of agile methodologies as a software development process. The contribution of these studies in relation to previous TAM models is that they incorporate aspects of knowledge management as an important factor in the acceptance of agile methodologies.

Bahli and Zeid [S12] performed a case study in a large company and added the knowledge creation item to TAM based on Tuomi's model to check on the level of acceptance when changing from a waterfall culture to an agile one (XP). Chan and Thong [S13] developed a conceptual framework to understand the acceptance of agile methodologies when making use of knowledge management outcomes like the creation retention, and transfer of knowledge.

These studies suggest that one way to improve the creation and sharing of knowledge in teams that adopt agile practices is the incorporation of practices based on knowledge management frameworks and theories, such as those of Nonaka and Takeuchi, and Tuomi. These frameworks and theories also serve to improve and to adapt the technology acceptance models to be applied in agile methodologies.

Although the results of these studies have been considered satisfactory, we realize the complexity and uncertainty in adopting and following knowledge management models in software development processes, because knowledge management techniques involve unstable aspects, such as human and social issues.

We suggest carrying out case studies using common metrics from software engineering, project management, and knowledge management as a way to verify the impact in the adoption of these knowledge management methods in agile projects. These case studies could be used to compare the performance of agile projects that adopt a knowledge management methodology, in a systematic way, with projects that do not adopt a methodology.

Tools used in some phase of knowledge management in the agile projects context.

Although there is a concern that knowledge management activities and techniques should intervene as little as possible in the process of agile development, there are proposals to use tools that help in knowledge management.

Table 7 displays studies that use technological solutions to improve knowledge management during the agile software development cycle.

Table 7 – Studies of tools used in some phase of knowledge management in the agile projects context.

Paper/ Year	Topic addressed	Empirical evaluation / Research method (RM)
[S2] 2003	A system (PRIME) for task-based capture and pro-active distribution of recurrent information for developers	No
[S3] 2003	A knowledge management tool for agile settings which uses collaborative filtering techniques	No
[S14] 2005	A tool for design knowledge management for web developers as a way of encouraging the change of a software development culture from waterfall-style to agile	Yes: web software development team RM: mixed
[S15] 2005	A collaborative semantic web portal prototype based on Wiki technology for knowledge acquisition and interaction flows in the agile project management	No
[S16] 2004	A wiki-based environment, which handles structured and unstructured information to facilitate the knowledge sharing in agile teams	No

We identified that all the tools and environments listed in Table 10 are focused on improving communication among agile project participants. Some focus on intra-team communication and others on inter-team. Another

common feature of these studies is the perception that in agile projects the knowledge explicitness is reduced in favor of a more face-to-face communication.

Two of the tools presented are wiki-based: i) Tosic *et al.* [S15] present a collaborative semantic web portal prototype based on Wiki technology. These authors argue that using a wiki approach with groupware techniques can improve communication and knowledge acquisition in agile projects. For a better functioning tool, a methodology for collaborative knowledge acquisition is suggested, in which some roles are described, such as: domain practitioner; domain expert data collector; methodology execution manager; ii) Chau and Maurer [S16] present the MASE, which is a tool for web-based collaboration and knowledge sharing in agile projects. This tool is similar to a Wiki and is used to manipulate unstructured information. It also has some plug-ins for working with structured information. These plug-ins are based on common documents in agile development projects. Both studies indicate that the tools used have improved the collaborative process and knowledge sharing among team members.

Two papers present the PRIME system. Its first version is presented in Holz and Maurer [S2], where it is described as a technical infrastructure for the task-specific capture of information needs and their distribution to developers, as well as feedback communication. It allows developers to associate their tasks (to-do list) with information that would be necessary and useful to accomplish these tasks. In this study comparisons with tools similar to PRIME are shown. Holz and Schafer [S3] present an extension of PRIME that is more appropriate for agile projects focusing on aspects such as: redo the same solution because of a lack of documented knowledge from previous projects; problems that have been solved before, but people do not remember the solution; problems when the team is not co-located and; loss of knowledge when an experienced employee leaves the company. This system, through collaborative filtering techniques allows for both the formal specification of “when to offer what information to whom”, and the functionality of heuristic-based information recommendation. We realize that the most important contribution of PRIME is to pro-actively provide users with task-specific recommendations of potentially relevant information items.

We point out two concerns of communication and knowledge sharing among agile teams in software development projects: when it occurs among different teams; and when it occurs among members of a distributed team. These are problems with different outcomes, but with the same root: the lack of face-to-face communication, which in agile projects is the main knowledge sharing form.

Chau and Maurer [S16] focus more on the first problem, which is inter-team learning and they propose a solution using communities of practice through the MASE, which is integrated into the Experience Base, a tool that we have developed as an implementation of the Experience Factory concept [53]. Holz and Maurer [S2] on the other hand, focus on the second problem, which concerns the use of virtual teams using communities of practice. They also approach the PRIME (mentioned above), a system to capture, distribute, and satisfy task-specific information needs, using an approach focused on connecting developers with knowledge sources that are recommended by communities of practice addressed by the developer’s current task.

We can find a wide variety of tools to assist activities in distributed software development, as can be observed in [25], but the focus of our systematic review is on the tools that use the knowledge management perspective in agile projects.

In Kuniavsky and Raghavan [S14] a case is presented concerning the use of the Guidelines system, which is used for knowledge management by web developers in order to change a culture from waterfall-style to user-centered practices. The tool applies some techniques of knowledge management to acquire and share knowledge among developers. Normally, in the literature we find proposals for using tools and techniques to assist the knowledge management in environments where an agile culture has been deployed. But this study proposes an opposite view, in which the first step is to deploy knowledge management practices to increase face-to-face communication, facilitating the later introduction of an agile culture.

Although one of the goals of knowledge management is to try to intervene as little as possible in established processes, we can observe that the adoption of some tools and techniques of knowledge management can improve the process of software development. In the agile project context, the use of tools should be well measured in order to avoid opposing the agile practices. Therefore, we highlight that the ideal solution is to minimize the knowledge externalization process facilitating the knowledge management in its tacit form.

According to Table 7, only one of the studies analyzed presented empirical results. We note the need for more case studies about the impact of using tools to assist in knowledge management. It would be interesting to have studies presenting metrics that compare agile project execution with and without the use of these types of tools.

The influence of human and social factors inside the team, and between the team and clients.

Any knowledge management solution, regardless of the area, involves issues related to human and social factors. In addition to the technological solutions and organizational arrangements adopted when dealing with

knowledge, we always have the human factor [13]. Table 8 presents studies on the influence of human and social factors in agile software development.

