

UNIVERSIDADE DE CAXIAS DO SUL
PROGRAMA DE PÓS-GRADUAÇÃO EM ADMINISTRAÇÃO - PPGA
CURSO DE MESTRADO EM ADMINISTRAÇÃO

RAFAEL POLTRONIERI PANOZZO

**A DIMENSÃO TÁCITA DO CONHECIMENTO NA DEFINIÇÃO DOS
REQUISITOS EM UMA FÁBRICA DE *SOFTWARE* DA SERRA GAÚCHA**

CAXIAS DO SUL

2012

RAFAEL POLTRONIERI PANOZZO

**A DIMENSÃO TÁCITA DO CONHECIMENTO NA DEFINIÇÃO DOS
REQUISITOS EM UMA FÁBRICA DE *SOFTWARE* DA SERRA GAÚCHA**

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Administração da Universidade de Caxias do Sul, como parte dos requisitos necessários para a obtenção do título de Mestre em Administração, Área de Concentração: Gestão da Inovação e Competitividade.

Orientadora: Prof^a. Dr^a. Ana Cristina Fachinelli

CAXIAS DO SUL

2012

“A Dimensão tácita do conhecimento nas decisão dos requisitos em uma fábrica de Software da Serra Gaúcha.”

Rafael Poltronieri Panozzo

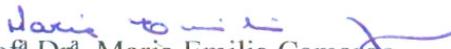
Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Administração da Universidade de Caxias do Sul, como parte dos requisitos necessários para a obtenção do título de Mestre em Administração, Área de Concentração: Administração da Produção.

Caxias do Sul, 28 de junho de 2012.

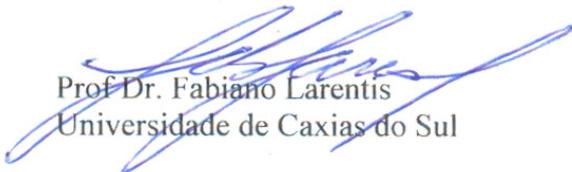
Banca Examinadora:



Prof. Dr. Ana Cristina Fachinelli(Orientadora)
Universidade de Caxias do Sul



Profª Drª. Maria Emilia Camargo
Universidade de Caxias do Sul



Prof Dr. Fabiano Larentis
Universidade de Caxias do Sul



Profª. Drª. Ana Elizabeth Moiseichyk
Universidade Federal de Santa Maria

Dados Internacionais de Catalogação na Publicação (CIP)
Universidade de Caxias do Sul
UCS - BICE - Processamento Técnico

P195d Panozzo, Rafael Poltronieri

A Dimensão tácita do conhecimento na definição dos requisitos em uma fábrica de *software* da serra gaúcha / Rafael Poltronieri Panozzo. - 2012.

138 f. : il ; 30 cm

Apresenta bibliografia.

Orientador: Prof^a. Dr^a. Ana Cristina Fachinelli

Dissertação (Mestrado) ó Universidade de Caxias do Sul, Programa de Pós-Graduação do Mestrado em Administração, 2012.

1. Gestão do conhecimento. 2. Engenharia de software.
3. Software ó Desenvolvimento. 4. Satisfação do consumidor. I.
Título.

CDU 2.ed.: 005.94

Índice para o catálogo sistemático:

1. Gestão do conhecimento	005.94
2. Engenharia de software	004.41
3. Software ó Desenvolvimento	004.415
4. Satisfação do consumidor	658.89

Catalogação na fonte elaborada pela bibliotecária
Ana Guimarães Pereira ó CRB 10/1460

AGRADECIMENTOS

Primeiramente a Deus, por permitir a conclusão deste trabalho.

A minha orientadora Prof^a. Dr^a. Ana Cristina Fachinelli, meus sinceros agradecimentos, não apenas pela orientação segura demonstrada, mas também pelo auxílio e confiança dedicados.

Aos meus Pais, pelo incentivo e imenso apoio.

A minha noiva Leila pela inesgotável paciência, compreensão, amor e carinho dedicados.

A todos amigos que contribuíram de alguma forma para a realização deste trabalho.

RESUMO

Em um ambiente de desenvolvimento de *software* a etapa de extração de requisitos é um processo complexo que envolve uma importante interação entre cliente e analista. Essa etapa é crucial para que um projeto se inicie de forma correta, desencadeando um efeito de assertividade e qualidade nos produtos desenvolvidos. A interação cliente analista pressupõe uma intensa troca de conhecimentos de natureza sobretudo tácita cuja interpretação afeta diretamente o resultado final do processo. Assim sendo, o objetivo do presente estudo é analisar a dimensão tácita do conhecimento no processo de definição de requisitos e sua relação com a qualidade em uma Fábrica de *Software* da Serra Gaúcha. A pesquisa é um estudo de caso único e utiliza métodos de *Sensemaking* para o resgate do conhecimento tácito. Para organização do conteúdo obtido nas entrevistas foi utilizado o *software* Atlas/Ti e para a análise foi utilizado o método dos mapas de associação de ideias. Os resultados indicam que o conhecimento tácito influencia a interpretação resultante da interação entre cliente e analista e por isso afeta o refinamento do requisito. Indicam também que o conhecimento tácito quando socializado, pode funcionar como um dispositivo de ajuste, promovendo a evolução do processo de definição de requisitos para que o produto final fique adequado à realidade.

Palavras-chave: Conhecimento Tácito. Cognição. Requisitos. Socialização. *Software*.

ABSTRACT

In a development environment *software* requirements phase extraction is a complex process that involves a significant interaction between client and analyst. This step is crucial for a project to start correctly, triggering an effect of assertiveness and quality in the products developed. The interaction customer analyst assumes an intense exchange of knowledge of nature, especially tacit interpretation of which directly affects the outcome of the process. Therefore, the objective of this study is to analyze the tacit dimension of knowledge in the process of requirements definition and its relation to quality in a *Software* Factory Serra Gaúcha. The research is a single case study and uses methods of *Sensemaking* to the rescue of tacit knowledge. To organize the content obtained in the interviews was the *software* used Atlas / Ti and the analysis method was used maps of association of ideas. The results indicate that tacit knowledge influences the interpretation resulting from the interaction between client and analyst and therefore affects the refinement of the requirement. They also show that tacit knowledge when socialized, can function as an adjustment, promoting the evolution of the process of defining requirements for the final product be suitable for reality.

Keywords: Tacit Knowledge. Cognition. Requirements. Socialization. *Software*.

LISTA DE ILUSTRAÇÕES

FIGURA 1 - Indicadores de mercado e evolução 2004-2010	16
FIGURA 2 - Componentes do <i>software</i>	23
FIGURA 3 - Características de qualidade de <i>software</i>	27
FIGURA 4 - Engenharia de <i>software</i> em camadas	31
FIGURA 5 - O processo de requisitos.....	35
FIGURA 6 - Processo de engenharia de requisitos	37
FIGURA 7 - Quatro modos de conversão do conhecimento.....	42
FIGURA 8 - Fluxo básico dos processos de gestão do conhecimento com processos de <i>Sensemaking</i>	51
FIGURA 9 - Esquema conceitual.....	53
FIGURA 10 - Histórico Totvs.....	55
FIGURA 11 - Conjunto metodológico.....	62
FIGURA 12 - Unidade hermenêutica - entrevistados	64
FIGURA 13 - Categorias e suas sub-categorias.....	65
FIGURA 14 - Exemplo de codificação	65
FIGURA 15 - Exemplo comentários.....	66
FIGURA 16 - Ferramenta de consulta 1	67
FIGURA 17 - Ferramenta de consulta 2	67
FIGURA 18 - Família colisão	68
FIGURA 19 - Família criação.....	69
FIGURA 20 - Socialização do conhecimento	77
FIGURA 21 - Características da qualidade de <i>software</i>	78
FIGURA 22 - Guia para processos de definição de requisitos	81

LISTA DE QUADROS

QUADRO 1 - Etapas e questionamentos.....	58
QUADRO 2 - Características dos profissionais entrevistados.....	63

LISTA DE SIGLAS

ABES	Associação Brasileira das Empresas de <i>Software</i>
CAQDAS	<i>Computer-Assisted Qualitative Data Analysis Software</i>
ER	Engenharia de Requisitos
ERP	<i>Enterprise Resource Planning</i>
FO	Ficha de Ocorrência
IDC	<i>International Data Corporation</i>
IEC	<i>International Engineering Consortium</i>
ISO	<i>International Organization for Standardization</i>
KARE	<i>Knowledge acquisition and sharing for requirement engineering</i>
MQI	<i>Message Questioning Interview</i>
<i>MQI</i>	<i>Message Questioning Interview</i>
NBR	Norma Brasileira
PFNETs	<i>Pathfinder Networks</i>
QA	<i>Quality Assurance</i>
TI	Tecnologia da Informação

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 PROBLEMATIZAÇÃO.....	14
1.2 OBJETIVOS.....	19
1.2.1 Objetivo geral	19
1.2.2 Objetivos específicos	19
1.3 JUSTIFICATIVA.....	19
1.4 DELIMITAÇÃO	21
2 FUNDAMENTAÇÃO TEÓRICA	23
2.1 FÁBRICA DE <i>SOFTWARE</i>	23
2.2 QUALIDADE DE <i>SOFTWARE</i>	26
2.3 ENGENHARIA DE <i>SOFTWARE</i>	30
2.3.1 Processo de desenvolvimento de <i>software</i>	31
2.3.2 Métricas e estimativas.....	32
2.3.3 Requisitos de <i>software</i>	33
2.4 GESTÃO DO CONHECIMENTO	39
2.4.1 Criação de conhecimento organizacional.....	40
2.4.2 O Conhecimento e a gerência de requisitos	45
2.4.3 <i>Sensemaking</i> e resgate da dimensão tácita do conhecimento.....	48
3 METODOLOGIA DA PESQUISA	54
3.1 DELINEAMENTO DA PESQUISA.....	54
3.1.1 Campo de estudo	54
3.2 COLETA DE DADOS	56
3.3 TÉCNICAS E CATEGORIAS DE ANÁLISE	59
4 RESULTADOS	63
4.1 CARACTERÍSTICAS DOS ENTREVISTADOS.....	63
4.2 COLETA E ORGANIZAÇÃO DOS DADOS.....	64
5 ANÁLISE E DISCUSSÃO	71
6 CONSIDERAÇÕES FINAIS	80
REFERÊNCIAS	83

ANEXO - TERMO DE CONSENTIMENTO	92
APÊNDICE A – ROTEIRO DE ENTREVISTAS.....	93
APÊNDICE B – MAPAS DE ASSOCIAÇÃO DE IDEIAS.....	95
APÊNDICE C – INSTRUMENTO DE PESQUISA.....	131

1 INTRODUÇÃO

Software se tornou um elemento indispensável para organizações dos mais diversos ramos de atuação. É também considerado um fator estratégico, pois possibilita uma série de diferenciais em termos de serviços e de produtos. Empresas e organizações voltadas para o desenvolvimento de *Software* investem cada vez mais em tecnologia e processos para obter um produto competitivo e de qualidade. Por esse motivo estas empresas estão adotando o conceito de Fábrica de *Software* que leva em conta os atributos de uma fábrica industrial no processo de desenvolvimento de *Software*.

O termo Fábrica de *Software* foi criado nos anos 1960 e 1970, nos Estados Unidos e no Japão. Desde o início a proposta central foi a de desenvolver *Softwares* utilizando preceitos de engenharia associados à manufatura (CUSUMANO, 1991).

O objetivo de uma Fábrica de *Software* é a geração de produtos requeridos pelos usuários ou clientes, com o mínimo de defeitos e com um preço competitivo, que forneça a margem necessária para os investimentos em manutenção e melhoria da fábrica. Estes aspectos são os principais atributos de uma fábrica industrial (FERNANDES e TEIXEIRA, 2004).

Outro fator que identifica uma Fábrica de *Software* é a existência de conjuntos de processos e métodos bem desenvolvidos, que proporcionam maior produtividade. Desta forma os processos operacionais acabam ganhando mais autonomia em relação ao fator humano, evitando falhas de qualidade. Estes métodos são identificados no processo de desenvolvimento de *software* e podem estar presentes tanto em um produto novo como na alteração ou melhoria de um já existente.

Além do conceito de Fábrica de *Software*, com o aumento da escala e da complexidade dos sistemas de informação, a abordagem do ciclo de vida foi levada em conta no desenvolvimento de *software*. Essa abordagem gerou maior ênfase no desenvolvimento de sistemas de informações gerenciais, ou seja, sistemas mais simples e que funcionavam de forma isolada foram descontinuados dando lugar a sistemas integrados aptos a atender toda a organização com capacidade de gerar informações de níveis gerenciais e estratégicas.

Muitos são os modelos de ciclo de vida de um *software*: cascata, espiral, prototipação evolutiva, prototipação incremental, desenvolvimento incremental, etc. Embora uns com mais ou menos capacidade, com vantagens e desvantagens, todos tem como característica descrever as etapas do processo de desenvolvimento de sistemas e as atividades a serem realizadas em cada etapa. Entre os modelos existem divergências no enfoque e na estratégia, pois cada

modelo representa um processo a partir de uma perspectiva particular (SOMMERVILLE, 1995). Porém, segundo Sanches (2001), para se alcançar um diferencial competitivo muitos países inclusive o Brasil utilizam a norma internacional NBR ISO/IEC 12207 –Tecnologia da Informação - Processos de Ciclo de Vida de *Software* (ISO, 1995a) como referência. Ela estabelece uma estrutura comum para os processos de ciclo de vida e desenvolvimento, proporcionando às empresas um melhor entendimento dos componentes existentes na aquisição e fornecimento de *software*.

Segundo Pfleeger (2004), é possível dizer que processo de desenvolvimento de *software* também pode ser chamado de ciclo de vida do *software*, pois ele descreve a “vida” do produto de *software* desde a concepção até a implementação, entrega, utilização e manutenção. Ele determina as fases e o relacionamento entre elas com processos fundamentais que atendem a operação e manutenção dos produtos de *software*. Tanto as fases como o relacionamento representam diversas medições, detalhamento de como os processos devem ser executados, qual método deverá ser utilizado, descrição das atividades e seus relacionamentos, artefatos consumidos, ferramentas utilizadas entre outros.

Para Rocha, Maldonado e Weber (2001) independente de paradigma, área de aplicação, do tamanho do projeto e da complexidade, o processo de desenvolvimento de *software* possui três fases genéricas: a definição, o desenvolvimento e a manutenção. Neste trabalho será abordada a fase de definição que objetiva a montagem do escopo através dos requisitos formando a base de todo o processo de desenvolvimento de *software*. Essa fase é tão importante que se for mal executada pode comprometer todo o projeto, limitando assim o que deverá ser feito e o que se espera como resultado.

Na fase de definição é que ocorre a explicitação e estruturação dos requisitos. “Requisito é uma característica ou a descrição de algo que o sistema será capaz de realizar” (PFLEEGER, 2004 p.111). Estes requisitos devem passar por algumas fases onde são reescritos, verificados e validados para que se garanta que a descrição corresponde ao que o cliente espera.

Os requisitos especificam o conjunto de funcionalidades que um *software* deve prover para satisfazer as necessidades de todos os envolvidos no projeto, as características de qualidade e as restrições a que devem estar sujeitas às funcionalidades (SAMPAIO et al., 2005). Assim a engenharia de requisitos pode ser definida como a ciência e disciplina preocupada com a análise e documentação dos requisitos citados anteriormente, além de fornecer mecanismos apropriados para facilitar as atividades de análise, documentação e validação.

Tanto na criação quanto na implementação de melhorias em *software*, uma das tarefas mais importantes a ser executada é a extração dos requisitos. Em diversos momentos as necessidades dos clientes não são claras. Por isso é necessário habilidade e experiência de analistas e gerentes de projetos, para tratar e identificar a incompletude, ambiguidade ou em alguns casos até a contradição dos requisitos, possibilitando dessa forma uma compreensão completa do problema. Nesse sentido, Leite (2001) diz que o processo de definição dos requisitos deve lidar com diferentes pontos de vista e usar uma combinação de métodos, ferramentas e equipes.

Se a análise for feita de forma incorreta o resultado final será desastroso podendo levar ao desenvolvimento de um produto que não atenda aos objetivos para o qual foi planejado, gerando a necessidade de um novo ciclo de especificação, projeto, e codificação. A análise de requisitos é tão importante que em casos mais graves o código criado pode ser até desperdiçado, afetando diretamente os prazos envolvidos, trazendo prejuízos financeiros elevados, além de comprometer a credibilidade da empresa.

Como se trata de um processo que ocorre a partir da relação cliente/analista é importante considerar os elementos relativos à interpretação da dinâmica informacional presente nesta interação. Muito mais do que uma simples troca de informações, trata-se de um ato cognitivo que diz respeito ao processo de conhecer, intimamente ligado à atenção, memória, pensamento, raciocínio, entre outras ações do ser humano. Assim, parece evidente que o processo cognitivo pode influenciar de forma direta na compreensão e interpretação feita pelos analistas.

Nesta interação cognitiva cliente/analista tão essencial à análise de requisitos, a troca de conhecimentos e sua interpretação é fundamental para que as necessidades do cliente se traduzam em produtos adequados à sua demanda. O trabalho do analista nesse caso refere-se às dimensões do conhecimento do cliente que podem não estar ainda declarados, ou seja, se encontram numa dimensão tácita. Acessar essa dimensão pode representar melhores níveis de adequação do produto final.

1.1 PROBLEMATIZAÇÃO

Com os constantes avanços da tecnologia nos dias atuais cada vez mais percebe-se o valor do *software*. Ele é sem dúvida um dos componentes de extrema importância em qualquer atividade de negócio, uma vez que o tratamento das informações, de forma precisa e correta, trará um diferencial competitivo para a empresa em relação aos seus concorrentes.

Durante as primeiras décadas da era do computador, aproximadamente de 1945 até meados da década de 70, um grande desafio que se tinha era desenvolver um *hardware* de baixo custo e alto desempenho. Hoje, o desafio passa ser a busca constante em aperfeiçoar a qualidade, aumentar a disponibilidade e capacidade, além de reduzir os custos das soluções baseadas em *software*.