Table 8 – Studies that present the influence of human and social factors inside the team, and between the team and clients.

Paper/ Year	Topic addressed	Empirical evaluation / Research method (RM)
[S1] 2004	The influence of face-to-face communication and social interactions in knowledge sharing in agile projects.	Yes: 28 computer professionals and students RM: mixed
[S17] 2005	Agile practices to work the social aspects as a way to improve knowledge sharing, motivation and team collaboration.	Yes: 2 agile projects; 18 programmers RM: single-case
[S18] 2009	The social interactions in software development teams; the use of tacit knowledge in individual and team-level; a team tacit knowledge measure (TTKM) for software developers.	Yes: 2 studies; 30 experienced project managers RM: multiple-case

Ryan and O'Connor [S18] present a team tacit knowledge measure (TTKM) for software developers, which explores “the relationships between tacit knowledge, explicit job knowledge and social interaction and their effect on team performance as measured by efficiency and effectiveness”. This study presents conclusions about: i) tacit knowledge construction, the lack of empirical studies as a result of limited empirical measures for tacit knowledge; ii) how tacit knowledge is acquired and shared in software development teams and; iii) predictive capacity of team tacit knowledge for team performance as measured by effectiveness and efficiency. Among the findings presented in this study, we highlight: tacit knowledge is acquired and shared directly through high quality social interactions and the frequency of this interaction leads to better knowledge sharing; efficiency in software development teams is related to explicit knowledge, formal procedures, while effectiveness depends on tacit knowledge.

The work of Law and Charron [S17] addresses the effects of agile practices on social factors. This work is based on two case studies in TransCanada, in which agile practices for working on social aspects were applied in order to increase knowledge sharing, motivation, and collaboration with customers. Knowledge sharing is shown as a solution to these issues, such as staff turnover and the learning curve problem. These authors highlight the importance of unity and cohesion within the team, the need of a sense of community for the group's growth and the difficulty to achieve these goals in traditional methodologies. They present four agile practices used in their projects: pair programming, co-location, daily status meeting and minimal documentation, and they argued that the use of these practices, by themselves, should improve the sharing of knowledge. In these projects, team motivation was addressed, highlighting some aspects such as i) Learning: highly motivational environments for learning ii) Autonomy: programmers are free to decide the technical aspects and; iii) Social activities: snacks together, parties, etc. By conducting the two case studies, the authors concluded that adopting agile practices, because of increasing direct communication, interferes in personal and social relationships within the team and when these factors are addressed, the team has a better chance of success in a software development project.

Discussing the role of conversation and social interactions, Melnik and Maurer [S1] address the efficiency of verbal and face-to-face communication, highlighting the importance of social interactions as a key element to knowledge sharing in agile projects. In this study, two perspectives discussed by epistemologists are presented: a “codification approach” based on the notion of knowledge-as-object, in which the objects can be created, collected, stored, and reused. In order for these objects to be efficiently managed and transferred, they must be packaged. This encoding can be in the form of documentation, knowledge bases, experience factories, etc. The other perspective, the “personalization approach”, is based on the sociology of knowledge and includes the concept of knowledge-as-relationship. In this perspective, knowledge is uncertain, soft, and embedded in work practices and social relationships. Through this division we can observe that knowledge-as-object is more focused on traditional software development methods, because it is more associated to documentation. The perspective of knowledge-as-relationship is more focused on agile methods. The authors point out direct communication effectiveness when there is a need to share complex knowledge (higher level of abstractionism) because “face-to-face channels offer the prospect of richer communication. This is due to its ability to transmit multiple cues (e.g. physical presence, voice inflection, and body language)”. This is important when there are high levels of ambiguity and uncertainty, situations likely to occur during the communication of requirements.

Thus, we can note that the success of any project related to knowledge management in agile projects, beyond the technological, organizational, and methodological aspects, should consider the human and social factors.

In agile projects these aspects are even more relevant than in traditional projects due to the increase of interactions between team members and stakeholders.

Use of communities of practice as a way to improve knowledge sharing.

In agile projects we have situations in which traditional practices do not indicate appropriate solutions to knowledge management, which can be observed in distributed software development projects and in the large companies that have agile teams. These issues are addressed in the studies presented in Table 9 that focus on the use of communities of practice.

Table 9 - Studies that use communities of practice as a way to improve knowledge sharing.

Paper/ Year	Topic addressed	Empirical evaluation / Research method (RM)
[S2] 2003	How to identify Communities of practice and how to support these communities in virtual teams.	No
[S6] 2009	Communities of practice as an alternative to improve interaction and communication among distributed teams.	Yes: 2 virtual agile teams; 22 professionals RM: single-case
[S19] 2004	The use of communities of practices to establish multiple cooperation among teams in large organizations that use agile practices.	No

Holz and Maurer and [S2] discuss about virtual agile teams and the problems that arise in acquiring and sharing knowledge in these environments and suggest communities of practice as a way to address these problems. The authors present definitions of communities of practice and they show differences from virtual and co-located communities. In the virtual agile teams, communities are geographically dispersed and are based on technological resources for information management. The authors note the need to identify the communities and verify what kind of support should be provided (expert directories, newsgroups, collaborative filtering, etc) to reduce the knowledge maintenance problem. This study presents a tool that incorporates features to support communities of practice in virtual agile teams by proactively capturing and distributing typical information for developers.

Yanzer Cabral *et al.* [S6] conducted a case study, in which the project followed agile methods and was performed by two geographically distributed teams, featuring distributed software development. This case study explains some problems in knowledge sharing and artifacts used by the teams. As proposed in [S2], one way to solve these problems was the adoption of communities of practice based on technological resources (groupware, expert directories, etc).

Kahkonen [S19] suggests the use of communities of practices in large companies as a way to address multi-team communication. His work is based on a case study in Nokia where communities of practice arose informally and voluntarily. This author presents a proposal that in addition to the presence of official team structures, communities of practice should be established beyond traditional team limits. The case study presents three agile methods developed independently in different business units that used facilitated workshops to promote cross-team communication. The conclusion section presents advantages and benefits of using communities of practice at Nokia by incorporating practices and methods that contribute to the creation and operation of such communities.

The organizations have been using communities of practice as an approach to manage knowledge in several scenarios [21]. In software development they are also seen as a viable alternative to solve communication problems in cross-team (co-located and virtual teams) and in large companies. We realized that communities of practice can be used as a way to minimize the problems of handling tacit knowledge, a common situation in agile projects. These communities can also be used, in certain situations, to externalize knowledge (documentation) through (semi) automatic mechanisms for acquisition, processing, and sharing knowledge that circulates within communities of practice.

Knowledge artifacts and experience knowledge and how they are used in the software development process

One of the difficulties perceived in the primary studies selected for our systematic review is that the majority of software development artifacts are based on the use of explicit knowledge, whereas in agile methodologies,

knowledge is treated primarily in tacit form. The concepts of knowledge artifacts and experience knowledge offer a new perspective on how to deal with knowledge in agile projects. In Table 10 two studies dealing with these issues are presented.

Table 10 - Studies that present knowledge artifacts and experience knowledge and how they are used in the software development process

Paper/ Year	Topic addressed	Empirical evaluation / Research method (RM)
[S11] 2008	Experience knowledge and knowledge-based artifacts definition and discussion about how these artifacts can be linked to iterative software development processes (XP)	No
[S20] 2008	A methodology for design and implementation of knowledge-based systems supported by knowledge artifacts.	Yes: 2 projects in two companies RM: multiple-case

Kokkonniemi [S11] discusses about documentation, using the concept of knowledge artifacts during the iterative development of software, and more specifically artifacts based on knowledge experience. The author revisits the concept of experience and experience knowledge [53] in order to propose the review points in XP (especially in inspection points and inspection processes) where these types of artifacts can be used based on experience. The idea is to show how experience knowledge-based artifacts can be linked to software development processes. Also, how experience knowledge can be put into practice in the context of iterative software development processes. This author defines that experience knowledge is based on organizational memory, which is mainly based on human experiences. The experience is appointed as tacit knowledge and experience knowledge is appointed as explicit knowledge. The knowledge artifacts are generated during the process of transforming experience into experience knowledge. Therefore, the transformation of tacit knowledge in explicit knowledge (experience knowledge) uses or generates an artifact-based knowledge. This study presents "one form reasonable and profitable to link extreme programming and experience knowledge" as being an improvement in knowledge management. This author also found some resistance in the companies participating in the research because they "do not want to add new tasks in agile methods".

Salazar-Torres *et al.* [S20] also discusses the artifacts of knowledge, bringing a more focused definition to the field of artificial intelligence, as the focus of their work is to develop knowledge-based systems using knowledge artifacts. In this study four definitions of a knowledge artifact are presented: an artifact which represents knowledge; an artifact which represents an encoding of knowledge; an artifact whose creation requires specialized knowledge and; an artifact made up of knowledge. Four perspectives on the usefulness of a knowledge artifact are also presented: "knowledge management perspective; community support perspective; computational perspective and; artificial intelligence perspective". These authors propose an agile knowledge engineering method to develop the KA-based systems, named KABS, which is a knowledge artifact embodied in the form of a software system. Two case studies of the KA-based systems are presented. The first was for the community of rubber compounders, in which the domain was the trucks business unit of Pirelli Tires. The second case study was implemented in a large manufacturing company in Brazil, in order to help in the diagnostics and repair of complex electromechanical devices.

We realize that the use of knowledge artifacts and concepts like experience knowledge, can add benefits to the software development process by presenting an approach more similar to the knowledge creation and sharing theories, but it may generate difficulties in agile projects by adding new tasks to the process.

Table 11 summarizes the main features of the primary studies, beyond those already presented in previous sections

Table 11 - Summary of main features of the primary studies

Paper / Year	Empirical evaluation	Source	Implications in
[S1] 2004	Yes: 28 computer professionals and students	(Conference) IEEE ADC - Agile Development Conference	Ways of communication; Human and social factors; Knowledge representation forms
[S2] 2003	No	(Conference) Lecture Notes in Computer Science - LSO 2002	Tools for knowledge management; Ways of communication
[S3] 2003	No	(Conference) IEEE WET ICE - Infrastructure for Collaborative Enterprises	Tools for knowledge management; Ways of communication
[S4] 2003	Yes: 9 software development companies	(Conference) Lecture Notes in Computer Science - LSO 2002	Software development cycle and artifacts; Knowledge representation forms
[S5] 2006	No	(Conference) Lecture Notes in Computer Science - ADVIS 2006	Software development cycle and artifacts
[S6] 2009	Yes: 2 virtual agile teams; 22 professionals	(Conference) ICEIS - International Conference on Enterprise Information Systems	Ways of communication; Human and social factors; Knowledge representation forms
[S7] 2005	Yes: students and Masters of technologies of software	(Conference) ACM - Symposium on Applied Computing	Ways of communication; Human and social factors
[S8] 2006	Yes: 3 software development companies; 18 professionals	(Journal) Communications of the ACM	Ways of communication; Software development cycle and artifacts
[S9] 2003	No	(Conference) Euromicro Conference	Software development cycle and artifacts
[S10] 2007	Yes: agile software development team	(Conference) HICSS - Hawaii International Conference on System Sciences	Software development cycle and artifacts
[S11] 2008	No	(Conference) HICSS - Hawaii International Conference on System Sciences	Software development cycle and artifacts; Knowledge representation forms; ways of communication
[S12] 2005	Yes: 2 projects, an agile and another waterfall; 14 professionals	(Conference) ICICT - International Conference on Information and Communication Technology	Software development cycle and artifacts
[S13] 2009	No	(Journal) Decision Support Systems	Software development cycle and artifacts
[S14] 2005	Yes: web software development team	(Conference) ADVIS - International Conference on Advances in Information System	Tools for knowledge management; Human and social factors
[S15] 2005	No	(Conference) EUROCON - International Conference on Computer as a Tool	Tools for knowledge management; Ways of communication
[S16] 2004	No	(Conference) Lecture Notes in Computer Science	Tools for knowledge management; Ways of communication
[S17] 2005	Yes: 2 agile projects; 18 programmers	(Conference) HSSE Workshop on Human and social factors of software engineering	Human and social factors
[S18] 2009	Yes: 2 studies; 30 experienced project managers	(Journal) Journal of Systems and Software	Human and social factors; Software development cycle and artifacts
[S19] 2004	No	(Conference) Agile Development Conference	Ways of communication; Software development cycle and artifacts
[S20] 2008	Yes: 2 projects in two companies	(Journal) Knowledge-Based Systems	Knowledge representation forms; Software development cycle and artifacts

Apêndice C – Formulário do *survey* aplicado no estudo de viabilidade

* Neste Apêndice algumas informações foram ocultadas, tanto na ontologia como nas reuniões, para evitar a identificação das organizações que participaram do estudo de caso