Brooks (1987) descreve que *software* é um conjunto de construções conceituais, complexas e não lineares, sujeitas a modificações, e invisível, destacando desta maneira a dificuldade e complexidade envolvida na atividade de construção de um *software*. Muitas metodologias para a criação de *software* apontam para a aproximação cada vez maior com a abordagem da criação de produtos na gestão da manufatura, ou seja, pode-se comparar a produção de *software* com a produção industrial, onde os equipamentos necessitam de atenção especial, pois incidem diretamente nas medidas de produtividade, requerendo estratégias que as fazem produzir mais com o mínimo possível de interrupções (OLIVEIRA e NETO, 2003). Ainda nesse sentido Oliveira e Neto (2003) entendem que o estágio atual da Tecnologia da Informação (TI) determina que *softwares* sejam constituídos por módulos ou peças menores que são acopladas para a montagem do produto final.

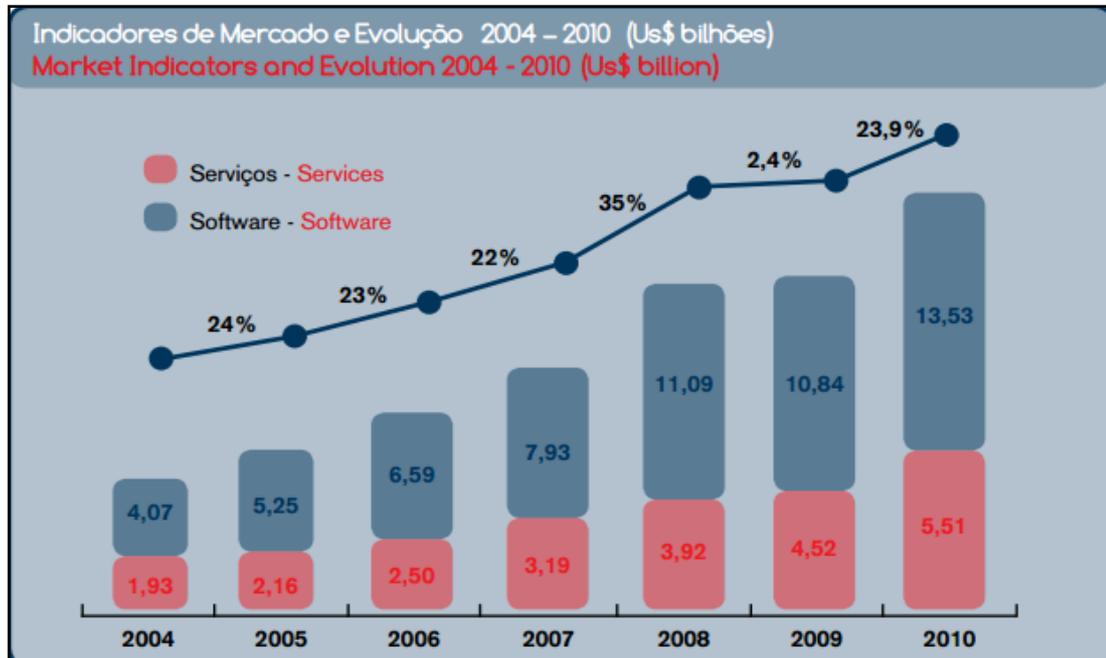
A criação destes módulos permite que, quando planejados de forma correta, sejam desenvolvidos individualmente passando por um processo de unificação para formar um único produto. Conforme Brito (2000) o conceito de Fábrica de *Software* está basicamente centrado na ideia de proporcionar soluções que atendam às necessidades específicas de cada cliente necessitando para isso da formalização das atividades com etapas e tarefas bem definidas para cada tipo de profissional, indo da produtividade da linha de produção à qualidade.

Segundo Fernandes (2004), o objetivo de uma Fábrica de *Software* deve ser a geração de produtos requeridos pelos usuários ou clientes, com o mínimo de defeitos possível e a um preço/custo competitivo e compatível, que forneça a margem necessária para os investimentos em manutenção e melhoria da fábrica. Dessa forma pode-se perceber a importância e necessidade de se possuir um produto com valor agregado, onde qualidade é algo indispensável.

O Brasil, segundo pesquisa feita no ano de 2010 pela Associação Brasileira das Empresas de *Software* (ABES), ocupa a 11ª colocação no cenário mundial de produção de *software*, mantendo-se estável neste cenário, onde movimentou cerca de US\$ 19,4 bilhões em 2010 saldo 23,9% acima do alcançado no ano anterior. Desse total, US\$ 5,51 bilhões referem-se a *softwares* o equivalente a 2,2% do mercado mundial, e US\$ 13,53 bilhões referem-se a serviços relacionados. No mesmo período o mercado mundial de *softwares* e serviços

registrou pequeno avanço de 0,5% e movimentou US\$ 884,5 bilhões. Destaque para os resultados alcançados com a exportação que em 2010 registrou avanço de 15,7% comparando com o ano de 2009, montante equivalente a US\$ 1,74 bilhões.

FIGURA 1 - Indicadores de mercado e evolução 2004-2010



Fonte: Associação Brasileira de *Software* (2012).

Ainda conforme a pesquisa, no ano de 2010 foram identificadas mais de 8.520 empresas pertencentes ao setor brasileiro de *software* que se dedicam ao desenvolvimento, distribuição e comercialização de *software*. Além disso, 94% deste grupo de organizações são classificadas como micro e pequenas empresas, as quais formam uma cadeia de valor com potencial de grande expansão pela sua capacidade de distribuição no mercado interno.

Nos dois últimos anos, quase 50% da demanda registrada foi proveniente dos mercados financeiro e industrial seguidos pelos mercados da agroindústria, governo, comércio e serviços. Este estudo foi realizado com base em dados coletados pelo *International Data Corporation* (IDC), junto a seus 50 escritórios no mundo, além de entrevistas com fornecedores e exportadores de *software*, inclusive empresas multinacionais instaladas no Brasil. Os dados aqui analisados são relativos ao Q4 - 2010 do "Black Book" do IDC.

Segundo a ABES, a expansão da computação em nuvem, cuja demanda deve triplicar nos próximos cinco anos; a procura por aplicativos de análise e de inteligência para o negócio, a demanda por informações e conhecimento sobre o conceito de Cloud são algumas tendências para 2012. Especificamente em relação a *software* e serviços, a indústria nacional deverá ter um crescimento de quase o dobro da média mundial, impulsionado pela economia,

planos de crescimento das empresas, explosão dos *smartphones* e apetite por eletrônicos na Classe C.

Com um viés de alta para os próximos anos as empresas que desenvolvem *software* devem estar preparadas para esse mercado de modo que fatores como qualidade, gerenciamento de processos, utilização de modelos confiáveis, segurança, entre outros devem estar presentes na construção de seus produtos. Por isso Fernandes (2004) afirma que uma Fábrica de *Software* deve possuir processos definidos e estruturados, que possam ser controlados e aperfeiçoados de forma contínua, voltados para o atendimento a múltiplas demandas de natureza e escopo distintas. Ainda segundo o autor, devem ser consideradas abordagens de outras engenharias como a industrial referindo-se à geração de produtos de *software*, conforme os requerimentos documentados pelos usuários e clientes, da forma mais produtiva e econômica possível.

Dias (2004) diz que associar Fábrica de *Software* à gestão da manufatura é possível, assim como também é possível dizer que se trata de um processo inovador, que permite a construção de *software* totalmente alinhado com a necessidade para qual será construído. A norma NBR ISO/IEC 9126, sob o título geral Engenharia de *software* - Qualidade do Produto, define qualidade de *software* como “A totalidade de características de um produto de *software* que lhe confere a capacidade de satisfazer necessidades explícitas e implícitas”.

Baseado no exposto acima, Rocha, Maldonado e Weber (2001) dizem que a questão da qualidade de *software* é tratada no processo de definição dos requisitos através de uma declaração clara das especificações que o sistema deve atender. Isso corresponde a um problema corriqueiro no processo de desenvolvimento de *software*, onde o não tratamento dos aspectos relacionados às características do produto no início do processo, ou seja, a extração dos requisitos acaba condicionando a qualidade de *software*.

Bugs ou falhas em sistemas podem causar prejuízos que vão desde a falta de uma informação em um relatório até a perda de milhões de reais a uma instituição financeira ou parar por horas uma companhia telefônica, impedindo a realização de telefonemas. Com o aumento exponencial das tecnologias e utilização maciça de *softwares* os problemas de qualidade tendem a aumentar, o que evidencia uma necessidade atual.

Segundo Davis (1993) um estudo dos custos dos erros de requisitos ocorridos em várias fases do ciclo de vida de projetos de desenvolvimento de *software* realizado em meados da década de 70 pelas empresas GTE[®], TRW[®] e IBM[®] dão conta que, se uma unidade de custo 1 é associada ao esforço requerido para detectar e reparar um erro durante a fase de codificação, o custo para detectar e reparar um erro durante a fase de requisitos está entre

cinco a dez vezes menor. Por outro lado, encontrar um erro na fase de manutenção tem um custo aproximadamente 200 vezes maior em relação à descoberta do mesmo erro na fase de análise de requisitos.

Outro dado relevante indica que em torno de 40% a 60% de todos os problemas encontrados em um projeto são causados por falhas no processo de requisitos seja pela ausência ou pela não utilização de um processo de definição de requisitos adequado (LEFFINGWELL, 1997).

Assim, fica claro que o processo de engenharia de requisitos é essencial para um projeto de desenvolvimento de *software*, pois além de garantir o desenvolvimento e aprovação de artefatos de requisitos, envolve informações de diversas origens e propósitos, permitindo a construção de um produto adequado.

Uma grande quantidade de conhecimento tácito e explícito é produzida ao longo do processo de desenvolvimento de *software* e este conhecimento precisa ser depositado em repositórios que facilitem a recuperação e agreguem valor ao processo (PARREIRAS e BAX, 2003). Seguindo nessa linha de raciocínio pode-se dizer que as diversas formas de conhecimento produzidas durante o envolvimento e a constante comunicação entre analistas e usuários acabam sofrendo influências de fatores humanos, sociais, políticos e organizacionais em todo o projeto de desenvolvimento de *software* podendo ser a causa de sucesso ou fracasso do mesmo.

As experiências cognitivas que pela própria natureza são inerentes aos sujeitos envolvidos no processo, normalmente não fazem parte de nenhum tipo de documentação em um projeto de desenvolvimento de *software* e mais dificilmente são transferidas para outros membros da equipe. Entretanto, para Bowen et al (1994), é possível compartilhar estas experiências através de redes informais de comunicação sem que estas informações estejam explícitas dentro da organização.

Para Desouza (2003), a engenharia de *software* que também abrange a engenharia de requisitos, é um domínio altamente orientado ao conhecimento, onde os fatores chave estão relacionados à experiência dos sujeitos envolvidos nas fases de desenvolvimento de *software*. Resgatar o conhecimento tácito e poder convertê-lo em algo que a organização possa utilizar é, na concepção de Nonaka e Takeuchi (1997), o processo de criação do conhecimento.

Para Love et al (2003) se gerenciado corretamente, o conhecimento pode ser usado para reduzir o tempo de projeto e aumentar sua qualidade além da satisfação dos *stakeholders*. Considerar esta gestão do conhecimento, seja ele explícito ou tácito, é um pré-requisito necessário para o sucesso de projetos nos ambientes dinâmicos de hoje.

Sendo assim, a questão que se coloca e que orienta o presente projeto de pesquisa refere-se à dimensão tácita do conhecimento presente na relação cliente/analista, ou seja, como a dimensão tácita do conhecimento pode contribuir para o desenvolvimento do processo de análise de requisitos, para que as necessidades dos clientes se traduzam em produtos adequados à sua demanda?

1.2 OBJETIVOS

1.2.1 Objetivo geral

Analisar a dimensão tácita do conhecimento no processo de definição de requisitos e sua relação com a qualidade de *software*.

1.2.2 Objetivos específicos

- a) Estudar os processos de interação cliente/analista e sua relação com a identificação do conhecimento tácito;
- b) identificar a interação cliente/analista e sua influência na transformação dos requisitos em produtos dentro do processo de desenvolvimento de *software*;
- c) investigar se a socialização do conhecimento produzida em cada projeto proporciona aprendizagem ao resto da equipe;
- d) verificar o uso do *Sensemaking* para o resgate e socialização do conhecimento tácito;
- e) analisar a influência dos requisitos na satisfação dos clientes.

1.3 JUSTIFICATIVA

Em meados da década de 80, início da década de 90, período em que permanecia a reserva de mercado no Brasil no setor de informática, a produção de *software* não recebia tanta importância como atualmente, pois a maioria dos incentivos era destinada as empresas produtoras de *hardware*. Somente a partir do fim da reserva de mercado, em 1992, é que as empresas produtoras de *software* começaram a receber algum benefício com as políticas de incentivos criadas pelo governo brasileiro (SOFTEX, 2002).

De acordo com Castells (2006), a indústria de TI (tecnologia da informação) é a indústria que abarca tudo o que é produzido em *software*, *hardware* e serviços. A TI sem dúvida tem um grande peso na economia de qualquer país, visto que, afeta profundamente o desempenho de todos os setores econômicos, sejam eles públicos ou privados (TAURION, 2004).

Ainda segundo Taurion (2004), na indústria de TI, muitas empresas que se destacavam como grandes produtoras de *hardware* passaram a investir em desenvolvimento de *software* e serviços especializados de consultorias, tornando o setor cada vez mais competitivo. Por isso nos últimos anos é possível perceber um aumento no interesse de empresas desenvolvedoras em se tornarem fábricas de *software*, visando o crescimento desta modalidade no Brasil. Um dos fatores são os investimentos feitos no país por parte de empresas dos EUA e da Europa, buscando uma maior solidez (CÉSAR, 2010).

Roselino (2006) afirma que a indústria brasileira de *software* apresenta aspectos promissores, mesmo com resultados externos pouco expressivos até o momento. Esse também é o pensamento de Silva Filho (2003) quando declara que o potencial do Brasil no exterior é pouco conhecido, mas que a demanda para a internacionalização da produção de *software* cria uma boa perspectiva para o país apesar de que para isso aconteça alguns desafios devam ser superados, principalmente no que diz respeito à qualidade de produtos.

Essa necessidade evidencia ainda mais a importância dos requisitos de *software*, pois eles são o início dos projetos e dão a direção destes. Em sua obra, Pressman (2006) deixa clara a dificuldade existente neste processo.

É seu pior pesadelo. Um cliente entra no seu escritório, senta-se, olha você direto nos olhos e diz: “Eu sei que você pensa que entende o que eu disse, mas o que você não entende é que, o que eu disse, não é o que eu queria dizer”. Invariavelmente, isso acontece no final de um projeto, depois que os compromissos de prazo de entrega forma feitos, que as reputações estão envolvidas e que dinheiro sério está em jogo (PRESSMAN, 2006, p. 116).

Por isso, a influência do conhecimento tácito na fase de definição dos requisitos se torna tão importante. Nessa etapa o resgate de tal conhecimento pode ser utilizado como suporte para definições tanto de analistas como de clientes. Além disso, a interação entre as partes pode auxiliar no nível de refino e detalhamento, influenciando diretamente na assertividade do requisito e conseqüentemente na qualidade do produto final.

O campo de estudo escolhido para o desenvolvimento do presente estudo é uma empresa de informática que utiliza o conceito com todas as características de uma Fábrica de *Software*. Trata-se da Totvs que é uma empresa de *software*, inovação, relacionamento e

suporte à gestão, líder absoluta no Brasil, com 49,1¹% de *share* de mercado, e também na América Latina, com 31,2%. É a maior empresa de *softwares* aplicativos sediada em países emergentes e a 6ª maior do mundo no setor (Totvs 2012). O estudo será realizado na Franquia de Desenvolvimento de Soluções para a Governança de Saúde (Totvs) localizada na cidade de Caxias do Sul a qual possui atualmente 139 funcionários divididos entre matriz e filial e uma carteira composta por aproximadamente 110 clientes responsáveis por um faturamento anual de 12 milhões.

Atualmente na literatura encontra-se trabalhos que relacionam a gestão do conhecimento com o processo de definição de requisitos de uma forma bem abrangente dando maior enfoque na conversão do conhecimento tácito e pessoal em explícito (KUDIKYALA e VAUGHN, 2005, SHAW e GAINES, 1992). Essa tratativa além de ser mais simplista também não aborda a utilização de técnicas como, por exemplo, o *Sensemaking* que também pode ser reconhecido em outros trabalhos de diversos autores (QIU, CHUI e HELANDER, 2007, KJÆRGAARD, KAUTZ e NIELSEN, 2007). No caso do *Sensemaking* o foco dado no presente trabalho evidencia a utilização de processos e perspectivas cognitivas para auxiliar nas tomadas de decisão, incluindo a etapa da definição dos requisitos.

Com isso, foi identificado um espaço vazio na teoria, justamente onde se quer enquadrar o presente estudo. Essa lacuna está na utilização da dimensão tácita do conhecimento juntamente com o *Sensemaking* de modo a aperfeiçoar a qualidade do *software* a partir do processo de levantamento de requisitos.

1.4 DELIMITAÇÃO

Esta pesquisa não teve a ambição de estudar todas as etapas do desenvolvimento de *software* nem de analisar multicasos de empresas situadas na serra gaúcha, mas sim uma em especial, uma franquia Totvs.

O escopo deste trabalho está em identificar o conhecimento tácito nos processos de interação entre cliente e analista na definição de requisitos. Para isso, foi estudada especificamente a utilização do *Sensemaking* como forma de acessar a dimensão tácita do conhecimento das pessoas envolvidas em uma das principais fases do desenvolvimento de *software*: a fase de levantamento de requisitos e sua relação com a qualidade final do produto.

¹Fonte: Gartner – Market Share ERP Software, Worldwide, 2009.

A pesquisa não teve por finalidade estudar os modelos ou conceitos do *Sensemaking* em si e tampouco visava identificar os processos de gestão ou criação do conhecimento nas Fábricas de *Software*. Tanto a gestão do conhecimento quanto a abordagem da criação do conhecimento são neste estudo conceitos estruturantes do objeto que é especificamente o conhecimento tácito e sua relação com a qualidade de *software*.

Também no que diz respeito ao conceito de fábrica de *software*, a delimitação do estudo se situa na engenharia de *software* como uma derivação desta abordagem e como é tratada a questão da qualidade neste contexto. Ou seja, os parâmetros definidos nesta literatura indicam os critérios utilizados na pesquisa para estabelecer a relação com o objeto de estudo.

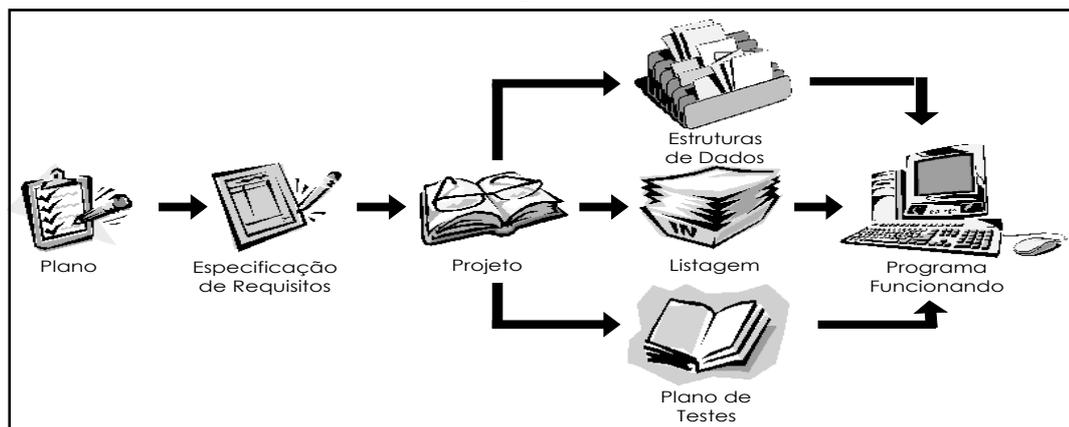
2 FUNDAMENTAÇÃO TEÓRICA

2.1 FÁBRICA DE SOFTWARE

Software é definido por Pressman (2006) como um produto que é executado em computadores. Estes produtos também são conhecidos como programas que são desenvolvidos e mantidos por profissionais da área e que estão presentes em praticamente todas nossas atividades do dia-a-dia. Também pode-se dizer que ele é a parte abstrata de um sistema computacional que é executado através de uma linguagem de programação. Toda e qualquer integração entre usuário e sistema é feita através de uma interface, que é composta por *hardware* e *software*. A figura 2 mostra de forma simplificada os componentes de um software.

Sommerville (1995) afirma que, ao contrário do que muitas pessoas pensam, *software* não está relacionado apenas aos programas de computador. *Software* não é apenas o programa, mas sim toda a documentação associada a ele que descreve sua estrutura e a documentação para o usuário. *Software* também compreende todo o conjunto de programas, procedimentos, dados e documentação associados a um sistema de computador e não apenas ao computador em si (PFLEEGER, 2004, p. 1)

FIGURA 2 - Componentes do *software*



Fonte: Pfleeger (2004).

Conforme Fernandes e Teixeira (2004), antes da década de 80 não havia processo disciplinado para desenvolvimento de *software*, pelos menos não no Brasil. Esse processo que antes era artesanal passou a ser mais industrializado com o surgimento do conceito de Fábrica de *Software*, mas que só pode ser aplicado em escala comercial a partir de 1993 no mercado de São Paulo.

Os conceitos e as propostas de fábricas de *software* encontrados na literatura como Greenfield e Short (2003), Fernandes e Teixeira (2004), Cusumano (2005), Woodward (2005), Demir (2006), abrangem práticas da manufatura industrial, relacionam às linhas de produtos de *software* ao uso de ferramentas computacionais específicas ao uso de processos, modelos de qualidade e reuso.

O conceito de Fábrica de *Software* ilustra a mudança do paradigma da produção de *software* centrada no trabalho intensivo para um modelo mais centrado no capital, onde investimentos são praticáveis e os riscos são de certa forma calculados. Apesar de existirem vínculos entre o desenvolvimento de *software* e a manufatura industrial, existem algumas diferenças fundamentais em relação à produção da manufatura industrial, tais como, a não produção em série e o vínculo da realização das atividades de desenvolvimento de *software* à criatividade e ao conhecimento prático e teórico das pessoas.

O modelo das fábricas clássicas, onde pessoas são vistas como máquinas na execução de atividades preestabelecidas, não é aceitável para uma Fábrica de *Software*. Nesse contexto, fazendo uma analogia com as fábricas clássicas, o que é plausível de ser utilizado são as práticas da manufatura como, por exemplo, ter processos definidos possibilitando flexibilidade aos projetos, a utilização de modelos de qualidade em processos e produtos e a reutilização no desenvolvimento de *software*. Em suma o objetivo principal é adequar princípios da manufatura industrial à produção de bens intensivos em conhecimento e dependentes de customização, como ocorre com o *software*.

Apesar dos vínculos com a manufatura industrial, ressalta-se que a essência de uma Fábrica de *Software* deve ser o objetivo técnico e de gerenciamento, com ênfase na qualidade. O foco técnico se refere a infra-estrutura de suporte e comunicação, a qual possibilita interação entre as pessoas. Já o gerenciamento pretende extrapolar modelos fixos de projetos de *software* através da especialização de modelos de processos, do reuso de conhecimento e de artefatos e da aplicação de qualidade e controle aos processos e aos produtos.

Castor (2004) reitera que a característica principal do modelo de Fábrica de *Software* é objetivar a criação de um ambiente de desenvolvimento produtivo de produtos de *software* com qualidade e baixo custo, para isso, são utilizadas técnicas da engenharia de produção em série, buscando a solução de problemas encontrados neste ambiente como a baixa produtividade e alto custo na produção de sistemas.

Siy et al. (2001) propõe uma definição parecida para Fábrica de *Software*, afirmando ser uma instituição que provê serviços e produtos em um curto prazo de tempo, com custos

baixos e qualidade proeminente, uma vez que o desenvolvimento se utiliza de processos bem definidos e tecnologia avançada.

Em síntese, as fábricas de *software* buscam a melhoria dos seus processos com o objetivo de otimizar a utilização dos recursos e diminuir os seus custos operacionais. Ainda segundo Fernandes e Teixeira (2004), uma Fábrica de *Software* deve possuir independente de seu escopo de fornecimento, alguns atributos básicos.

Segundo o autor, estabelecer processo definido e padrão para o desenvolvimento de produtos de *software*, além de monitorar de forma eficiente o andamento da execução de cada demanda permite construir produtos de *software* com maior produtividade e redução dos custos de operação (FERNANDES e TEIXEIRA, 2004).

Para Cusumano (2005) uma Fábrica de *Software* deve ter o foco ampliado no gerenciamento de projetos, no favorecimento do reuso e em atividades de garantia da qualidade.

Uma definição mais ampla para Fábrica de *Software*, atentando para o ambiente onde os processos são realizados, é dado por Fabri et al. (2004). Esses autores definem Fábrica de *Software* como uma organização estruturada, direcionada para a produção de *software* fundamentada na engenharia além de ser caracterizada pela organização do trabalho e na sua capacidade de modularização dos componentes e de escalabilidade produtiva.

Na concepção de Fábrica de *Software*, existe o reuso de artefatos provenientes das tarefas dos processos de *software*, além do reuso do aprendizado derivado da execução de atividades como modelagem e implementação de sistema de *software* e do uso de ferramentas computacionais.

Para ser reusada, uma informação precisa estar armazenada em algum meio físico ou eletrônico e a semântica e seu contexto passam a ser muito importantes, pois os dois devem estar bem definidos e compreendidos. Uma ontologia semântica que possibilita definir o conteúdo semântico da informação, permitindo armazená-la, recuperá-la e usá-la com mais eficiência pode ser uma solução. Baseado nos conceitos de Fábrica de *Software* citados, um ambiente de desenvolvimento de *software* é considerado tudo o que é necessário para o desenvolvimento dos projetos de *software*, como, processos, pessoas, recursos, artefatos de *software*. Dessa forma o ambiente de desenvolvimento de *software* passa a ter um significado mais abrangente do que o encontrado na literatura, que está atrelado a ferramentas computacionais para a automatização do desenvolvimento de *software*.

2.2 QUALIDADE DE *SOFTWARE*

Um *software* de qualidade tem boa usabilidade, não possui erros ou falhas, é de fácil manutenção e acima de tudo assegura a integridade dos dados. Entretanto, nem sempre é isso que acontece, e são essas falhas, tratativas erradas das regras de negócio entre outros problemas que a qualidade do *software* busca corrigir.

Pressman (2006, p. 577) inicia o capítulo referente à gestão da qualidade de seu livro fazendo a seguinte referência: “A abordagem de engenharia de *software* descrita neste livro trabalha buscando uma única meta: produzir *software* de alta qualidade”. Ele diz mais, afirmando que muitos desenvolvedores de *software* acreditam que devem se preocupar com a qualidade de seu produto depois que o código tenha sido gerado. Segundo ele esse já é um erro, pois a gestão da qualidade de *software* é uma atividade guarda-chuva que é aplicada ao longo de todo o processo de desenvolvimento do *software*.

Nessa linha, Sommerville (2003) fala que o objetivo da maioria das organizações é produzir produtos e prestar serviços com os maiores níveis de qualidade possíveis. Nos dias atuais não é mais admissível entregar produtos de baixa qualidade e identificar falhas depois que os mesmos já estão sendo utilizados por seus clientes. O *software*, da mesma forma que outros produtos de manufatura tem esse mesmo tipo de problema.

O termo qualidade tem definições diferentes, segundo Plant (1991). Ele cita dois exemplos claros. O primeiro tem relação com o que o cliente pensa: Qualidade é um julgamento feito pelos clientes ou usuários de um produto ou serviço: é a medida em que os clientes ou utilizadores acreditam que o produto ou serviço ultrapassa as suas necessidades e expectativas.

Já o segundo está associado a um contexto fabril: Qualidade é conformidade com os requisitos. O desvio da especificação implica em redução na qualidade. Esta definição se aproxima do conceito Fábrica de *Software* e alinha-se com o objetivo deste trabalho.

Atualmente, qualidade é mais do que isso. Para Machado (2001) a conquista da certificação ISO 9000 que é reconhecida por todos os países e setores, não apenas pelo setor de *software* significa alcançar padrão internacional de qualidade também em processos. O processo de garantia de qualidade na norma ISO/IEC 12207 garante que os processos e produtos de *software* aderem aos requisitos especificados pelos clientes. Além disso o processo de garantia da qualidade deve estar coordenado com os processos de verificação, validação, revisão conjunta e auditoria (MACHADO, 2001).

A norma internacional ISO/IEC 9126, emitida em 2001 e que no Brasil é localizada com número NBR 13596 ou NBR ISO/IEC 9126-1:2003, sugere um modelo de escopo amplo e que determina seis grandes categorias as quais tratam das características de qualidade de *software*. Estas categorias são subdivididas em subcaracterísticas:

FIGURA 3 - Características da qualidade de *software*

CARACTERÍSTICAS	SUBCARACTERÍSTICAS	SIGNIFICADO
Funcionalidade O conjunto de funções satisfazem as necessidades explícitas e implícitas para a finalidade a que se destina o produto?	Adequação	Propõe-se a fazer o que é apropriado?
	Acurácia	Gera resultados corretos ou conforme acordados?
	Interoperabilidade	É capaz de interagir com os sistemas especificados?
	Segurança de acesso	Evita acesso não autorizado, acidental ou deliberado a programas e dados?
	Conformidade	Está de acordo com normas e convenções previstas em leis e descrições similares?
Confiabilidade O desempenho se mantém ao longo do tempo e em condições estabelecidas?	Maturidade	Com que frequência apresenta falhas?
	Tolerância a falhas	Ocorrendo falhas como ele reage?
	Recuperabilidade	É capaz de recuperar dados após uma falha?
Usabilidade É fácil utilizar o software?	Inteligibilidade	É fácil entender os conceitos utilizados?
	Apreensibilidade	É fácil aprender a usar?
	Operacionalidade	É fácil de operar e controlar a operação?
Eficiência Os recursos e os tempos utilizados são compatíveis com o nível de desempenho requerido para o produto?	Comportamento em relação ao tempo	Qual é o tempo de resposta e de processamento?
	Comportamento em relação aos recursos	Quanto recurso utiliza?
Manutenibilidade Há facilidade para correções, atualizações e alterações?	Analisabilidade	É fácil encontrar uma falha quando ocorre?
	Modificabilidade	É fácil modificar e remover defeitos?
	Estabilidade	Há grandes riscos de <i>bugs</i> quando se faz alterações?
	Testabilidade	É fácil testar quando se faz alterações?
Portabilidade É possível utilizar o produto em diversas plataformas com pequeno esforço de adaptação?	Adaptabilidade	É fácil adaptar a outros ambientes sem aplicar outras ações ou meios além dos fornecidos para esta finalidade no software considerado?
	Capacidade para ser instalado	É fácil instalar em outros ambientes?
	Capacidade para substituir	É fácil substituir por outro software?
	Conformidade	Está de acordo com padrões ou convenções de portabilidade?

Fonte: Adaptado da NBR ISO/IEC 9126-1:2003.

Porém, segundo Weber, Rocha e Nascimento (2001), qualquer que seja a abordagem utilizada para a melhoria dos processos é necessário para as empresas de *software* brasileiras:

- a) investir em métodos para prevenção de defeitos;
- b) cultivar o hábito de medir os seus processos de *software*;
- c) aprender a identificar as causas dos problemas ou defeitos;
- d) saber agir corretiva e preventivamente para eliminar esses problemas ou defeitos e, principalmente, as suas causas.

Dessa forma, a responsabilidade dos gerentes de projeto em uma organização é garantir que o nível de qualidade exigido para os produtos seja sempre atingido. O gerenciamento de qualidade inicialmente envolve definir procedimentos e padrões que devem ser utilizados durante o desenvolvimento de *software* e seguidos por toda a equipe de engenheiros (SOMMERVILLE, 2003).

Para o autor, os bons gerentes de qualidade objetivam implantar uma cultura de qualidade na sua equipe. O conceito de qualidade deve fazer parte dos processos e novas abordagens de melhorias devem ser instigadas por estes gerentes.

O gerenciamento de qualidade de *software* pode ser estruturado em três atividades principais segundo Sommerville (2003):

Garantia de qualidade

É o estabelecimento de uma estrutura de procedimentos e de padrões organizacionais, que conduzem ao *software* de alta qualidade.

As atividades de garantia de qualidade *Quality Assurance* (QA) definem uma estrutura para atingir a qualidade de *software* (SOMMERVILLE, 2005). Esse processo de QA envolve definir ou selecionar os padrões que devem ser aplicados ao processo de desenvolvimento de *software* ou ao produto de *software*. Segundo o autor, esses padrões podem ser integrados em procedimentos ou processos que são aplicados durante o desenvolvimento e ferramentas podem fornecer o apoio necessário.

Segundo Sommerville (2005), existem dois tipos de padrões que podem ser estabelecidos como parte do processo de garantia de qualidade: Padrões de produto e padrões de processo. Um ponto que merece destaque são os padrões de documentação, incluídos nos padrões de processos, visto que os documentos são a única maneira tangível de representar o *software* e seus processos. Eles devem ser de fácil leitura e compreensão. Sommerville (2005) identifica três padrões de documentação os quais são: Padrões de processo de documentação, de documento e de intercambio de documentos.

Uma documentação de qualidade proporciona uma maior organização durante o desenvolvimento de um sistema, possibilitando que futuras manutenções sejam realizadas sem grandes problemas. Permite também que no caso da ausência de usuários chave do processo ou especialistas o desenvolvimento de fases posteriores gere menos transtornos, além de contribuir para redução de erros, aumentando assim, a qualidade do processo e do produto gerado. Dessa forma, a criação da documentação é tão importante quanto a criação do *software* em si (SANCHES, 2001).

Planejamento de qualidade

É a seleção de procedimentos e de padrões adequados a partir de uma estrutura e a adaptação destes para um projeto específico de *software*.

O planejamento de qualidade deve começar em um estágio inicial do processo de *software*. Um plano de qualidade deve estabelecer as qualidades desejadas para o produto. Ele deve definir como essas qualidades devem ser avaliadas, portanto o plano define o que *software* de “alta qualidade” realmente significa (SOMMERVILLE, 2006).

Controle de qualidade

É a definição e a aprovação de processos que assegurem que os procedimentos e os padrões de qualidade do projeto sejam seguidos pela equipe de desenvolvimento de *software*.

Conforme Pressman (2006) o controle de qualidade envolve uma série de inspeções, revisões e testes usados ao longo do processo de *software* garantindo que cada produto de trabalho satisfaça os requisitos que para ele foram estabelecidos. Existem duas abordagens para o controle de qualidade que se complementam. São as revisões de qualidade que como o nome já diz, implicam em revisões por um determinado grupo de pessoas no *software* e sua documentação e as avaliações automáticas, onde uma ferramenta analisa o *software* e seus documentos com um padrão aplicado a este tipo de projeto.

Contudo o gerenciamento de qualidade deve ser separado do gerenciamento de projeto, de modo que a qualidade não seja comprometida pelas responsabilidades de gerenciamento quanto ao orçamento e aos prazos de projeto.

Uma atenção especial deve ser dada para a qualidade dos requisitos de *software*. Eles sem dúvida interferem diretamente na qualidade do mesmo, visto que, um *software* que não possua erro, mas que também não faça o que foi solicitado será enquadrado como um produto sem qualidade.

Segundo experiências de Leite (2001) não há nada mais decepcionante do que construir um *software* que não atenda as necessidades dos clientes. Recursos são gastos, sem falar na frustração dos clientes e na perda de credibilidade por parte da empresa. Para os autores, este tipo de problema e outros mais são derivados da falta de atenção para a tarefa de definição e de acompanhamento dos requisitos no processo de desenvolvimento do *software*.