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

Ensaio para avaliação dos termos de uma ontologia do
domínio comparados aos conceitos extraídos das reuniões de projeto

Aluno: Anderson Yanzer

Professor Orientador: Marcelo Blois

Sumário

1. Instruções para o ensaio	159
2. Glossário	160
3. Reuniões	160
3.1. Reunião 1 (17/05/10)	161
3.2. Avaliação da Reunião 1	161
3.3. Reunião 2 (07/06)	162
3.4. Avaliação da Reunião 2	162
3.5. Reunião 3 (05/07)	162
3.6. Avaliação da Reunião 3	164

1. Instruções para o ensaio

Este ensaio faz parte do processo de validação experimental de uma metodologia de aquisição de conhecimento de reuniões de projeto de desenvolvimento de software. Esta metodologia está sendo desenvolvida no projeto de doutorado do aluno (supracitado) e prevê o uso de técnicas e ferramentas de Gestão do Conhecimento e PLN (Processamento de Línguas Naturais) para o processo de aquisição e validação do conhecimento compartilhado nas reuniões.

O objetivo deste ensaio é comparar os “conceitos relevantes” identificados de forma “automática” em uma reunião através do uso da nossa metodologia proposta com os “conceitos relevantes” da reunião que seriam apontados através da percepção humana.

O ensaio consistirá em, a partir de uma lista de termos, analisar as reuniões e apontar os termos relevantes para os tópicos tratados em cada reunião. Estes termos relevantes seriam os conceitos associados à reunião que serviriam, por exemplo, como indexadores para auxiliar na identificação e localização das reuniões.

O ensaio é composto pelos seguintes passos:

- I. Analisar o glossário fornecido para um melhor entendimento das siglas e termos específicos do domínio que foi tratado nas reuniões. O glossário está na seção 2 deste roteiro.
- II. Analisar cada uma das reuniões realizadas sobre o projeto (três reuniões) que estão descritas na seção 3 deste trabalho. Ler todas as reuniões (itens 3.1, 3.3 e 3.5) para depois realizar o passo III.
- III. Apontar na Planilha de avaliação os termos relevantes de cada reunião. Esta planilha apresenta na primeira coluna os termos e a avaliação a ser realizada consiste em apontar para cada reunião específica quais são os termos relevantes associados.

Aponte seu nível de leitura na língua Inglesa: () Excelente () Muito bom () Bom () Regular () Fraco

*O tempo médio estimado para realização deste ensaio é trinta minutos.

**Os dados obtidos através deste ensaio serão utilizados somente no contexto desta pesquisa.

2. Glossário

Term	Description
Commodity	Indica o tipo de componente, como um disco rígido, memória, motherboard, etc.
DDW	Data Warehouse da companhia ABC.
DES	Interface das despesas.
DSP	Descrição do Produto.
ENF	Estorno de Nota fiscal.
FDR (WTCS)	Sistema "Track Code", responsável por localizar e gerenciar o fluxo de produção.
Functional Requirements	Requisitos técnicos de software que compõem o sistema, que descrevem ações que o sistema deve ser capaz de executar, ou em outras palavras, o que o sistema deve fazer.
	Sistema de configuração da gestão e produção de contratos utilizados pelos grupos internos da empresa ABC para fornecer informação de materiais relacionados com a fábrica e cadeias de abastecimento da empresa ABC.
	Sistema responsável pela entrada de número do componente (part number) na linha de produção.
	Interface de cálculo da taxa.
	Linha de Negócio, que indica o tipo de produto: servidor, notebook e desktop.
	Composição dos modelos.
	Sistema que controla o livro fiscal e as obrigações fiscais para o Brasil.
	Plano de vendas Master.
	Interface Movement.
	Notas Fiscais de Entrada.
	Notas Fiscais de Saída.
	Notas Fiscais de transferência.
Non-functional requirements	Requisitos técnicos de software que compõem o sistema, que descrevem atributos que o sistema deve possuir ou restrições sob os quais ele deve operar.
<i>Nota Fiscal</i>	Um tipo de recibo fiscal proposto no Brasil.
OPR	Ordem de produção.
PAR	Interface "Partner".
PN - Part Number	Identifica os itens de configuração que são utilizados para a montagem de um produto como memórias, disco rígido, placa wireless, etc.
PEA	Embalagem do produto.
PIS/COFINS	PIS significa "Programa de Integração Social" e COFINS representa "Contribuição para o Financiamento da Seguridade Social". Estes são impostos incidentes sobre receitas das empresas no Brasil.
PPB	Lei Brasileira para o processo de produção básica.
PPID	Número serial do "Part Number". Usado no sistema para controlar se um componente é local ou importado.
PRD	Interface de Produto.
Pre-Staging Database	Banco de dados que servirá como armazenagem intermediária de dados do PPB e RECOF, antes que ele seja inserido no DDW. Esta base de dados servirá também como fonte de dados para geração de interface RECOF.
RECOF	Significa "Regime Aduaneiro de Entrepasto Industrial sob Controle informatizado", que é uma Lei que parametriza as importações Brasileiras.
SKU	Stock Keeping Unit - identifica um produto vendável
SLD	Balanço.
SME	O especialista no assunto/domínio.
Synchro	Sistema de gera notas fiscais e calcula as taxas.

3. Reuniões

3.1 Reunião 1 (17/05/10): as falas na reunião são de pessoas que fazem parte de uma equipe de desenvolvimento de software discutindo sobre um aplicativo que está sendo desenvolvido.

Adriana: If we are all here, let's get started. Today we will speak about the project PPB-RECOF. This is a ABC Company's project. Firstly, we list the participants of today's meeting: Adriana, Jaque and Anderson.

The project is the PPB-RECOF and the client is ABC Company

Jaque: Thank you for having me, I'm looking forward to today's meeting. I have some news about the project.

Anderson: Hi, I have some news about the project. I hope that we have an active meeting.

Adriana: Anderson, I hope so too. OK, let's start.

Adriana: people, we will review the last meeting. Jaque, please...

*** o restante do texto da reunião foi ocultado ***

3.2 Avaliação da Reunião 1

Depois de realizada a leitura da Reunião 1 deverão ser apontados na planilha a seguir (marcar com "X") os termos relevantes, que seriam os conceitos associados à reunião e que serviriam, por exemplo, como indexadores para auxiliar na identificação e localização desta reunião.