Conforme já dito anteriormente, quando se fala em qualidade deve-se dar atenção tanto para processo como para os produtos. Durante muito tempo a engenharia de *software* centrou sua atenção na qualidade do produto. Atualmente uma visão mais abrangente se faz necessária.

A parte mais difícil do desenvolvimento de *software* é decidir precisamente o que será desenvolvido. Nenhuma outra parte do trabalho é tão difícil quanto estabelecer (definir) os detalhes técnicos necessários incluindo todas as interfaces para pessoas, máquinas e para outros sistemas de *software*. Nenhuma outra parte do trabalho é tão possível de ocasionar erros no sistema como essa. Nenhuma outra parte é tão difícil de ser posteriormente consertada (BROOKS, 1987, p. 53)

É possível encontrar na literatura diversos artigos que abordam a importância da qualidade em produtos de *software* (Alshayeb, 2009; Chiu, 2011 e Barney, 2012). Dentre eles Plant (1991), que dá foco na qualidade de sistemas baseados em conhecimento afirmando que a utilização limitada destes sistemas se deve a baixa qualidade. Para o autor a forma de levantamento de requisitos difere dos sistemas tradicionais, pois não é possível realizar toda a especificação antes do desenvolvimento. Ainda segundo Plant (1991) a criação de técnicas de especificação aliado a criação de modelos adequados pode trazer aumento significativo na qualidade destes *softwares*.

Por fim, para Leite (2001), a questão de qualidade é tratada no processo de definição dos requisitos através da definição clara dos critérios de qualidade que o *software* deverá atender. Essa estratégia integra de uma maneira coerente aspectos funcionais e aspectos não funcionais, possibilitando o tratamento de situações antes apenas descobertas em outras fases do processo de desenvolvimento de *software*. Dessa forma aspectos de qualidade devem ser tratados na definição dos requisitos, processo que está presente na engenharia de *software* a qual veremos a seguir.

2.3 ENGENHARIA DE SOFTWARE

A Engenharia de *Software* combina aspectos referentes a adoção de processos, métodos, técnicas, ferramentas e ambientes de suporte ao desenvolvimento de *software*.

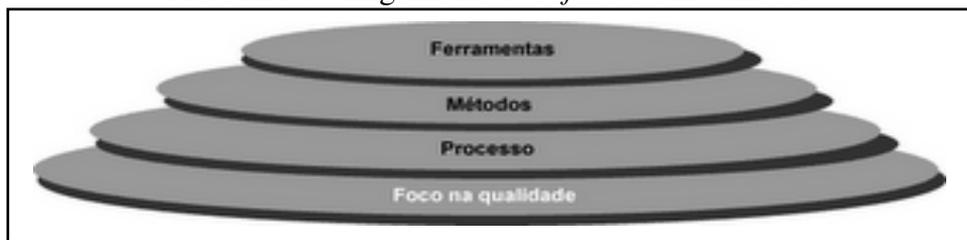
O conceito de engenharia de software foi inicialmente sugerido em 1968, em uma conferência a NATO Science Committee que discutia a “crise de software”. O termo enunciava as complicações do desenvolvimento de software diante das rápidas mudanças no ambiente e do alto crescimento da demanda por *software*. Enunciava também o nível de complexidade dos problemas a serem resolvidos e a inexistência de técnicas e modelos estabelecidos para que o desenvolvimento de *software* fosse executado de forma satisfatória.

Sommerville (2007) declara que a engenharia de *software* pode ser considerada uma disciplina da engenharia, a qual tem como objetivo o desenvolvimento de sistemas de *software* com a melhor relação custo-benefício possível. Engloba todos os aspectos da

produção de *software*, indo desde estágios iniciais como a análise de requisitos até sua manutenção depois de concluído.

Na visão de Pressman (2006), a engenharia de *software* é uma tecnologia em camadas que se amparam em um compromisso organizacional com a qualidade, o que remete a um aperfeiçoamento contínuo, por isso é baseado na camada de foco na qualidade. A Figura 4 representa essa abordagem.

FIGURA 4 - Engenharia de *software* em camadas



Fonte: Pressman (2006).

O autor assegura que o alicerce da engenharia de *software* é a camada de processo, já os métodos de engenharia de *software* fornecem a técnica de “como fazer” para construir *softwares*. Tarefas como comunicação, análise de requisitos, construção de programas, testes, entre outros fazem parte destes métodos que são auxiliados pelas ferramentas.

Ou seja, a engenharia de *software* abrange etapas que envolvem métodos, ferramentas e procedimentos. Entretanto, não pode-se classificar uma abordagem como sendo a melhor para a engenharia de *software* em virtude dos mais diversos tipos de *softwares* e organizações que os utilizam. Isso não significa que noções básicas de processo e de organização de sistemas devam ser ignoradas.

2.3.1 Processo de desenvolvimento de *software*

Um processo pode ser definido a partir do momento em que pode ser repetido. Encontra-se em Souza (2004), que processos são operações realizadas no desenvolvimento de um produto desde seu início até o final.

Trazendo esse conceito para o mundo do *software*, tanto Lonchamp (1993) quanto Pessoa (2003) afirmam que, para um bom funcionamento, o processo deve ser formado por passos ordenados com o objetivo de produzir e garantir que o produto de *software* acabado seja realmente o que foi solicitado. “Um processo de *software* descreve uma abordagem para a construção, implantação e, possivelmente, a manutenção de *software*” (LARMAN, 2004, p.37).

Assim, muitos engenheiros de *software* afirmam que garantindo qualidade do processo de desenvolvimento de *software* consequentemente se assegurará a qualidade do produto final (KITCHENHAM, 1996; PFLEEGER, 1998; HUMPRHEY, 1989 e HARVEY, 1994).

Por isso, Pressman (2006) afirma que existem algumas atividades que são fundamentais no processo de construção de um *software*, ou seja, o ciclo de vida de todo processo de desenvolvimento deve cumprir um conjunto mínimo de atividades para se alcançar um produto de *software*. São elas:

- a) comunicação: envolve muita comunicação e colaboração com o cliente e outros envolvidos além de compreender o levantamento de requisitos e outras atividades relacionadas: procura descobrir, determinar, especificar e documentar as funcionalidades gerais do *software*;
- b) planejamento: estabelece um lineamento para o trabalho de desenvolvimento do *software* descrevendo as tarefas técnicas a serem realizadas, os riscos do projeto, os recursos fundamentais, os artefatos que serão produzidos e um cronograma de trabalho;
- c) modelagem: cria os modelos que possibilitam compreender melhor os requisitos do *software* e o projeto que vai atender a esses requisitos. Nesse momento a representação de engenharia do *software* será construída;
- d) construção: combina a criação do código-fonte, que deve conter as especificações dos requisitos com os testes necessários para encontrar erros na função, no comportamento e no desempenho do *software*;
- e) implantação: momento da entrega do *software* junto ao cliente que avalia o produto e devolve um feedback com base na avaliação.

Realizar todas as etapas acima no decorrer do desenvolvimento de *software*, principalmente as de comunicação, garantem que atividades básicas sejam realizadas evitando assim que se entregue um produto com baixa qualidade.

2.3.2 Métricas e estimativas

As medições e as métricas permitem verificar um melhor entendimento do processo utilizado para desenvolver um produto, inclusive permite identificar uma melhor avaliação do próprio produto.

O levantamento de esforços e estimativas de custo devem ser realizados com cautela e alguma metodologia. Estimativas elevadas podem fazer com que os clientes cancelem suas demandas de projetos, impactando inclusive na credibilidade da empresa, enquanto que, quando realizadas abaixo do real, podem forçar a equipe a trabalhar mais sem a remuneração adequada gerando reflexos na qualidade, pois com um cronograma apertado alguns cuidados extras acabam sendo deixados de lado.

Para Pressman (2006), estimar é uma arte e uma ciência ao mesmo tempo, porém por se tratar de uma atividade de extrema importância, não pode ser conduzida de modo aleatório. Técnicas existem para estimativa de tempo e esforço, onde métricas de processo e projeto fornecem uma perspectiva histórica importante. Segundo o autor, experiências de todo o pessoal envolvido ajudam muito à medida que estimativas são desenvolvidas e reavaliadas.

O risco de uma estimativa é medido pelo grau de incerteza das estimativas quantitativas. Se o escopo do projeto é mal entendido ou se os requisitos mudam com frequência, a incerteza e o risco tornam-se perigosamente altos (PRESSMAN 2006).

Realizar de forma adequada a relação entre custo do projeto e o preço do mesmo não é uma tarefa simples. Sommerville (2003) prega que estimativas de produtividade geralmente se baseiam em medir alguns atributos de *software* e dividir o resultado pelo esforço total exigido para o desenvolvimento. Para ele dois tipos de medidas são mais utilizadas: Medidas relacionadas a tamanho e medidas relacionadas a funções.

Porém para o autor cada técnica tem seus pontos positivos e negativos, por isso, em grandes projetos é importante a utilização de várias técnicas e após um comparativo entre seus resultados, identificando qual a mais adequada ao projeto.

Neste sentido Pressman (2006), conclui dizendo que a estimativa de projetos de *software* nunca será uma ciência exata, mas uma combinação de bons dados históricos juntamente com técnicas sistemáticas, assim é possível tornar uma estimativa mais precisa. Por isso da importância de um requisito claro e detalhado, pois o mesmo impacta diretamente na estimativa de tempo e valor conforme a necessidade, criticidade e complexidade.

2.3.3 Requisitos de *software*

Um fator preponderante de sucesso para um *software* é o grau em que atende aos requisitos para os quais foi construído. A fase de análise de requisitos é extremamente importante no desenvolvimento de um sistema, pois é o alicerce para o desenvolvimento do *software*.

Segundo o IEEE Standard Glossary of *Software Engineering Terminology* ou Glossário de termos de *software* do IEEE, requisito é:

- a) uma condição ou capacidade necessária para o usuário resolver um problema ou alcançar um objetivo;
- b) uma condição ou capacidade que deve ser encontrada ou possuída por um sistema ou componente do sistema para satisfazer um contrato, padrão, especificação ou outro documento imposto formalmente ou;
- c) uma representação documentada de uma condição ou capacidade como em (a) ou (b).

Por isso, os requisitos de *software* determinam o que o sistema deve fazer e sob quais circunstâncias deve operar. Outra boa definição para requisitos é dada por Leite (1994), que explica que requisitos são uma condição fundamental para a aquisição ou para o preenchimento de certo objetivo. Como os requisitos formam a base para o desenvolvimento de *software*, é possível dizer que requisitos de má qualidade provavelmente também irão gerar um *software* de baixa qualidade. Isso pode ser consequência da falta de processos definidos para o levantamento destes requisitos o que levará a um documento que não expressa à verdadeira necessidade do cliente.

Isso pode acontecer em dois cenários distintos. Um ocorre pela falta de conhecimento das técnicas por parte de analistas e gerentes de projetos e o outro ocorre pelo excesso de intimidade com o cliente o que faz com que os profissionais responsáveis por esse levantamento suponham conhecer tão bem o processo e os problemas a ponto de deixarem de lado considerações importantes dos clientes. É possível diminuir tais falhas documentando e compartilhando estes requisitos, produzindo uma rica e completa definição dos mesmos.

Pfleeger (2004), questiona porque os requisitos são importantes. Ele mesmo responde com uma pesquisa feita pelo Standish Group em 1994 que pesquisou mais de 350 empresas sobre seus mais de 8000 projetos de *software*. 31% dos projetos foram cancelados antes da sua conclusão e apenas 9% foram entregues dentro do prazo e valor estimado nas grandes empresas.

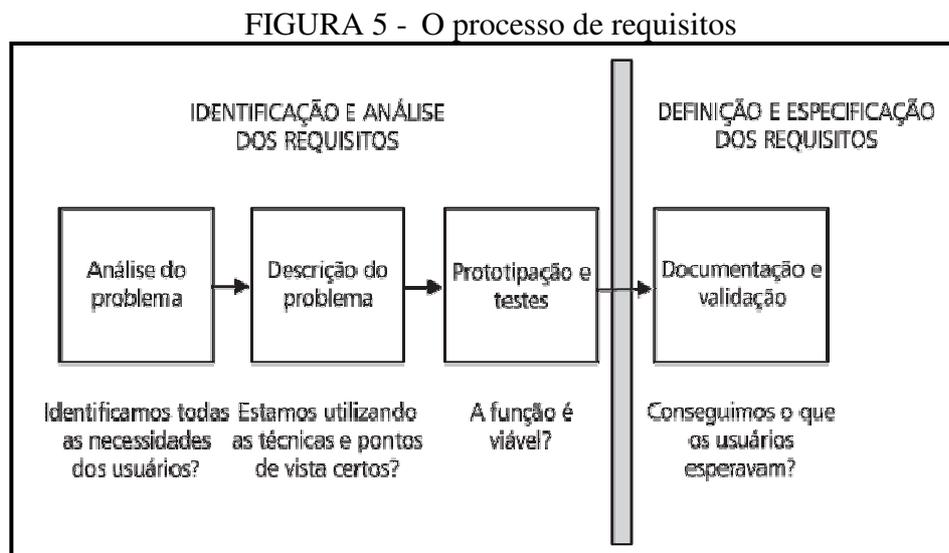
O mais esclarecedor e que também responde muito bem o questionamento feito pela autora é que requisitos incompletos representam a maior causa dos projetos terem sido falhos segundo os entrevistados, com 13,1%. Outras partes de definição, identificação e gerenciamento de requisitos também aparecem em outras respostas, como expectativas não realistas com 9,9% e modificações nos requisitos e nas especificações com 8,7%.

Outros estudos, como o de Boehm (1984), indicam que o custo de correção de um erro encontrado depois do *software* implementado é muito maior, erros em requisitos de *software* são até 20 vezes mais caros de se corrigir que qualquer outro tipo de erro.

Leffingwell (1997) afirma que em torno de 40% a 60% dos problemas detectados em um projeto são cometidos na fase de análise, ou seja, no levantamento dos requisitos, e que estes podem ser facilmente detectados. As consequências de não se detectar os erros nesta fase impactam diretamente na não satisfação do usuário, em um possível desentendimento entre os envolvidos, e na perda de tempo e dinheiro das partes.

Muitos problemas na construção do *software* se originam da imprecisão na especificação dos requisitos. É natural para um desenvolvedor de sistemas interpretar um requisito ambíguo para simplificar sua implementação. Isso faz com que na maioria das vezes novos requisitos sejam estabelecidos e mudanças no sistema sejam efetuadas (SOMMERVILLE, 2003).

Assim, percebe-se que a identificação dos requisitos é parte extremamente importante no processo de desenvolvimento de *software*. Segundo Pfleeger (2004), é necessário a utilização de várias técnicas para determinar o que os usuários e clientes realmente querem. A Figura 5 ajuda a ilustrar o processo de identificação dos requisitos de um sistema:



Fonte: Pfleeger (2004).

Segundo Pressman (2006), todos nós lutamos quando tentamos levantar requisitos com nossos clientes, pois temos dificuldade de entender a própria informação conseguida. Poucos são os registros destes requisitos levantados e pouco tempo é gasto para se verificar o que realmente foi registrado. Para o autor ao invés de criarmos mecanismos de controle e

gerenciamento destes requisitos, permitimos que as modificações nos controlem, falhando ao tentar estabelecer uma base consistente para o *software* que será construído. Quando estes problemas se combinam o futuro é incerto, porém existem soluções para contornar estas dificuldades.

Uma solução é a Engenharia de Requisitos (ER) que de forma geral, é o processo de identificação de todos os envolvidos, de localizar os objetivos e necessidades e documentá-los de forma correta para análise, comunicação e subsequente implementação (SOMMERVILLE, 2003; PRESSMAN, 2006).

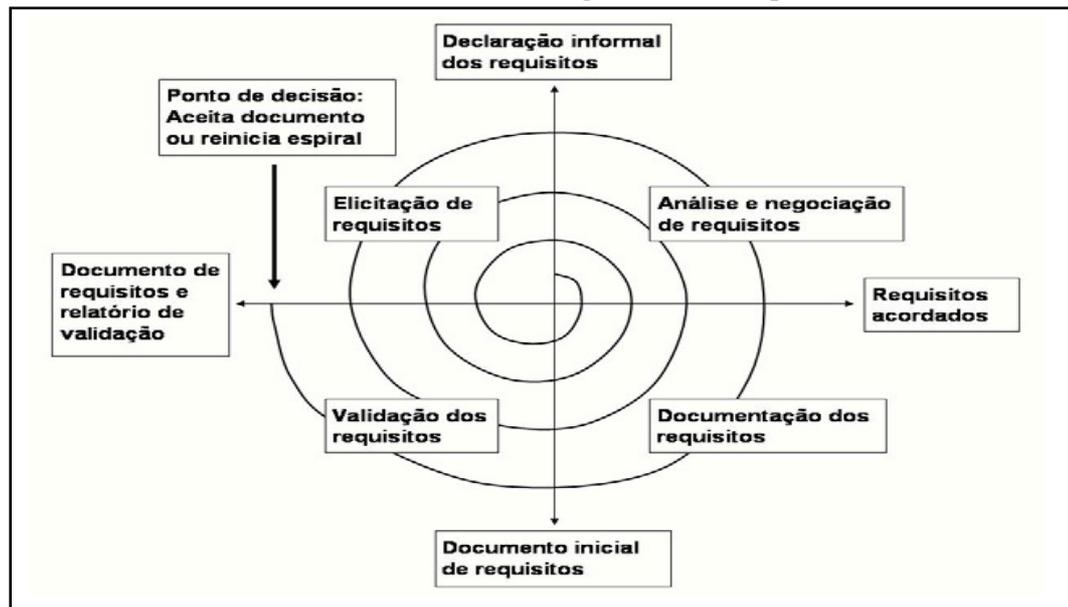
Segundo Thayer (2000), a engenharia de requisitos pode ser definida como a ciência e disciplina preocupada com a análise e documentação dos requisitos, incluindo análise das necessidades e análise e especificação dos próprios requisitos. Além disso, a engenharia de requisitos fornece mecanismos apropriados para facilitar as atividades de análise, documentação e verificação.

Ainda de acordo com Pressman (2005), engenharia de requisitos auxilia os engenheiros de *software* a compreenderem melhor o problema que precisam resolver, se utilizando para isso de tarefas que ajudam a chegar a um entendimento da dimensão que o sistema dará ao negócio, além de levantar o que os usuários querem e como interagirão com o sistema.

Um processo de engenharia de requisitos é um conjunto estruturado de atividades que são obedecidas para derivar, validar e manter um documento de requisitos de um sistema. Uma descrição completa desse processo inclui as atividades que devem ser guiadas, o mecanismo de execução destas atividades, as entradas e saídas de cada atividade e as ferramentas utilizadas para suportar a engenharia de requisitos (SOMMERVILLE e SAWYER, 1997).

Kotonya e Sommerville (1998), definem a engenharia de requisitos em um processo com quatro etapas conforme a Figura 6:

FIGURA 6 - Processo de engenharia de requisitos



Fonte: Kotonya e Sommerville (1998).

O que será visto a seguir é o que os autores falam sobre estas quatro etapas:

- a) elicitação: É o levantamento dos requisitos, seja por meio de entrevistas com os usuários, clientes e especialistas, consulta a documentação do sistema, ou conhecimento tácito e explícito sobre o domínio de conhecimento em questão;
- b) análise e negociação: Avaliar e revisar o escopo do *software*, envolvendo negociação com diferentes interessados, estabelecendo em conjunto um acordo dos requisitos, os quais devem estar consistentes e sem ambiguidades, para que possam ser utilizados como base para o desenvolvimento do *software*;
- c) documentação: É a documentação dos requisitos acordados, que serve como meio de comunicação entre gerente de projeto/analista e cliente, formalizando-os em um nível de detalhe que todos os interessados consigam entender, geralmente por meio de linguagem natural e diagramas. Analistas experientes possuem maior facilidade na atividade de construção de *software* por possuírem conhecimento de soluções recorrentes que podem ser aplicadas em diversas situações similares. Tais soluções podem ser documentadas adequadamente no formato de padrões para facilitar a sua utilização diversas vezes, sem, no entanto implementar a solução da mesma forma duas vezes (SOMMERVILLE e SAWYER, 1997);
- d) Validação: Trata da verificação dos requisitos em termos de consistência e completude, verificando os requisitos e modelos documentados se atendem às reais necessidades dos usuários antes que sejam utilizados no desenvolvimento do sistema.

Para que a definição de requisitos seja realizada com sucesso, segundo Ratchev et al. (2003), é fundamental uma correspondência exata entre os requisitos do cliente, capacidade do produto e conhecimento do processo por parte do analista. Para os autores apesar dos recentes desenvolvimentos na área, ainda existe falta de transparência e definição coerente na integração das atividades de engenharia de requisitos.

Ainda segundo o trabalho dos autores que é parte do projeto ESPRIT colaborativo *Knowledge acquisition and sharing for requirement engineering* (KARE) financiado pela Comissão Européia, há falta de métodos estruturados para a captura do conhecimento empresarial relevante para posterior implementação em suporte de tomada de decisão no processo de especificação de requisitos.

Kudikyala e Vaughn (2005) abordam a definição de requisitos sob outra ótica onde aplicam uma técnica já utilizada na engenharia de *software*, mas desta vez na engenharia de requisitos. O trabalho dos autores é utilizar modelos mentais para reduzir os problemas de verificação e validação no processo de definição de requisitos.

Para os autores a evidência empírica obtida no estudo indica que tais modelos podem também ser úteis na identificação de mal entendidos, requisitos duplicados e ambíguos. A técnica utilizada é conhecida como redes *pathfinder* (PFNETs). Nesse caso um *software* que aplica a técnica PFNET é utilizado para analisar documentos de requisitos de cliente e desenvolvedor separadamente.

No estudo feito pelos autores o principal benefício é a capacidade de gerar um modelo com aspectos de memória semântica humana. Ele fornece a capacidade de avaliar matematicamente e comparar estas redes de semelhanças e diferenças para revelar possíveis mal entendidos.

Um dos problemas para a falta de métodos referenciado no trabalho de Ratchev pode ser o que relatam Shaw e Gaines (1992) quando dizem que os clientes geralmente codificam o conhecimento relacionado às suas necessidades, em textos, desenhos ou mensagens verbais, ou seja, em informação. A conversão desse conhecimento em informação explícita depende muito do uso do conhecimento tácito, o conhecimento pessoal que está implícito e é dificilmente transmitido. Este conhecimento juntamente com a forma de transmiti-lo, fazem parte da gestão do conhecimento e serão estudados a seguir.

2.4 GESTÃO DO CONHECIMENTO

A concepção do conceito de Gestão do Conhecimento segundo Sveiby (1998) possui três origens: nos Estados Unidos originou-se da inteligência artificial, quando foi possível identificar que grande parte dos sistemas se tornara obsoleto após seis meses. Dessa forma o conhecimento foi identificado como necessário na condução dos negócios. No Japão, desde 1980 a preocupação se relacionava aos temas de inovação e conhecimento, juntamente com a escassa valorização dos ativos intangíveis por não estarem descritos nos balanços das organizações. Já na Suécia existiam preocupações com medições estratégicas baseadas em competência, que invariavelmente dependiam do conhecimento dos funcionários das organizações. Estas preocupações conduziram a uma abertura para a gestão do conhecimento (SVEIBY,1998).

Nos últimos 35 anos a revolução da informação ocasionou mudanças na economia global transformando alguns padrões de competitividade e produtividade. Nesse novo modelo econômico as capacidades de gerenciamento da informação e do conhecimento juntamente com o processo de aprendizagem figuraram como diferenciais competitivos no desenvolvimento das competências organizacionais (CASTELLS, 1999).

Drucker (1993) define o conhecimento necessário para a organização, como aquele capaz de dotar de relevância e propósito os dados existentes, transformando-os em informação valiosa para o negócio. O autor também afirma que na medida em que as organizações pensam dessa forma, ou seja, tratam a conversão de dados brutos em informação, a maneira de se trabalhar e os processos de tomada de decisão são alterados.

Davenport (1998) coloca que não existe uma definição literal para conhecimento ocorrendo o mesmo para o termo gestão do conhecimento. Entretanto obras como a de Nonaka e Takeuchi (1997) e mesmo Davenport (1998), indicam que a gestão do conhecimento é uma atividade consciente e intencional que pretende conceder sustentabilidade ao desenvolvimento de uma empresa.

Carvalho (2000) destaca que fazer uso apropriado da informação, e potencializá-la em conhecimento/inteligência além de agregar valor aos seus produtos e processos é um desafio para empresas de qualquer porte e setor.

Davenport e Prusak (1998) afirmam que gestão do conhecimento é a prática de geração, codificação, coordenação e transferência que se faz do conhecimento. Já para Sveiby (2003) gestão do conhecimento pode ser vista como uma ferramenta estratégica competitiva

que possui a capacidade de aproveitar os recursos existentes na própria organização ao mesmo tempo em que propicia aos empregados as melhores práticas.

Já Nonaka e Takeuchi (1997) não tratam o conceito de gestão do conhecimento, mas sim do “conjunto de processos que orientam a criação, divulgação e utilização do conhecimento, visando alcançar os objetivos da organização”.

Grande parte dos gestores pensa a gestão da informação e a gestão do conhecimento como sendo relativas à implantação de uma ou de várias tecnologias que fornecerão maior agilidade às questões informacionais, deixando de lado outros elementos primordiais para esse gerenciamento, como por exemplo, a cultura organizacional, a comunicação, a estrutura (formal e informal), a racionalização (fluxos e processos), as redes de relacionamento etc (VALENTIM, 2002).

Valentim (2002) deixa claro que os papéis da gestão da informação e da gestão do conhecimento são diferentes, porém complementares. Só assim é possível ter o entendimento da gestão da informação como um grupo de estratégias que pretende reconhecer as carências de informação, mapear os fluxos formais com a finalidade de apoiar o desenvolvimento das atividades do dia a dia e a tomada de decisão na esfera corporativa. E a gestão do conhecimento, por sua vez, é entendida como um grupo de estratégias para instituir, alcançar, compartilhar e usar ativos de conhecimento, além de fundamentar fluxos que garantam a informação necessária no tempo e formato condizentes, a fim de cooperar na criação de ideias, solução de problemas e na tomada de decisão (VALENTIM, 2002).

Ainda que conhecimento e sua gestão sejam condições essenciais e de grande relevância para a conquista de um posicionamento estratégico eficiente nas organizações, apenas a pura e simples gestão desse conhecimento acaba sendo insuficiente para a criação de habilidades dentro da organização. As organizações dos tempos atuais são desafiadas a formar estratégias organizacionais que possam ir além, tratando da criação de novos conhecimentos, e colaborando na criação e manutenção de competências.

2.4.1 Criação de conhecimento organizacional

Em busca do desenvolvimento de novos conhecimentos, o surgimento de teorias que miram o entendimento do processo de criação de conhecimento organizacional passa a ser extremamente fundamental. Um exemplo é a teoria de criação do conhecimento organizacional de Nonaka e Takeuchi (1997), que tem como base a identificação do

conhecimento tácito tratando-o como o elemento principal na criação do conhecimento organizacional. Este entendimento deixa claro a dependência direta do conhecimento com as pessoas e, obviamente, com a ação humana, que envolve a criação de um ambiente organizacional favorável ao desenvolvimento de habilidades como a autonomia, criatividade, compartilhamento e disponibilidade das informações.

A criação do conhecimento nas organizações está conectada à habilidade da empresa em construir e propagar conhecimento, incorporando-o a sistemas, produtos e serviços beneficiando-se, para isso, do componente mais básico e universal da organização: o conhecimento humano (NONAKA E TAKEUCHI, 1997)

Na definição sugerida por Terra (2000), é possível compreender a carência de renovação do conhecimento, pois caso isso não ocorra, as vantagens alcançadas podem deixar de existir. Assim:

A Gestão do Conhecimento está ligada à capacidade das empresas em utilizarem e combinarem as várias fontes e tipos de conhecimento organizacional para desenvolverem competências específicas e capacidade inovadora, que se traduzem, permanentemente, em novos produtos, processos, sistemas gerenciais e liderança de mercado (TERRA, 2000, p.70).

Segundo Nonaka e Takeuchi (1998) a base para sua teoria do conhecimento está na distinção entre dois tipos existentes, o conhecimento tácito e o explícito. O conhecimento tácito é pessoal e difícil de ser formalizado, ele está enraizado nas ações, procedimentos, valores, emoções, etc. É a forma menos conhecida, ou não convencional do conhecimento. Este conhecimento não é codificado, e sim adquirido pela partilha de experiências, através da observação e imitação (KIKOSKI e KIKOSKI, 2004; HALL e ANDRIANI, 2003). Conhecimento tácito e explícito são complementares, o que significa que ambos são essenciais para a criação de conhecimento uma vez que o conhecimento é criado através de interações entre eles (NONAKA, TOYAMA e KONNO, 2000).

Nonaka, Toyama e Konno, 2000) e outros autores, como Kikoski e Kikoski (2004) descrevem o conhecimento explícito como aquilo que pode ser traduzido em um código ou uma linguagem e, conseqüentemente, pode ser verbalizado e comunicado, processado, transmitido e armazenado de maneira clara. Na maioria das vezes é público, amplamente conhecido e pode ser encontrado em formas convencionais de conhecimento como em livros, revistas e meios de comunicação em massa, como jornais, televisão, internet etc.

A partir da interação entre conhecimento tácito e explícito segundo Nonaka e Takeuchi (2008) surgem os quatro modos de conversão de conhecimento – Socialização, Externalização, Combinação e Internalização. Esses modos são os mecanismos através dos

quais o conhecimento individual é articulado e desenvolvido na organização, por isso são considerados o motor do processo de criação do conhecimento.

Para Nonaka (*apud* Nonaka e Takeuchi, 1998), trocas entre conhecimento tácito e explícito podem ser definidas como um processo de conversão social e através dessa conversão esses conhecimentos se espalham tanto em termos de qualidade quanto de quantidade. Esta ideia permitiu que fosse desenvolvido um modelo, que contempla os quatro modos de conversão de conhecimento conforme visto na Figura 7 onde a transformação do conhecimento se dá a partir do resultado de um processo contínuo, interativo e em movimento espiral:

FIGURA 7 - Quatro modos de conversão do conhecimento



Fonte: Adaptado de Nonaka e Takeuchi (1998).

Como é possível observar, o modelo proposto por Nonaka e Takeuchi (1998), contém quatro modos para conversão do conhecimento: socialização, externalização, internalização e combinação. A utilização de cada modo ocorre conforme o conhecimento existente e para o qual ele está se transformando. A seguir um breve resumo para cada um dos modos de conversão contidos no modelo:

a) socialização – do conhecimento tácito em conhecimento tácito;

- a socialização é um processo de compartilhamento de experiências como, por exemplo, modelos mentais ou habilidades técnicas compartilhadas, onde um indivíduo pode adquirir conhecimento tácito diretamente de outros sem nem mesmo usar a linguagem, ou seja, pela simples observação. O segredo para a aquisição do conhecimento tácito é a experiência. Sem experiência compartilhada, é praticamente impossível para uma pessoa compreender o processo de raciocínio do outro indivíduo;

- uma atenção especial deve ser dada ao conhecimento tácito e ao processo de socialização do conhecimento, pois conforme Terra (2000) a principal vantagem competitiva que as empresas possuem está baseado no capital humano, ou seja, no conhecimento tácito que seus funcionários possuem;

b) externalização - do conhecimento tácito em conhecimento explícito;

- na externalização a transferência do conhecimento ocorre em um processo de junção do conhecimento tácito em conceitos explícitos através do uso basicamente da linguagem expressa na forma de metáforas, analogias, conceitos e hipóteses ou modelos, provocada pelo diálogo e pela reflexão coletiva auxiliando os indivíduos a articularem o conhecimento tácito oculto que, de outra forma, se torna mais difícil de ser comunicado;

- a metáfora é uma forma de perceber ou assimilar de forma intuitiva uma coisa imaginando de forma simbólica outra. As contradições ligadas a uma metáfora são harmonizadas pela analogia, o que diminui o desconhecido dando destaque ao caráter comum de duas coisas diferentes;

c) combinação - do conhecimento explícito em conhecimento explícito;

- a combinação está baseada na sistematização de conceitos em um sistema de conhecimento, englobando a combinação de diferentes conhecimentos explícitos. Também pode-se dizer que são diferentes conjuntos de conhecimento explícitos sendo trocados. Essa conversão é realizada pelos indivíduos a partir de trocas ou combinação de conhecimento utilizando meios como documentos, reuniões, conversas ao telefone ou redes de comunicação computadorizada. Esta forma de conversão de conhecimento pode ser realizada através da educação e do treinamento formal dado em escolas;

d) internalização – do conhecimento explícito em conhecimento tácito.

- processo de incorporação do conhecimento explícito em conhecimento tácito sob a forma de modelos mentais e *know-how* compartilhado com os outros integrantes da organização, iniciando uma nova espiral de criação de conhecimento o qual também pode ser entendido como “aprender fazendo”. Na visão dos autores a internalização das experiências através dos modos de socialização, externalização e combinação para o conhecimento tácito dos indivíduos são considerados ativos valiosíssimos. Esse conhecimento acumulado pelos indivíduos no processo de internalização pode e deve ser

utilizado como meio para estimular a propagação do conhecimento existente e a partir deste promover a criação do conhecimento novo.

Segundo Borges e Falbo (2001), a gestão da informação e do conhecimento permite oferecer alternativas de apoio na melhoria contínua do processo de desenvolvimento de *software*, além de permitir o aprendizado da organização como um todo auxiliando no compartilhamento do conhecimento. Parreiras (2004) vai mais além, afirmando que o processo de desenvolvimento de *software* está intimamente ligado à gestão da informação e do conhecimento, em virtude da possibilidade de se mapear, sistematizar, tratar e propagar adequadamente o conhecimento.

Natali (2003) comenta que a gestão do conhecimento é tratada como uma fonte de vantagem competitiva dentro das fabricas de desenvolvimento de *software*. Além disso, segundo a autora a gestão do conhecimento apresenta-se como uma forma de apoiar os desenvolvedores, permitindo que executem suas atividades da melhor e mais rápida forma, fazendo uso de experiências anteriores.

Estas experiências ajudam de diversas formas integrantes de um grupo de desenvolvimento de *software*. Caso um indivíduo queira aprender sobre uma nova tecnologia desenvolvida e aplicada em determinado projeto a partir de experiências anteriores, este conhecimento já obtido pode sem duvida auxiliar outros indivíduos na implantação e desenvolvimento desta nova tecnologia.

Da mesma forma no ambiente de programação, experiências de outros programadores podem auxiliar no desenvolvimento de *software*. Por exemplo, se existirem categorias referentes à programação, *bugs*, controle de qualidade, relatórios, etc., em um repositório de conhecimento, caso um programador encontre um erro no momento da codificação, ele pode consultar o repositório ler e aprender como resolver o problema encontrado. Isto impede a reinvenção da roda, evita desperdício de tempo e desgaste na busca de uma solução que já existe, tornando o processo de codificação mais simples e agradável.

Abecker et al. (1998) relata que uma infra-estrutura que gerencie o conhecimento pode ser tratada como sistema de suporte à gerência do conhecimento em uma organização, utilizando a memória organizacional como o centro desse sistema fornecendo aos usuários informações e conhecimento necessário para execução de outras tarefas. O'Leary (1998) afirma que os repositórios de conhecimento são de forma mais específica, a memória organizacional, os quais podem ser utilizados por máquinas ou por usuários. Rocha reforça a importância do armazenamento e gerenciamento das informações afirmando:

O armazenamento e o gerenciamento da informação adquirida por uma empresa ao desempenhar as atividades dos seus processos de negócios devem ser representados na memória organizacional de um sistema de gestão do conhecimento. O conhecimento armazenado, se reutilizado para melhorar o conhecimento sobre o desempenho de uma empresa, pode proporcionar o aprendizado organizacional. Esse aprendizado pode ser apoiado pelo armazenamento e pelo uso de lições aprendidas, visando evitar a repetição dos erros realizados em experiências passadas (COSTA e ROCHA, 2003, p, 343).