Termos da Ontologia	Termo relevante
Brazilian_Law	
Brazilian_Tax	
Chassis	
COFINS	
Commodity	
Company	
CPU	
D3	
Database	
Datawarehouse	
Desktop	
FDL	
Fiscal	
Functional_Requirement	
Glovia	
GRM	
Hard_Drive	
IMC	
Interface	
IPI	
IT_Admin	
Manufactory	
Memory	
Modem	
Monoprocessor	
Motherboard	
Multiprocessor	
Non-Functional_Requirement	
Nota_Fiscal	
Notebook	
Operational_System	
Other_Line	
PIS	
PPB	
PPB_Admin	
PPB_Quota	
PPB_User	

Product	
Product_Line	
Product_Part	
Production_Order	
Project	
RECOF	
RECOF_Admin	
RECOF_User	
RecofSys	
Report	
Requirement	
Role	
Sales_System	
Server	
SoftWay	
Synchro	
System_Type	
Track_Code	
TRD	
User	
Video_Card	
Windows	

3.3 Reunião 2 (07/06): as falas na reunião são de pessoas que fazem parte de uma equipe de desenvolvimento de software discutindo sobre um aplicativo que está sendo desenvolvido.

Adriana: If we are all here, let's get started. Firstly, we list the participants of today's meeting: Adriana, Jaque and Anderson.
The project is the PPB-RECOF
The client is ABC Company

Jaque: Thank you for having me and I'd like to apologize for my voice, I'm still hoarse.

Anderson: I took the flu shot, but I was not bad. I was lucky! Ahhhh for about three days my arm was hurting, just it... I hope that we have an active meeting.

Adriana: Anderson, I hope so too. OK, let's start. people, we will review the last meeting. Jaque, please...

Jaque: In the last meeting we reviewed the concepts about the requirements and we covered the following functional requirements:

*** o restante do texto da reunião foi ocultado ***

3.4 Avaliação da Reunião 2

Depois de realizada a leitura da Reunião 2 deverão ser apontados na planilha a seguir (marcar com "X") os termos relevantes, que seriam os conceitos associados à reunião e que serviriam, por exemplo, como indexadores para auxiliar na identificação e localização desta reunião.

Termos da Ontologia	Termo relevante
Brazilian_Law	
Brazilian_Tax	
Chassis	
COFINS	
Commodity	
Company	
CPU	
D3	
Database	
Datawarehouse	
Desktop	
FDL	
Fiscal	

Functional_Requirement	
Glovia	
GRM	
Hard_Drive	
IMC	
Interface	
IPI	
IT_Admin	
Manufactory	
Memory	
Modem	
Monoprocessor	
Motherboard	
Multiprocessor	
Non-Functional_Requirement	
Nota_Fiscal	
Notebook	
Operational_System	
Other_Line	
PIS	
PPB	
PPB_Admin	
PPB_Quota	
PPB_User	
Product	
Product_Line	
Product_Part	
Production_Order	
Project	
RECOF	
RECOF_Admin	
RECOF_User	
RecofSys	
Report	
Requirement	
Role	
Sales_System	
Server	
SoftWay	
Synchro	
System_Type	
Track_Code	
TRD	
User	
Video_Card	
Windows	

3.5 Reunião 3 (05/07): as falas na reunião são de pessoas que fazem parte de uma equipe de desenvolvimento de software discutindo sobre um aplicativo que está sendo desenvolvido.

Adriana:

If we are all here, let's get started. Firstly, we list the participants of today's meeting: Adriana, Jaque and Anderson.

Jaque: People, today my voice is better and I hope that our meeting will be nice.

Anderson: Hi, I'm a little sad today, because of world cup, but the Germany won and the things are better.

Adriana: OK! We've found a guilty, the player felipe. There are more...but... Well, now let's go back to work.

The project is the PPB-RECOF. The client is ABC Company people, we will review the last meeting. Jaque, please...

Jaque: In the last meeting we reviewed the concepts about the requirements and we covered the following functional requirements:

*** o restante do texto da reunião foi ocultado ***

3.6 Avaliação da Reunião 3

Depois de realizada a leitura da Reunião 3 deverão ser apontados na planilha a seguir (marcar com “X”) os termos relevantes, que seriam os conceitos associados à reunião e que serviriam, por exemplo, como indexadores para auxiliar na identificação e localização desta reunião.

Termos da Ontologia	Termo relevante
Brazilian_Law	
Brazilian_Tax	
Chassis	
COFINS	
Commodity	
Company	
CPU	
D3	
Database	
Datawarehouse	
Desktop	
FDL	
Fiscal	
Functional_Requirement	
Glovia	
GRM	
Hard_Drive	
IMC	
Interface	
IPI	
IT_Admin	
Manufactury	
Memory	
Modem	
Monoprocessor	
Motherboard	
Multiprocessor	
Non-Functional_Requirement	
Nota_Fiscal	
Notebook	
Operational_System	
Other_Line	
PIS	
PPB	
PPB_Admin	
PPB_Quota	
PPB_User	
Product	
Product_Line	
Product_Part	
Production_Order	
Project	
RECOF	
RECOF_Admin	

RECOF_User	
RecofSys	
Report	
Requirement	
Role	
Sales_System	
Server	
SoftWay	
Synchro	
System_Type	
Track_Code	
TRD	
User	
Video_Card	
Windows	

Apêndice D – Informações dos ensaios realizados no estudo de viabilidade

Informações da Reunião 1

Reunião 1

Dados gerais:

Participantes: 3

Duração: 08 minutos

Idioma: Inglês

Quantidade de palavras identificadas pelo tagger no texto da reunião: 1057

Quantidade de palavras identificadas pelo tagger como conceitos: 348

WER na transcrição da fala de todos participantes: 12,2%

WER na transcrição da fala de um dos participantes: 8,7%

Arquivos gerados:

- roteiro da reunião com a fala de todos participantes (txt)
- áudio da reunião com fala de todos os participantes (mp3)
- áudio da reunião com fala de um dos participantes (mp3)
- transcrição do áudio da fala de todos os participantes (txt)
- transcrição do áudio da fala de um dos participantes (txt)
- texto etiquetado da transcrição do áudio da fala de todos os participantes (txt)
- texto etiquetado do roteiro da reunião (txt)
- planilha com os conceitos extraídos do texto etiquetado da transcrição (xls)
- planilha com os conceitos extraídos do texto etiquetado do roteiro da reunião (xls)
- planilha com a análise da comparação dos conceitos do roteiro da reunião com os da transcrição (xls)

Resultados:

Conceitos com frequência ≥ 2

Quantidade no texto da reunião	58
Quantidade que foram identificados na transcrição	49
Porcentagem de acerto	84,5%

Conceitos com frequência ≥ 3

Quantidade no texto da reunião	36
Quantidade que foram identificados na transcrição	35
Porcentagem de acerto	97,2%

Conceitos com frequência ≥ 4

Quantidade no texto da reunião	27
Quantidade que foram identificados na transcrição	27
Porcentagem de acerto	100,0%

Conceitos com frequência ≥ 5

Quantidade no texto da reunião	19
Quantidade que foram identificados na transcrição	19
Porcentagem de acerto	100,0%

* A frequência é o número de vezes em que o conceito aparece no texto.