Para Fischer e Oswald (2001), trabalhadores devem adquirir um novo pensamento no qual eles não são mais receptores passivos do conhecimento, e sim pesquisadores, construtores e comunicadores ativos do conhecimento. Os autores afirmam que o conhecimento não deve ser criado apenas por gerentes, mas deve sim ser construído de forma colaborativa, no contexto do trabalho. Dessa forma a memória organizacional que é uma representação explícita do conhecimento e das informações capitais para uma organização, e que tem como uma das principais funções aumentar a competitividade da organização, pelo aperfeiçoamento da forma como ela gerencia seu conhecimento (ABECKER, 1998), evolui com o tempo, promovendo o compartilhamento e disseminação de informação relevante para toda a organização. Por isso a armazenagem deste conhecimento deve ser eficiente, visando a facilitação do seu reuso.

Lindvall et al. (2001) apresenta alguns benefícios da institucionalização de programas de Gestão do Conhecimento nesse contexto:

- a) reutilização eficiente de experiências documentadas;
- b) facilidade de encontrar soluções para problemas dentro da organização;
- c) identificação e armazenamento de experiências de valor;
- d) facilidade de propor medidas para melhorar a execução de processos e aumentar a qualidade de produtos de *software*.

Para Basili, Lindvall e Costa (2001), a representação explícita de conhecimentos sobre processos de *software*, pode potencializar a produtividade dos membros da empresa através da definição de programas de treinamento. Já Landes (1999) destaca que soluções de gestão do conhecimento aparam de forma eficaz atividades de membros da organização com pouca experiência em uma área ou domínio específico.

2.4.2 O Conhecimento e a gerência de requisitos

Conforme Damm (2002), o conhecimento produzido em projetos pode ser categorizado como conhecimento em projetos, conhecimento sobre projetos e conhecimento

proveniente de projetos. O conhecimento em projetos abrange dados e informações detalhadas que são produzidas ao longo do mesmo.

O conhecimento proveniente dos projetos é fundamental para que as pessoas envolvidas saibam como e quais projetos estão sendo encaminhados em algum momento. Isso contribui para a promoção de uma coordenação eficiente e efetiva das atividades, além do controle dos recursos utilizados nos projetos.

Já o conhecimento pertinente ao projeto corresponde às informações importantes geradas com a execução deste. Este conhecimento é definitivo para a busca do sucesso no projeto e contribui para o aprendizado organizacional.

Durante a execução de um projeto, particularmente na etapa de levantamento de requisitos são necessárias várias informações detalhadas sobre eles e as requisições dos clientes. Dentre estas informações, pode-se destacar algumas que podem ser consideradas como conhecimento em projeto:

- a) requisições de alteração de requisitos e seus respectivos estados;
- b) relatórios de análise de impacto das alterações requisitadas;
- c) estimativas de custo e esforço para novos requisitos ou requisições de alteração em requisitos já existentes;
- d) projetos em andamento e as necessidades organizacionais tratadas pelos mesmos;
- e) produtos de *software* sendo criados ou modificados por cada projeto;
- f) indivíduo ou equipe responsável por requisitos em cada projeto;
- g) métricas relacionadas aos requisitos.

No processo de gestão de requisitos pode-se identificar os quatro modos de conversão de conhecimento propostos por Nonaka e Takeuchi (1998) os quais abrangem informações de origens e objetivos distintos.

O compartilhamento de experiências entre os integrantes do projeto, principalmente sob a forma da linguagem e observação, é segundo Nonaka e Takeuchi (1998), o modo de socialização onde ocorre a conversão do conhecimento tácito em conhecimento tácito. Na busca do entendimento dos desejos e objetivos do projeto de *software* analistas de sistemas adquirem com usuários e gerentes uma expertise no contexto do projeto a ser desenvolvido através do compartilhamento de experiências. Isso vem ao encontro do que dizem Nonaka e Takeuchi (1998) sobre a relação entre aquisição do conhecimento tácito e envolvimento, acentuado pelas emoções associadas nos contextos particulares nos quais as experiências partilhadas são embutidas.

A conversão do conhecimento tácito em conhecimento explícito, caracterizado como modo de externalização segundo Nonaka e Takeuchi (2008) ocorre na medida em que os participantes do projeto criam suas representações em forma de metáforas, pareceres, hipóteses ou modelos. Documentos que são fundamentais no processo de especificação e levantamento de requisitos e são normalmente a forma adquirida dessas representações. De acordo com Nonaka e Takeuchi (1998), a externalização é o principal modo para a criação do conhecimento, pois cria conceitos novos e explícitos a partir do conhecimento tácito.

Outro modo de conversão percebido na engenharia de requisitos é a combinação, que segundo Nonaka e Takeuchi (1998) ocorre quando existe conversão do conhecimento explícito em conhecimento explícito. É possível notá-la quando da sistematização de documentos e conceitos criados por analistas no momento em que se estuda toda a documentação inerente ao projeto e a sua respectiva transformação em documentos de especificação de requisitos com um nível de detalhamento elevado a fim de permitir que a fase seguinte do processo de desenvolvimento de *software* ocorra sem problemas. Na etapa seguinte a combinação das tecnologias selecionadas possibilita que protótipos sejam construídos e utilizados para auxiliar no processo de construção do *software*.

Por fim a documentação sistematizada passa pela homologação e análise de viabilidade dos indivíduos envolvidos no projeto os quais utilizam a verbalização e esquematização do conhecimento sob diversas formas como, por exemplo, manuais e histórias orais. Esse processo também é conhecido como incorporação do conhecimento, onde existe a conversão de conhecimento explícito em conhecimento tácito, ou seja, o modo de internalização segundo Nonaka e Takeuchi (1998). Os autores declaram ainda que para que ocorra a criação do conhecimento organizacional, é fundamental que o conhecimento tácito capitalizado seja difundido para o restante dos membros da organização, para que se inicie uma nova espiral de criação do conhecimento.

A criação do conhecimento organizacional é uma interação contínua e dinâmica entre o conhecimento tácito e o conhecimento explícito (NONAKA e TAKEUCHI, 1998). Esse tipo de interação é percebido entre usuários e gerentes antes do desenvolvimento do produto e após sua implantação. Dessa forma a internalização produz conhecimento sobre o gerenciamento de requisitos o qual é socializado iniciando assim um novo ciclo, de acabamento do produto existente ou desenvolvimento de uma inovação.

Alguns trabalhos recentes, como o de Scott (2008), confirmam a importância do conhecimento tácito existente no capital intelectual das empresas. O estudo de Scott (2008) examinou a criação e partilha de conhecimentos e percebeu que grande parte do conhecimento

que as pessoas mais experientes possuem é tácito e aprendido com a experiência pessoal e de interagir com outros empregados. A maioria do conhecimento tomado como certo não é registrado em qualquer banco de dados, manual de procedimentos, ou programa. Nessa mesma linha, alguns autores indicam que criar e compartilhar conhecimento de forma mais eficaz do que os concorrentes possibilita ganho de vantagem competitiva (BOISOT, 1998; GRANT, 1996; KOGUT e ZANDER, 1992; TEECE, 1998). Além da competitividade, o conhecimento é particularmente importante para as organizações em processos de mudança.

Nesse sentido, Cecez-Kecmanovic e Jerram (2002) trabalham com o modelo de gestão do conhecimento baseado em *Sensemaking*. O modelo dos autores permitiu compreender os processos de criação e partilha de conhecimento em um momento de mudança organizacional. Segundo os autores, no estudo se distinguiu diferentes tipos de conhecimento em diferentes níveis: individual, coletivo, organizacional e incorporado na cultura. Com base nesse entendimento, o trabalho dos autores possibilitou ter a compreensão das necessidades dos *stakeholders* em vários momentos dos processos de gestão aumentando a compreensão das mudanças nos processos e os seus problemas essenciais.

Outros estudos feitos na área dão conta que a partilha de conhecimento tácito está sujeita a interação social (KÄSER e MILES, 2002; NONAKA, 1994; OSTERLOH e FREY, 2000). Yang e Farn (2009) investigaram o compartilhamento do conhecimento tácito entre os membros de grupos de trabalho a partir da perspectiva do capital social e controle comportamental. Eles afirmam que a intenção do empregado em compartilhar conhecimento tácito é afetada pela confiança entre as pessoas. Também asseguram que a intenção de um funcionário em partilhar conhecimento não necessariamente levará a um comportamento tácito de partilhar conhecimentos. Como exemplos eles citam boa parte dos funcionários de Taiwan os quais não são acostumados a partilhar experiências e conhecimento, a menos que tenham um relacionamento de confiança com outros colegas.

Todos esses trabalhos nos permitem perceber uma lacuna na utilização da dimensão tácita do conhecimento associado ao *Sensemaking*. Atualmente, essas atividades são vistas de forma isolada dentro do processo de levantamento de requisitos, sem deixar claro a sua importância na qualidade do *software*. Dada a relevância dessas etapas, utilizar a expertise adquirida no dia-a-dia de tal forma que faça sentido na superação dos obstáculos existentes na fase de levantamento de requisito é o que será visto a seguir na abordagem do *Sensemaking*.

2.4.3 *Sensemaking* e resgate da dimensão tácita do conhecimento

Vários estudiosos ajudaram a compor o conceito de *Sensemaking* e por isso muitas são as formas de grafia, porém optou-se neste trabalho por utilizar a palavra *Sensemaking* escrita desta forma, visto que, também é utilizado pelo grupo de Psicologia Social e Administração Norte-Americano, dos quais se distinguem autores como Daft (1984), Thomas, Clark e Gioia (1993), Gioia e Mehra(1996) e Weick (1995), por exemplo. Mas voltando ao conceito de *Sensemaking* foi abordado aqui Ferreira (1997) que diz o seguinte:

A base conceitual do *Sensemaking* foi desenvolvida com suporte na teoria de vários estudiosos, como Bruner & Piaget (cognição), Kuhn & Habermas (constrangimento das ciências tradicionais e alternativas), Ascroft; Beltran & Rolins (teórica crítica), Jackins & Roger (teoria psicológica) e principalmente em Carter, teórico da comunicação, afirmando que o homem cria ideias para transpor as lacunas que lhes são apresentadas em decorrência da descontinuidade sempre presente na realidade (FERREIRA, 1997, p. 02)

Na visão de Karl Weick (1995), quando se fala de *Sensemaking*, está se falando de algo em andamento, pois a definição do conceito ainda não é consenso no meio acadêmico. Segundo ele o conceito de *Sensemaking* foi definido como o significado literal de construção de sentido. Weick (1995) cita definições de *Sensemaking* apontando outros autores como, por exemplo, Huber e Daft quando dizem que *Sensemaking* é a formação acertada e significativa de eventos vividos por agentes sociais, ou a estruturação do desconhecido, Waterman (*apud* WEICK, 1995).

Entretanto, Weick (1995) mostra que o conceito pode ser baseado em muitas perspectivas, iniciando nas definições, que partem de uma revisão do *Sensemaking* como atividade simplesmente individual até as análises que o colocam como fenômeno grupal.

Já para Evans e Chi (2008), um modelo de *Sensemaking* inclui características sociais. A sugestão dos autores é que este comportamento social seja suportado por ferramentas de tecnologia da informação, utilizando, por exemplo, um método que torne acessível o conhecimento de comunidades e indivíduos de uma determinada rede, auxiliando desta forma os usuários no estágio de preparação de determinada tarefa. Para os autores é interessante que tais ferramentas estejam disponíveis também em uma fase posterior à busca.

Neste sentido, Dervin (1992) interpreta a teoria de *Sensemaking* a partir de ontologias e epistemologias relacionadas às atividades humanas de interpretar, discernir e reparar os acontecimentos ao seu redor, seguindo para o processo de inferência de sentido lógico ao fato observado. É possível verificar que a autora analisa a atividade de *Sensemaking* como sendo um comportamento interno no sentido cognitivo, e também externo relacionado à postura, reações, entre outros comportamentos relacionados com o meio social. É preciso trazer à tona o conceito do sentido cognitivo por se tratar de algo fundamental na ideia de

Sensemaking. Assim, Manis (1973), afirma que existem varias abordagens teóricas dentro do campo das ciências cognitivas. Porém a maioria delas inclina-se para o conceito de que a cognição representa a aquisição de um determinado conhecimento por meio da percepção, reconhecimento, classificação, memória e compreensão, envolvendo para isso operações cognitivas como atenção, memória, raciocínio, pensamento, percepção, etc.

Entretanto a cognição vai além da simples aquisição de conhecimento. Para obter uma melhor adaptação ao meio, a cognição pode ser considerada um mecanismo dentro de nossa mente que capta dados do exterior através dos sentidos e converte para o nosso modo de ser interno através da percepção. Ou seja, utiliza a informação do meio em que vivemos e o que já está registrado na nossa memória.

A memória é segundo Leiderman (2002), um sistema que permite a manutenção provisória e o processamento da informação para elaborar e dirigir nossa conduta. Por isso, Cardoso (2006) afirma que a memória é uma faculdade cognitiva de alta importância, porque forma a base para a aprendizagem, característica que pode ser considerada a habilidade de mudarmos o nosso comportamento através das experiências que estão armazenadas.

Furnas e Russell (2005) dizem que a prática de *Sensemaking* é comum e constante nas atividades do dia-a-dia, quando tentamos entender o que acontece à nossa volta. O processo tem seu início quando nos deparamos com novos problemas e não temos conhecimento suficiente para superá-los. Esta carência de entendimento dispara uma sequência de atividades de busca de informação, indicando que *Sensemaking* pode ser considerado um processo de relações cognitivas e sociais.

Dervin (1992) afirma que, a informação é um minimizador de incertezas, pois ao se confrontar com um obstáculo, o indivíduo busca diversas soluções e formas de conhecimento para suprir a sua necessidade. Assim, são analisadas individualmente as situações, características e ações cognitivas e construtivistas de cada pessoa, na tentativa de se chegar a padrões comuns para a maioria.

Dessa forma é possível analisar com mais clareza a importância do *Sensemaking* nas organizações, mais precisamente em empresas de *software*, as quais tratam o levantamento dos requisitos como um momento fundamental na vida de um projeto. Cabe aos analistas superar os obstáculos citados anteriormente por Dervin (1992), utilizando aspectos cognitivos como, interpretação, discernimento e análise de outros acontecimentos que estão ao seu redor e se inter-relacionam com a busca de conhecimentos, muitas vezes em relações sociais na tentativa de desenvolver representações mais aprimoradas para que auxiliem na execução desta tarefa. Essa ação de *Sensemaking* juntamente com a prática de troca de experiências

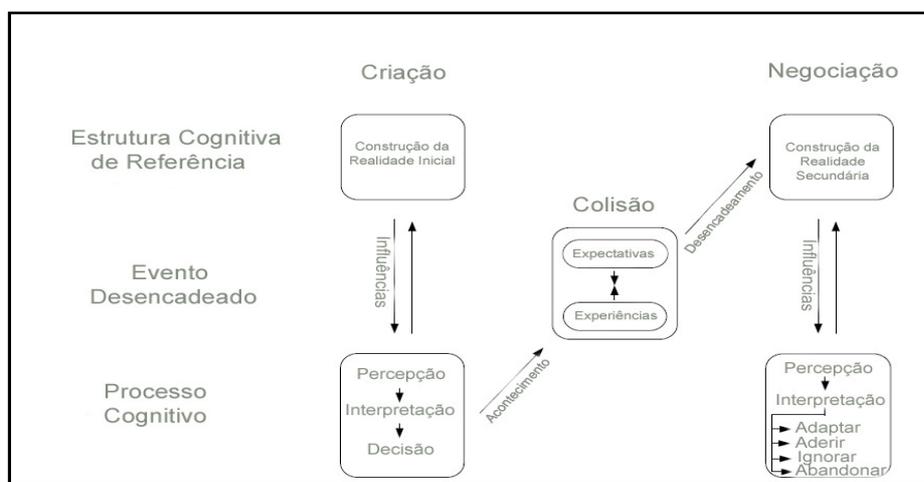
entre analista e cliente, faz com que o processo de inferência feito pelo analista ajude a visualizar mais detalhes relevantes do projeto inclusive novas necessidades de negócio.

Por isso, a partilha de experiências e conhecimentos também conhecida como socialização (NONAKA e TAKEUCHI 1998), se torna fator chave para que se suceda o correto entendimento dos requisitos no contexto do projeto que será desenvolvido.

Nessa linha, merece destaque o estudo de Qiu, Chui e Helander (2007), que teve como foco principal analisar como abordagens cognitivas auxiliam na utilização e gestão do conhecimento para tomada de decisões por equipes virtuais de desenvolvimento de produtos. Os autores indicam que para se obter sucesso na resolução de problemas juntamente com sucesso no desenvolvimento assertivo de produtos que atendam aos requisitos, é fundamental a compreensão dos processos cognitivos das pessoas envolvidas.

Outro estudo, com uma abordagem mais “híbrida” também aporta processos cognitivos, porém com maior ênfase na construção de sentido. Este estudo realizado por Kjærgaard, Kautz, e Nielsen (2007) também foi realizado na área de desenvolvimento de *software*, que pode ser considerada intensiva em conhecimento. O estudo de Kjærgaard, Kautz e Nielsen (2007) teve como foco principal a criação de um manual de gerenciamento de projetos em desenvolvimento de *software*, utilizando como lente analítica o *Sensemaking*. Os autores construíram o modelo apresentado na Figura 8 a partir do mapeamento de processos e desdobramentos de *Sensemaking* e cognição, coletivos e individuais dos envolvidos na pesquisa, a fim de melhorar as práticas de gestão de projetos buscando estimular a partilha de conhecimentos entre os envolvidos:

FIGURA 8 - Fluxo básico dos processos de gestão do conhecimento com processos de *Sensemaking*.



Fonte: Adaptado de Kjærgaard, Kautz, Nielsen (2007).

A camada superior do modelo organizacional mostra dois grupos dominantes, a criação e a negociação. Estas interpretações da realidade conforme Isabella (1990), são realizadas coletivamente e criam a percepção de como se comportar na organização. A interpretação da realidade é um quebra-cabeça de informações e crenças, que juntas formam uma imagem que confirma ou constrói uma realidade (ISABELLA, 1990). Também pode ser visto como uma moldura, um processo no qual as experiências dos atores podem ser utilizadas como *Sensemaking*.

A utilização de experiências no processo de elicitação de requisitos é exatamente o que acredita-se ser o diferencial para que falhas neste processo sejam evitadas e consequentes problemas com a qualidade do *software* ocorram.

A camada inferior do modelo mostra os processos cognitivos construídos na forma organizacional, por parte de seus membros, através de *Sensemaking*. As setas da interpretação da realidade no sentido dos processos cognitivos mostram que os processos individuais são influenciados pelos entendimentos mantidos coletivamente na organização. Os processos cognitivos no primeiro período de criação são baseados no entendimento das pessoas nas partes específicas do fluxo de informações a que estão expostos continuamente.

Intermediando os dois sub-processos de criação e negociação e as duas camadas de processos cognitivos, existe uma espécie de colisão entre as expectativas e experiências. Elas são as percepções e discordâncias dos membros em relação às expectativas e às experiências que geram insegurança e fazem com que essas pessoas encontrem a estabilidade através da criação de um novo quadro em que suas experiências se encaixam.

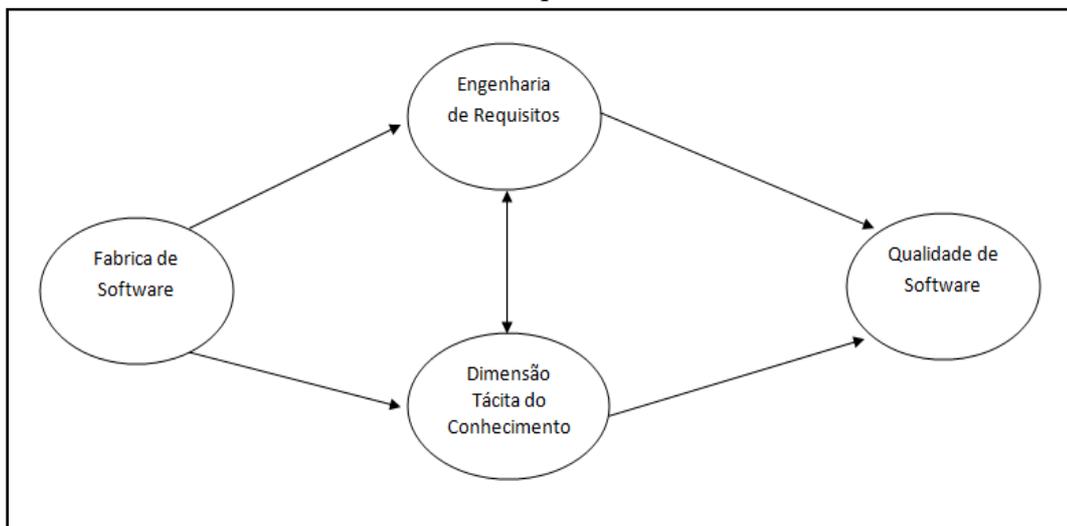
É possível assim fazer uma relação com o levantamento de requisitos de um projeto de *software* onde muitas vezes o analista conhece o funcionamento atual de seu sistema, mas desconhece o detalhe da regra de negócio. Ele tem apenas uma ideia superficial e isso faz com que um levantamento equivocado ou com poucos detalhes e informações seja construído, oferecendo insegurança ao levantamento. Isso pode ser superado com a existência da socialização de conhecimento entre analista e cliente onde as lacunas do desconhecimento podem ser preenchidas pelas experiências de ambas as partes.

O estudo de Kjærgaard, Kautz, Nielsen (2007) deixa claro que a gestão do conhecimento foi inicialmente limitada a focar o processo de externalização do conhecimento dos participantes em procedimentos escritos e, portanto, trouxe uma visão um tanto reduzida de como as formas de conhecimento são importantes no gerenciamento de projetos. Isso também levou a um processo completamente ineficaz de criação do conhecimento que é justamente o que se quer evitar nesse trabalho.

No entanto, segundo os autores quando ocorre à colisão desta abordagem com as experiências em gestão de projetos, os processos de negociação podem revelar uma realidade complexa. Este trabalho busca verificar se estes processos de negociação também ocorrem entre analista e cliente sob a ótica de ajustar divergências existentes na etapa de levantamento de requisitos através da socialização de conhecimento. Além disso, tentar-se-á identificar se houve externalização desse conhecimento, e o mais importante, se houve a conversão do conhecimento tácito do grupo.

Assim, a partir da revisão da literatura estabelece-se um esquema conceitual que estrutura o presente trabalho, apresentado na Figura 9, onde quatro grandes variáveis são relacionadas. O grau de interação entre estas variáveis está ligado diretamente a qualidade final do *software* e por isso ao buscar o estado da arte foi possível perceber uma lacuna na utilização da dimensão tácita do conhecimento que pode ser mais explorada no processo de gerenciamento de requisitos de uma Fábrica de *Software*.

FIGURA 9 - Esquema conceitual



Fonte: Desenvolvido pelo autor.

3 METODOLOGIA DA PESQUISA

3.1 DELINEAMENTO DA PESQUISA

Para classificar a presente pesquisa, considera-se Vergara (2005), que qualifica a pesquisa quanto aos fins e quanto aos meios. No que se refere aos fins, este estudo foi direcionado como uma pesquisa exploratória, do tipo qualitativa e quanto aos meios, a pesquisa foi bibliográfica, de campo e estudo de caso único com o objetivo de responder ao problema em estudo.

No desenvolvimento deste trabalho, optou-se por utilizar como método de pesquisa o estudo de caso por se tratar de uma pesquisa que procura estudar como a dimensão tácita do conhecimento pode contribuir para a melhoria do processo de análise de requisitos, para que as necessidades dos clientes se traduzam em produtos adequados à sua demanda.

3.1.1 Campo de estudo

O campo de estudo desta pesquisa é a empresa Totvs, mais especificamente a franquia de desenvolvimento de soluções para a governança de saúde sediada em Caxias do Sul. Nesta franquia são percebidos conceitos e características que dialogam com as características de uma Fábrica de *Software*. Na unidade de Caxias do Sul é possível perceber aspectos como a existência de processos definidos e estruturados, que buscam o atendimento a múltiplas demandas de natureza e escopo distintas possibilitando flexibilidade aos projetos.

Também é possível evidenciar a não produção em série, dando mais liberdade à criatividade e ao conhecimento prático e teórico das pessoas o que ocorre na produção de bens intensivos em conhecimento. Isso proporciona a produção de *software* com a reutilização de código proporcionando maior qualidade e menor custo.

O início da empresa se dá quando a DZset começa suas atividades em 26 de abril de 1993, quando da terceirização da área de informática da empresa caxiense Fras-le S.A. Seus 17 sócios buscaram reunir esforços técnicos e recursos financeiros para alavancar o surgimento da empresa.

Ao mesmo tempo em que prestava assessoria para a Fras-le, a empresa conquistou um grande cliente: a Unimed Nordeste – RS, para o qual desenvolveu *software* para a gestão de planos de saúde. A partir da consolidação do produto junto à Unimed Nordeste – RS, a DZset expandiu seus serviços para outras empresas na área da saúde, firmando parcerias na

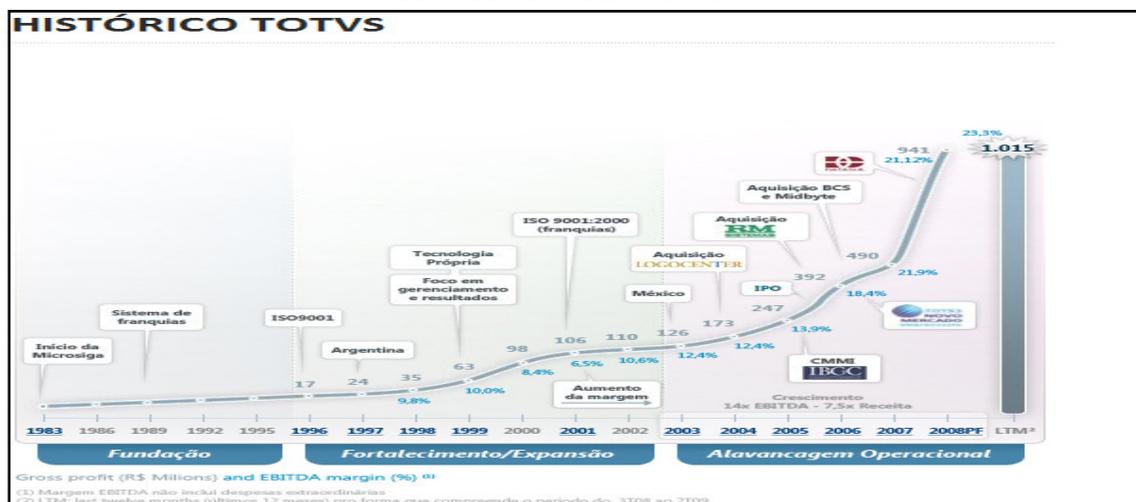
criação de um sistema integrado para a gestão de cooperativas médicas, o SERIOUS – Gestão de Planos de Saúde.

Em 1º de junho de 2005, os ativos da DZset S/A e Datamedical Ltda, foram adquiridos pela Datasul S/A empresa pioneira no desenvolvimento e comercialização de *softwares* de gestão empresarial no Brasil. Após a aquisição, a DZset passou a chamar-se Seventeen, tornando-se então uma franquia de desenvolvimento responsável pela produção de sistemas de gestão para o setor de saúde denominada Datasul Medical. Após uma fusão em novembro de 2007 com a subsidiária Informange de Sorocaba o nome é alterado novamente, desta vez para Datasul Saúde. Em agosto de 2008, a última mudança: a Datasul S/A é adquirida pela Totvs, e passa a fazer parte da maior empresa brasileira de *softwares* de gestão *Enterprise Resource Planning* (ERP), tornando agora a Datasul Saúde uma franquia de desenvolvimento de *software* Totvs- Franquia de desenvolvimento de Soluções para a Governança de Saúde.

O nome Totvs vem do latim e significa tudo, todos, apropriado para uma companhia que fornece soluções em dez segmentos para todos os portes e tipos de empresa. Os produtos são destinados para as áreas: administrativas, sistêmicas, de processos, de desempenho e de infra-estrutura. Toda essa gama de produtos e serviços garantem para a Totvs maior competitividade e, permite cada cliente focar em sua atividade fim e terceirizar parcialmente sua operação administrativa/sistêmica.

Com 26 anos de atuação, a Totvs foi a primeira do setor em toda a América Latina a abrir capital. Tem mais de 25,2 mil clientes ativos, conta com o apoio de nove mil participantes e está presente em 23 países. A seguir é possível verificar na Figura 10 a evolução da empresa.

FIGURA 10 - Histórico Totvs



Fonte: Totvs (2012).

A Totvs mantém como Política da Qualidade: "Qualidade sempre, na busca da superação dos resultados esperados por nossos clientes, participantes, acionistas, fornecedores, parceiros e pela sociedade na qual estamos inseridos". As áreas de gestão de processos, qualidade e qualidade de *software* corporativa, tem como foco a manutenção dessa política, atuando na identificação, unificação, modelagem, padronização, medição, monitoramento e melhoria de processos para conquista de resultados consistentes alinhados com os objetivos estratégicos da empresa (TOTVS, 2012).

Conforme já mencionado no início do trabalho, o estudo foi realizado na Franquia de desenvolvimento de Soluções para a Governança de Saúde (Totvs) localizada na cidade de Caxias do Sul no estado do Rio Grande do Sul a qual possui atualmente 139 funcionários divididos entre matriz e filial e uma carteira composta por aproximadamente 110 clientes responsáveis por um faturamento anual de doze milhões.

O desenvolvimento de novas implementações em produtos existentes é responsável por 44% do faturamento da franquia. Essas demandas sempre partem dos clientes e atualmente chegam aos analistas de sistemas por meio de Ficha de Ocorrência (FO), documento onde ficam descritas as necessidades sob a ótica do cliente. Após análise do profissional de *software*, os requisitos voltam ao cliente para esclarecimentos ou para sua aprovação sob a forma de orçamento. Assim, estudar a Totvs proporcionou dados pertinentes e relevantes para alcance do objetivo desta pesquisa.

3.2 COLETA DE DADOS

Na coleta de dados Yin (2005) afirma que estudos de caso podem receber dados de seis fontes diferentes: documentos, registros em arquivo, observação direta, observação participante, entrevistas e artefatos físicos. A fim de maximizar a validade da construção do estudo, baseados no intento do autor, foram utilizadas nesta pesquisa, dois tipos de fontes de evidências, as quais são complementares: A documentação e as entrevistas.

Ainda segundo Yin (2005), as informações documentais representam uma parte fundamental na coleta de dados, pois corroboram e valorizam as evidências vindas de outras fontes, fornecem detalhes particulares possibilitando também a execução de inferências a partir desses documentos. Dessa forma a busca por documentos relevantes representa um papel notável na coleta de dados.

No que diz respeito às entrevistas foi utilizado o modelo do *Sensemaking* que está centrado no triângulo situação-lacuna-uso. Para usar esta metodologia o pesquisador deve ser capaz de fazer a pergunta certa para ir ao ponto central do assunto.

Esse modelo representa o estado cognitivo do ser humano como um movimento contínuo simbolizando movimento e espaço. Esses momentos do modelo são: a situação, momento inicial onde existe a busca e uso da informação. A lacuna, momento em que ocorre a problemática da tal busca e o uso, momento em que se faz necessário uma estratégia para se alcançar o resultado esperado formando um triângulo Situação-Lacuna-Uso.

Esforços de pesquisas de *Sensemaking* têm criado algumas técnicas de coleta de dados. Dervin (1983) desenvolveu quatro técnicas de entrevistas: *Micro-moment Time-Line Interview*, *Helps/Hunts Interview*, *Close-Ended Sense-Making* e *Message Questioning Interview (MQI)*.

O instrumento para a coleta de dados utilizado nesse trabalho foi a entrevista de *Micro-Moment Time-Line* (entrevista da linha do tempo em micro-momento). Esta técnica de coleta de dados foi desenvolvida ao longo do tempo e é descrita em diferentes níveis de detalhe, com exemplos, em várias publicações (DERVIN, 1983, 1992, 1999). O micro-momento é uma das principais técnicas do *Sensemaking* e a sequencia da entrevista é o ponto central da técnica da abordagem, onde o entrevistador tenta organizar a fala em triângulos (situação/lacuna/uso). Em síntese Dervin explana:

Significa questionar ao entrevistado, detalhadamente, executando um passo a passo do que aconteceu em determinada situação em termos do que aconteceu primeiro, segundo, depois e assim por diante. Na sequencia, em cada etapa, o entrevistado é perguntado sobre quais duvidas ele ou ela teve, quais as coisas que esperava ou necessitava encontrar, o que aprendeu, o que entendeu, se ficou confuso ou mais claro (DERVIN, 1983). (tradução do autor).

O processo básico da entrevista *Micro-Momento Time-Line* significa levantar ao máximo, fatos que aconteceram em uma situação, com a ideia de linha do tempo. Então, para cada etapa chamada *Time-Line*, o entrevistado é questionado sobre o que aconteceu em determinada situação que se queira estudar, seguindo com perguntas sobre os questionamentos que ele possuía, as coisas que ele precisava descobrir, aprender, se ficou mais confuso ou mais elucidado e se o *Sensemaking* foi necessário (DERVIN, 1983).

Então, para cada uma destas perguntas ou lacunas, o entrevistado é questionado de forma que suas respostas mostrem algo sobre seu mundo, possibilitando fazer conexões entre experiências passadas e presentes (DERVIN, 2001), permitindo a compreensão dos três vértices do triângulo do *Sensemaking*: situação-lacuna-uso.

Essa entrevista é formada por três passos básicos que precisam estar sob o controle do entrevistador. Além disso, a metodologia do *Sensemaking* recomenda que para entrevistas em grupo e individuais sejam formuladas questões simples e universais semelhantes ao Quadro 1 mostrado abaixo:

QUADRO 1 - Etapas e questionamentos

Passos Questionamentos	1º narração de eventos ordenados no tempo ou sem ordenar. (Dervin, 1983)	2º expressão de confusões, ideias, emoções, sentimentos, perguntas e conclusões.(Dervin, 1983)	3º Respostas as perguntas (Dervin, 1983)
Dervin 1983 Dworkin et al., 1999	O que aconteceu? O que você pensou? Sentiu? Concluiu sobre o que aconteceu? (Dworkin et al., 1999)	O que te confundiu? Como esses pensamentos, sentimentos, conclusões se relacionam com sua vida até aqui? (Dworkin et al., 1999)	O que ajudou você nesse momento? O que levou você a essa conclusão? Quais conexões você percebe? Como você quer ser ajudado agora? (Dervin, 1983) O que espera para adiante? O que empurra ou limita seu mundo? (Dworkin et al., 1999).
Autor	Existia um problema? Como iniciou o processo de levantamento de requisitos? Houve contato com o cliente? De que forma foi? Você julga importante esta atividade? Qual o próximo passo?	Existiram dúvidas na elicitação dos requisitos? Tudo o que o cliente solicitou é de seu conhecimento? Você precisava aprender algo mais? O cliente sabia exatamente o que queria? Se sabia, soube explicar? Existe uma forma viável de implementar a demanda? Você já fez algo parecido? Qual o próximo passo?	O que foi feito no momento de confusão ou dúvida? Porque você conduziu por este caminho? O que foi feito para elucidar os requisitos? Houve algum contato com o cliente? Sua experiência ajudou na definição dos requisitos? O cliente o ajudou em algum momento? Como? O que foi feito por último?

Fonte: Desenvolvido pelo autor.

As entrevistas foram realizadas com doze pessoas que ocupam cargos de analistas de sistemas, os quais estão divididos em analistas júnior, pleno e sênior responsáveis por fazer o levantamento de requisitos de *software* normalmente sob a forma de orçamentos. Estas 12 pessoas representam 100% da amostra, ou seja, todos os analistas responsáveis pelo levantamento de requisitos na área de desenvolvimento de novas implementações em produtos existentes.