Reunião 2

Dados gerais:

Participantes: 3

Duração: 11 minutos

Idioma: Inglês

Quantidade de palavras identificadas pelo tagger no texto da reunião: 1128

Quantidade de palavras identificadas pelo tagger como conceitos: 455

WER na transcrição da fala de todos participantes: 11,5%

WER na transcrição da fala de um dos participantes: 7,9%

Arquivos gerados:

- roteiro da reunião com a fala de todos participantes (txt)
- áudio da reunião com fala de todos os participantes (mp3)
- áudio da reunião com fala de um dos participantes (mp3)
- transcrição do áudio da fala de todos os participantes (txt)
- transcrição do áudio da fala de um dos participantes (txt)
- texto etiquetado da transcrição do áudio da fala de todos os participantes (txt)
- texto etiquetado do roteiro da reunião (txt)
- planilha com os conceitos extraídos do texto etiquetado da transcrição (xls)
- planilha com os conceitos extraídos do texto etiquetado do roteiro da reunião (xls)
- planilha com a análise da comparação dos conceitos do roteiro da reunião com os da transcrição (xls)

Resultados:

Conceitos com frequência \geq a 2

Quantidade no texto da reunião	70
Quantidade que foram identificados na transcrição	57
Porcentagem de acerto	81,4%

Conceitos com frequência \geq a 3

Quantidade no texto da reunião	48
Quantidade que foram identificados na transcrição	40
Porcentagem de acerto	83,3%

Conceitos com frequência \geq a 4

Quantidade no texto da reunião	30
Quantidade que foram identificados na transcrição	24
Porcentagem de acerto	80,0%

Conceitos com frequência \geq a 5

Quantidade no texto da reunião	20
Quantidade que foram identificados na transcrição	16
Porcentagem de acerto	80,0%

* A frequência é o número de vezes em que o conceito aparece no texto.

Informações da reunião 3

Reunião 3**Dados gerais:**

Participantes: 3

Duração: 09 minutos

Idioma: Inglês

Quantidade de palavras identificadas pelo tagger no texto da reunião: 854

Quantidade de palavras identificadas pelo tagger como conceitos: 353

WER na transcrição da fala de todos participantes: 10,2%

WER na transcrição da fala de um dos participantes: 7,6%

Arquivos gerados:

- roteiro da reunião com a fala de todos participantes (txt)
- áudio da reunião com fala de todos os participantes (mp3)
- áudio da reunião com fala de um dos participantes (mp3)
- transcrição do áudio da fala de todos os participantes (txt)
- transcrição do áudio da fala de um dos participantes (txt)
- texto etiquetado da transcrição do áudio da fala de todos os participantes (txt)
- texto etiquetado do roteiro da reunião (txt)
- planilha com os conceitos extraídos do texto etiquetado da transcrição (xls)
- planilha com os conceitos extraídos do texto etiquetado do roteiro da reunião (xls)
- planilha com a análise da comparação dos conceitos do roteiro da reunião com os da transcrição (xls)

Resultados:**Conceitos com frequência \geq a 2**

Quantidade no texto da reunião	58
Quantidade que foram identificados na transcrição	50
Porcentagem de acerto	86,2

Conceitos com frequência \geq a 3

Quantidade no texto da reunião	43
Quantidade que foram identificados na transcrição	40
Porcentagem de acerto	93,0

Conceitos com frequência \geq a 4

Quantidade no texto da reunião	34
Quantidade que foram identificados na transcrição	31
Porcentagem de acerto	91,2%

Conceitos com frequência \geq a 5

Quantidade no texto da reunião	17
Quantidade que foram identificados na transcrição	16
Porcentagem de acerto	94,1%

* A frequência é o número de vezes em que o conceito aparece no texto.

Reunião 4

Dados gerais:

Participantes: 3

Duração: 06 minutos

Idioma: Inglês

Quantidade de palavras identificadas pelo tagger no texto da reunião: 596

Quantidade de palavras identificadas pelo tagger como conceitos: 226

WER na transcrição da fala de todos participantes: 11,0%

WER na transcrição da fala de um dos participantes: 7,4%

Arquivos gerados:

- roteiro da reunião com a fala de todos participantes (txt)
- áudio da reunião com fala de todos os participantes (mp3)
- áudio da reunião com fala de um dos participantes (mp3)
- transcrição do áudio da fala de todos os participantes (txt)
- transcrição do áudio da fala de um dos participantes (txt)
- texto etiquetado da transcrição do áudio da fala de todos os participantes (txt)
- texto etiquetado do roteiro da reunião (txt)
- planilha com os conceitos extraídos do texto etiquetado da transcrição (xls)
- planilha com os conceitos extraídos do texto etiquetado do roteiro da reunião (xls)
- planilha com a análise da comparação dos conceitos do roteiro da reunião com os da transcrição (xls)

Resultados:

Conceitos com frequência \geq a 2

Quantidade no texto da reunião	45
Quantidade que foram identificados na transcrição	36
Porcentagem de acerto	80,0

Conceitos com frequência \geq a 3

Quantidade no texto da reunião	23
Quantidade que foram identificados na transcrição	21
Porcentagem de acerto	91,3

Conceitos com frequência \geq a 4

Quantidade no texto da reunião	13
Quantidade que foram identificados na transcrição	12
Porcentagem de acerto	92,3%

Conceitos com frequência \geq a 5

Quantidade no texto da reunião	11
Quantidade que foram identificados na transcrição	10
Porcentagem de acerto	90,9%

* A frequência é o número de vezes em que o conceito aparece no texto.

Apêndice E – Formulário do *survey* aplicado no estudo de caso explanatório

* Neste Apêndice algumas informações foram ocultadas, tanto na ontologia como nas reuniões, para evitar a identificação da organização que participou do estudo de caso

** O instrumento foi aplicado para as dez reuniões que compuseram o estudo de caso, mas neste apêndice será mostrado todo o processo para apenas a primeira reunião.

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

Ensaio para avaliação dos termos de uma ontologia do
domínio comparados aos conceitos extraídos das reuniões de projeto

Aluno: Anderson Yanzer

Professor Orientador: Marcelo Blois

1. Instruções para o ensaio

Este ensaio faz parte do processo de validação experimental de uma metodologia de aquisição de conhecimento de reuniões de projeto de desenvolvimento de software. Esta metodologia está sendo desenvolvida no projeto de doutorado do aluno (supracitado) e prevê o uso de técnicas e ferramentas de Gestão do Conhecimento e PLN (Processamento de Línguas Naturais) para o processo de aquisição e validação do conhecimento compartilhado nas reuniões.

O objetivo deste ensaio é comparar os “conceitos relevantes” identificados de forma “automática” em uma reunião através do uso da nossa metodologia proposta com os “conceitos relevantes” da reunião que seriam apontados através de percepção humana.

O ensaio será realizado com os participantes das reuniões ocorridas no estudo de caso na empresa XXXXXXX, que aborda a atualização de um módulo do aplicativo XXXXXXX, desenvolvido por esta empresa. Este ensaio consistirá em, a partir de uma lista de termos extraídos da ontologia do domínio, os participantes deverão apontar quais destes termos são relevantes para identificar os tópicos tratados em cada reunião. Estes termos relevantes seriam os conceitos associados à reunião que serviriam como indexadores para auxiliar na identificação e localização das reuniões.

O ensaio é composto pelos seguintes passos:

- É oferecido ao participante o áudio e um texto com a transcrição do áudio de cada reunião para que o mesmo relembre dos assuntos tratados nas reuniões que ele participou.

- O participante deve apontar na Planilha de avaliação os termos relevantes de cada reunião. Esta planilha apresenta na primeira coluna os termos da ontologia e a avaliação a ser realizada consiste em apontar para cada reunião específica quais são os termos relevantes associados.

*O tempo médio estimado para realização deste ensaio é uma hora e trinta minutos.

**Os dados obtidos através deste ensaio serão utilizados somente no contexto desta pesquisa.

2. Reuniões

2.1 Reunião 1: Relatórios do XXXXXX

- Reunião do dia vinte de junho de 2011, com os seguintes participantes: AAAAA, BBBBBB, CCCCCC e DDDDDDD.
- Bem, vamos começar falando do relatório de mesas abertas, onde o sistema lista,... vai listar todos os relatórios que contém movimentação, que é as mesas que estão abertas no momento, que tem itens né, na mesa?
- sim.
- Neles sai só a data em que a mesa foi aberta, o horário, e...
- é, vai ser listado justamente isso: a mesa, o número da mesa, a data em que foi aberta e o horário. Que mais que também poderia botar nesse relatório?
- Ah Poderia botar nesse relatório, quem sabe seria interessante, botar o garçom que está atendendo a mesa.
- Sim.
- Seria uma boa colocar.
- Seria uma boa possibilidade.
- Sim? Até se o cara pode pegar essa informação da classe de pedidos onde tem, já tem essa descrição, já tem a ... a descrição não, já tem um atributo informando qual é o garçom que está... está atendendo aquela mesa, correto? Já tem como fazer....
- é,... temos que lembrar também que pelo PAF esse é um relatório obrigatório, então, temos que...
- É, seguir as regras né.
- Aham, seguir exatamente as regras do PAF.
- É, saindo o horário... data, horário... e se a mesa está aberta. Se a mesa....
- vai levar... Esse relatório vai levar em consideração a classe de mesas que define quais, qual são as mesas que o sistema está trabalhando e também a classe de pedidos, que é onde vai definir se a mesa está aberta ou não.
- é, tem o status da mesa né.
- é, lembrando que não pode ter uma mesa aberta que não esteja sendo utilizada no sistema, tipo, na classe mesas que lista as mesas utilizadas pelo sistema tem que está todas as mesas que vão ser utilizadas pelo sistema, tanto em todas...
- Sim, todas as mesas né?
- sim.
- Sim, mas aqui no caso é só as mesas abertas mesmo, que é com base na classe pedido. Se na classe de pedidos constar que aquela mesa ainda está aberta ele, vai sair nesse relatório que é... no caso, a mesa ainda não foi encerrada, ainda tem pedido nela.
- Outra coisa também que a gente tem que levar em consideração é que ao imprimir a leitura Z e uma mesa estiver aberta, este relatório tem que ser impresso antes de imprimir a leitura Z.
- Ah sim, tem que sair antes da Leitura Z porque aí tem que encerrar a mesa, ele é obrigado a encerrar a mesa. É, acho que esse relatório de mesas abertas, até, ele é bem simples, não tem muito... vai informar quais as mesas estão abertas mesmo, não tem nada de mais.
- só o importante é a parte da emissão da leitura Z mesmo, que esse tem que ter pelo PAF, então não pode deixar pra traz.
- sim, e não esquecer também de colocar no relatório a data e o horário em que a mesa foi aberta. Isso aí é importante sair no relatório.
- Beleza! O relatório tá legal.
- o outro é a conferência de mesas, que é um relatório onde vai... semelhante ao cupom, onde sai todos os itens que tem naquela mesa, todos os itens que ainda não foram pagos né, não foram vendidos e, também, entra em classe de mesa e classe de pedidos, e aí tem itens de pedido, que herda da classe de produtos lá o...
- os dados que a gente vai precisar para esse relatório que são: o código do produto, a descrição do produto, a quantidade, o valor unitário desse produto, o total do produto.
- é, o total tu pode calcular da quantidade vezes o valor unitário.
- lembrando também que os produtos que estiverem cancelados nessa mesa também devem aparecer nesse relatório.
- semelhante ao cupom fiscal.
- tem que sair exatamente o que vai sair no cupom tem que sair ali, tem que mostrar exatamente o que vai pro cliente pra depois não dar nenhum problema.
- Interessante também é sair que cobra o couvert e os 10% do garçom. A porcentagem do garçom, também não vai ser obrigado a ser 10%.
- sim até porque o sistema pode configurar a quantidade da porcentagem.

- faz tudo configurado, se o cliente quer que apareça no relatório ou não, mas que é interessante que saia... é uma boa sim, que tenha essa informação no relatório.
- Sim, até porque na nota vai ser cobrado, e se for cobrado comissão ou couvert vai aparecer na nota lá que foi cobrado e então a descrição tem que ser exata.
- e tem que colocar também, conforme o PAF, tem que ter uma mensagem de 'aguardando a emissão do cupom fiscal'. Porque na real ele não é um comprovante de venda é só de conferência de mesa, um relatório não fiscal vai sair na ECF.
- assim como essa mensagem tem que aparecer na conferência de mesa, se for impresso uma conferência de mesa para a mesa 10, por exemplo, ao imprimir o cupom dessa mesa 10 tem que aparecer no cupom que foi gerado a conferência dessa mesa.
- só uma dúvida: voltando ao relatório de mesas abertas, se tiver mesas juntas nesse relatório também é interessante sair as mesas juntas né?
- Sim, as mesas abertas que estão juntas com a mesa central, digamos assim. Até porque o sistema trata como essas mesas juntas estão abertas, então é interessante que saiam como abertas.
- Então tá, beleza. E daí, depois, tem o relatório de transferência entre mesas.
- só um pouquinho: na conferência de mesas também é interessante que tu vai botar a mesa que tu vai querer fazer essa conferência, isso é necessário. E o sistema também dá uma opção, muito interessante, que o usuário pode ou não querer utilizar, que é a opção de definir o número de pessoas, de pagantes, da mesa; no caso, pode dividir o valor total da mesa pelo número de pagantes.
- interessante, daí já vai sair um total para cada um.
- sim, porque é muito comum mais de uma família numa mesa e três pagantes...
- poder dividir a conta. Isso é interessante mesmo...
- bem mais prático já.
- Interessante mesmo, bem comum isso aí acontecer.
- é, até teria um diferencial para o sistema porque normalmente não vem isso pro sistema, é bem legal.
- seria uma boa mesmo. Bom, acho que nesse relatório também... é isso. A maior parte do sistema é em cima da classe de mesas e pedidos, então agora vamos pro relatório de transferência entre mesas. É um relatório informando os produtos que saíram de uma mesa e foram para uma outra mesa. Por exemplo: saiu da mesa cinco pra mesa 10, tem que ter um relatório onde informe de onde é que veio aquele produto, não se originou direto na mesa 10. Então é um relatório que fale de onde vem o produto, a quantidade transferida, o valor...
- até, pelas regras do PAF, é obrigatório botar no complemento essa observação: transferido da mesa tal para a mesa tal, é obrigatório botar essa descrição caso for transferido. É uma regra que a gente é obrigado a seguir.
- não tem como evitar. E é uma coisa bem interessante, é uma informação muito boa, tirar um relatório e ver de onde veio aquele produto, saber de onde é que se originou e tal.
- e assim como conferência de mesas, os dados do produto são os mesmos que são listados lá: código, descrição...
- também com base na classe de mesas e pedidos.
- e, no caso, se eu transferi da mesa 8 pra mesa 10 e também depois transferi esse mesmo produto que tá na mesa 10 para a mesa 20, por exemplo, eu transferi o mesmo produto de uma mesa para a segunda mesa e da segunda mesa para a terceira mesa. Qual seria a informação que deveria aparecer no produto?
- no produto aparece a transferência da última mesa.
- da penúltima até a última. Exato.
- é isso aí. E esses relatórios a gente também pode criar para a comanda, seguir a mesma regra para a comanda. Na classe comanda segue a mesma idéia, que é mesas, só que muda o código de identificação, que no caso da comanda é o código de barra. Também vai utilizar a classe de comandas e a classe de pedidos.
- vão ter métodos bem parecidos mas ...
- é, na forma de trabalhar são semelhantes, mas vai tratar conforme é cada um. Nos relatórios dá pra seguir a mesma idéia da comanda; seria uma conferência de comandas, comandas abertas e transferência entre comandas...
- até porque o sentido das comandas é basicamente o mesmo das mesas, é só uma maneira diferente de tratar as vendas.
- é. Até teria, tipo, as comandas abertas eu acho que não teria uma necessidade nesse relatório.
- sim, porque automaticamente uma comanda feita ela....
- não, até é interessante. É interessante ter esse relatório porque como ela vai ser uma comanda eletrônica...
- ... pode ser que ela esteja aberta mas ela ainda não foi paga.
- Exato. É interessante ter... seria comandas abertas.
- uma coisa também que poderia aparecer na conferência de mesas já que lista todos os produtos, é aparecer de que mesa ela foi transferida também, já que é uma mensagem útil pra conferência da mesa. Até porque se foi transferido um produto para uma mesa tal tu vai lá e gera conferência de mesas e o cliente pode perguntar: de onde é que veio esse produto? Veio da mesa tal pra tua mesa. Não necessariamente gerar dois relatórios...
- saindo direto tudo no conferências, né?
- Isso.
- tanto pra... e a mesma idéia pra comanda. Pra comanda a gente faz a mesma coisa.
- sim. Mesmo sentido. Mais alguma coisa pra falar na questão destes relatórios?
- não, acho que nesses relatórios é isso aí. Já que a gente vai usar tanto pra mesa, quanto pra comanda, usando a classe de comandas e a classe de mesas, junto com a classe de pedidos né, onde é o pedido que vai identificar os status de cada um, tanto da comanda quanto da mesa. Acho que nesse relatório é isso.

2.2 Avaliação da Reunião 1

Depois de revisada a Reunião 1 deverão ser apontados na planilha a seguir (marcar com “X”) os termos relevantes, que seriam os conceitos associados à reunião e que serviriam, por exemplo, como indexadores para auxiliar na identificação e localização desta reunião.

Termos da Ontologia	Termo relevante
Bairro	
Balança	
CEP	
Cidade	
Cliente	
CNPJ	
Código da Entrega	
Código de Barras	
Código do Produto	
Comanda	
CPF	
Data	
Data de aniversário	
Data_Hora	
Descrição	
Endereço	
Entrega	
Garçom	
ID do veículo	
IP de Comunicação	
Mesa	
Modelo	
Moto-boy	
Nome	
Número da Comanda	
Número da Mesa	
Número do pedido	
Observação	
Pedido	
Produto	
Quantidade	
Restaurante	
RG	
Reserva	
Status da Mesa	
Status do Pedido	
Taxa de transmissão	
Telefone	
Taxa de Comissão	
Tipo de Comissão	
Tipo de Comunicação	
Tipo de pedido	
Tipo de Produto	
Transferida	
UF	
Valor Total	
Valor Unitário	
Veículo	