Por mais que existam diversas técnicas, todos os métodos de coleta de dados têm algumas características em comum como, por exemplo, uso de estruturas livres de roteiro de entrevistas, estrutura de audiência baseada na compreensão do outro e a construção (ou não) de sentido (DERVIN, 1983).

Dervin (1999) escreve que o método de pesquisa como *Sensemaking* oferece uma teoria da entrevista e não um *script*. A aplicação pode ter inúmeras formas, dependendo da finalidade do estudo. A técnica de entrevista Micro-Momento *Time-Line* entra em detalhes sobre um passo da linha do tempo (Dervin, 1992), normalmente escolhido pelo pesquisado com base em vários critérios, por exemplo, mais importante, mais recente, mais problemático, mais lembrado. Dessa forma a entrevista *Time-Line* permite obter dados sobre situações memoráveis ou mais significativas. Uma vez realizadas as entrevistas, os diálogos serão registrados para a análise.

Como já dito anteriormente este estudo é qualitativo e tem seu foco nos analistas de uma Fábrica de *Software*, porém entende-se que destinar um olhar para o cliente é extremamente importante, realizando dessa forma uma verificação da percepção do cliente quanto à qualidade dos produtos finais cujos requisitos foram definidos junto aos analistas entrevistados. Para tanto foi construído um questionário a partir da norma NBR ISO/IEC 9126-1:2003 referenciada no capítulo 2.5 deste trabalho, onde a mesma foi adaptada para que contivesse apenas questões fechadas.

Em geral o questionário refere-se a uma forma de alcançar respostas às questões por uma fórmula que o próprio informante preenche. Assim, os questionários foram enviados a 12 respondentes que representam 100% da amostra e que ocupam a função de analistas de negócios dos clientes para que respondessem acerca da qualidade do projeto entregue pelo analista da Totvs previamente entrevistado. Os analistas e projetos foram identificados nos questionários apenas para fins comparativos recebendo nomes fictícios ao longo do trabalho.

Este instrumento foi aplicado através da internet onde um link encaminhado por email aos respondentes remetia a uma página web a qual gravava as informações. Tal instrumento encontra-se no Apêndice C.

3.3 TÉCNICAS E CATEGORIAS DE ANÁLISE

Uma vez coletados os dados, estes foram analisados segundo as categorias derivadas do estudo de Kjærgaard, Kautz, Nielsen (2007) que referem-se à Criação, Colisão e

Negociação. O trabalho foi referenciado na seção 2.4.3 e propôs uma concepção de como é o significado a partir da partilha de conhecimentos em um grupo de gerentes de projeto em uma empresa dinamarquesa de *software*. O estudo realizado produziu uma visão de como utilizar o *Sensemaking* no gerenciamento de projetos com base na interpretação de suas próprias realidades e por isso é adotado neste estudo como modelo de análise.

A criação é a fase inicial, onde existe a busca pela informação, por algo que não está condizente com a realidade das organizações ou simplesmente melhore os processos operacionais das mesmas. É nesse momento em que a necessidade do cliente é levada até a empresa de *software* para que o processo de desenvolvimento da demanda seja realizado, iniciando pelo levantamento de requisitos feito por um analista.

A colisão é o passo seguinte, momento em que pode existir algum tipo de confusão, ou seja, algum tipo de problema na elicitação dos requisitos, com a possibilidade de ocorrência de colisão entre expectativas criadas pelo cliente e experiências de projetos anteriores dos analistas.

Por fim, verificou-se se ocorreu a negociação, processo que pode acontecer entre cliente e analista na transformação do conhecimento tácito através da socialização, afim de, deixar os requisitos mais claros para que se alcance um produto aderente a necessidade do cliente gerando dessa forma um novo quadro.

Para viabilizar a análise, as respostas foram organizadas com o uso do Mapa de Associação de Ideias associado ao *software* Atlas Ti. Mapas de associação de ideias são instrumentos de visualização cujo objetivo é subsidiar o processo de análise e interpretação dos dados da pesquisa, a fim de facilitar a comunicação de seus resultados (VERGARA, 2005).

Segundo a autora o mapa de associação de ideias confere visibilidade ao processo de análise por meio da organização de dados em estado bruto, em colunas que correspondem a categorias definidas pelo pesquisador. Outra característica é que ele permite a apresentação dos dados na sequência em que foram coletados, sem fragmentá-los, evidenciando o todo ao se percorrermos as colunas do mapa (VERGARA, 2005).

Para a utilização do mapa de associação de ideias utilizou-se inicialmente o *software* Atlas Ti para organizar as respostas segundo as categorias propostas. Foram realizadas 12 entrevistas que representaram sete horas de conversas, algo que ao ser transcrito gerou aproximadamente 96 páginas de texto.

Foi verificado entre as diversas possibilidades de uso do computador no processo de organização e tratamento dos dados coletados e foi definido que o *software* Atlas/Ti poderia

permitir gerenciar a complexidade dos dados qualitativos além de sistematizar, acelerar e facilitar a interpretação dos mesmos.

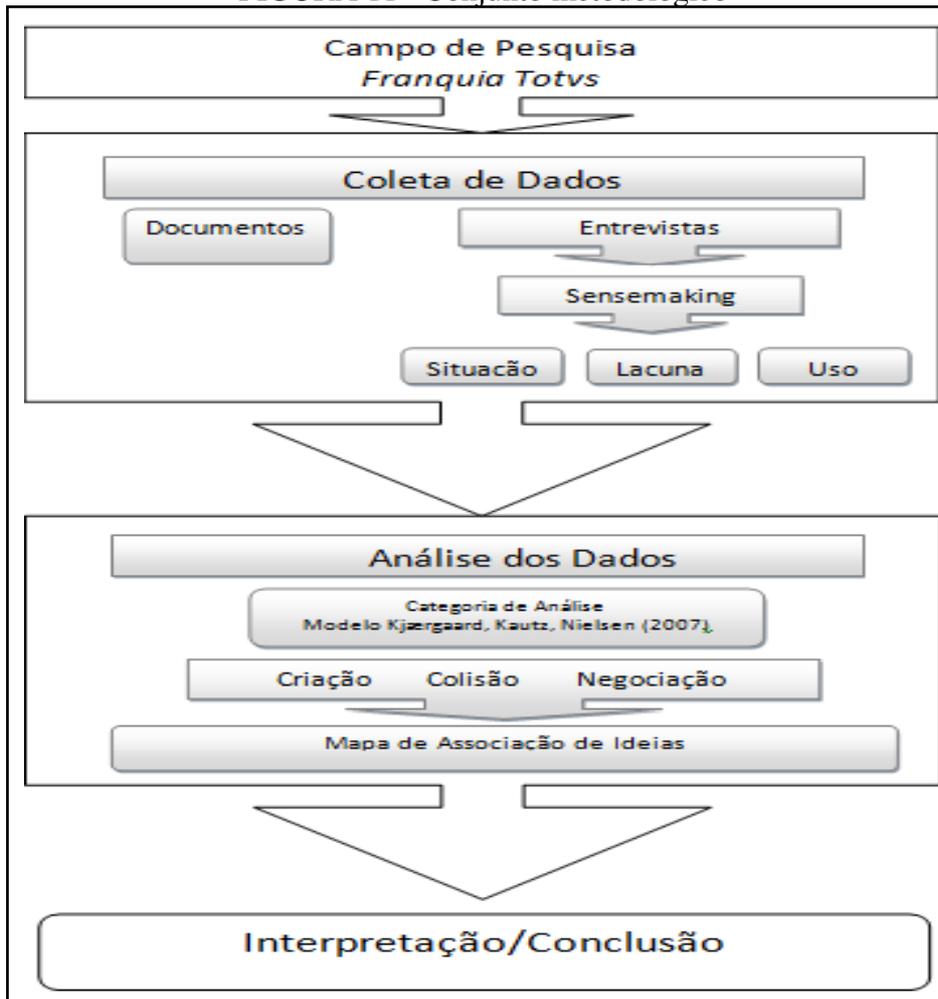
O Atlas/Ti consiste em um *software* de análise de dados qualitativos *Computer-Assisted Qualitative Data Analysis Software* (CAQDAS). Foi desenvolvido na Universidade Técnica de Berlin, Alemanha, e a sigla “Atlas” tem um significado em alemão que pode ser traduzido como “arquivo para tecnologia, o mundo e a linguagem cotidiana”. Já a sigla “ti” advém de “Text Interpretation”, ou seja, interpretação de texto (BANDEIRA-DE-MELLO, 2006).

O objetivo do *software* não é automatizar o processo de análise, mas desenvolver uma ferramenta que apoie e facilite a interpretação humana. A utilização do *software* Atlas/ti apresenta algumas vantagens como o fato de ser flexível, podendo ser ajustado conforme os dados, escopo e estratégia da pesquisa.

O Atlas/Ti apresenta quatro princípios básicos: Visualização que gerencia a complexidade do processo de análise, a Integração que reúne todos os dados relevantes em um único projeto, a Casualidade que analisa de forma intuitiva os dados, encontrando respostas, achados inovadores e por fim a Exploração que examina o percurso de interpretação através dos elementos constitutivos do programa.

Tanto o Atlas/Ti como o mapa de associação de ideias foram estruturados a partir das três categorias provenientes do estudo de Kjærgaard, Kautz e Nielsen (2007) que referem-se à criação, colisão e negociação, tornando-se nosso filtro interpretativo além de determinar a quantidade de colunas do mapa. Colunas essas que receberam o conteúdo das entrevistas identificadas pelo Atlas/Ti de acordo com as categorias que se relacionam. Em seguida ocorreu a interpretação dos dados, resgatando-se o problema que provocou a investigação possibilitando identificar, por exemplo, quais categorias predominam. No modelo abaixo pode-se visualizar como ficou estruturado o conjunto metodológico:

FIGURA 11 - Conjunto metodológico



Fonte: Desenvolvido pelo autor.

4 RESULTADOS

As entrevistas semiestruturadas foram realizadas a partir do roteiro contido no item 3.3 deste trabalho. O material coletado foi gravado e transcrito, com a autorização dos entrevistados e, conforme assegurado a eles, as identidades foram mantidas em sigilo.

4.1 CARACTERÍSTICAS DOS ENTREVISTADOS

Foram entrevistados 12 funcionários da franquia de desenvolvimento Totvs representando 100% dos profissionais da área de desenvolvimento de novas implementações em produtos existentes, os quais têm suas características observadas no Quadro 2:

QUADRO 2 - Características dos profissionais entrevistados

Entrevistado	Cargo	Nível	Tempo na empresa
Analista A	Analista de Sistemas	Sênior	Quase 10 anos
Analista B	Analista de Sistemas	Junior	Menos de 1 ano
Analista C	Analista de Sistemas	Pleno	Entre 5 e 10 anos
Analista D	Analista de Sistemas	Junior	Menos de 1 ano
Analista E	Analista de Sistemas	Sênior	Mais de 10 anos
Analista F	Analista de Sistemas	Pleno	Entre 5 e 10 anos
Analista G	Analista de Sistemas	Junior	Entre 1 e 5 anos
Analista H	Analista de Sistemas	Sênior	Mais de 10 anos
Analista I	Analista de Sistemas	Junior	Entre 1 e 5 anos
Analista J	Analista de Sistemas	Pleno	Entre 1 e 5 anos
Analista K	Analista de Sistemas	Junior	Entre 1 e 5 anos
Analista L	Analista de Sistemas	Pleno	Entre 1 e 5 anos

Fonte: Desenvolvido pelo autor.

Descritas as características dos entrevistados e expondo a forma pela qual foi composta a amostra em uma empresa possuidora de características de Fábrica de *Software*, passa-se à explicação de como foi conduzida a análise de conteúdo das entrevistas.

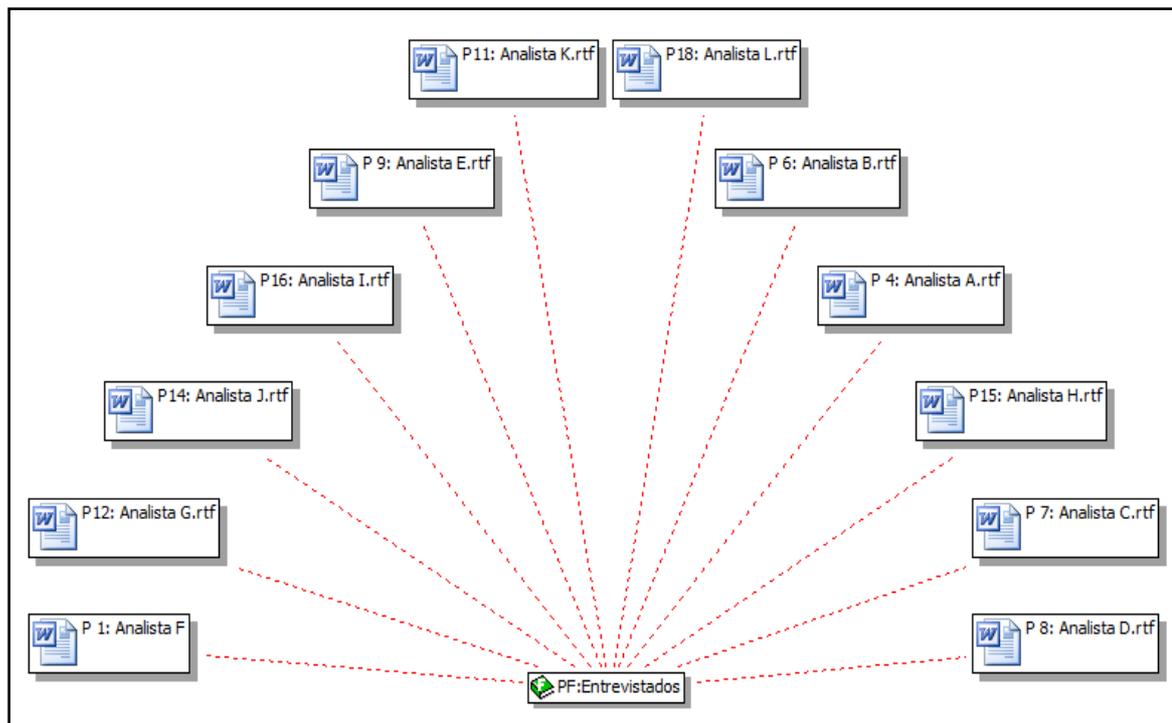
4.2 COLETA E ORGANIZAÇÃO DOS DADOS

Utilizando-se da técnica de *Sensemaking* Micro-Momento *Time-Line* proposta por Dervin (1983), buscou-se saber nas entrevistas o que aconteceu no evento lembrado pelos respondentes e o que os levou a proceder de determinada forma, principalmente nas dificuldades encontradas no dia-a-dia onde somos instigados a ter novas ideias para transpor as barreiras criando assim uma nova realidade (Furnas e Russell, 2005; Ferreira, 1997). Baseado nesta perspectiva, o sistema de análise dos dados decorreu por meio de três passos.

Inicialmente o primeiro passo foi a escuta e transcrição exata de todas as entrevistas realizadas.

O segundo passo consistiu em utilizar o *software* Atlas/Ti para analisar as entrevistas já transcritas. Inicialmente estas entrevistas foram importadas para dentro do *software* criando um projeto único (Unidade Hermenêutica) como pode-se verificar na Figura 12:

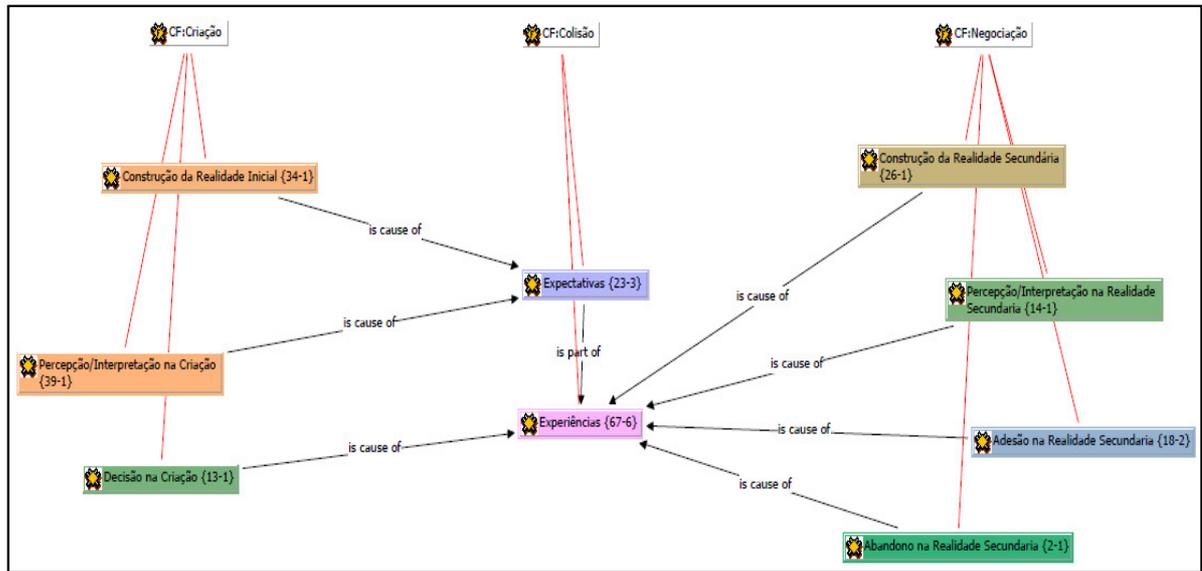
FIGURA 12 - Unidade hermenêutica - entrevistados



Fonte: Desenvolvido pelo autor.

Com as entrevistas disponíveis foi necessário criar as categorias, também chamadas de famílias, de acordo com o artigo dos autores Kjærgaard, Kautz e Nielsen (2007). Pode-se perceber na Figura 13 a criação das categorias e suas subcategorias de tal forma que se assemelhassem ao modelo proposto pelos autores para que pudessem ser utilizadas para codificação:

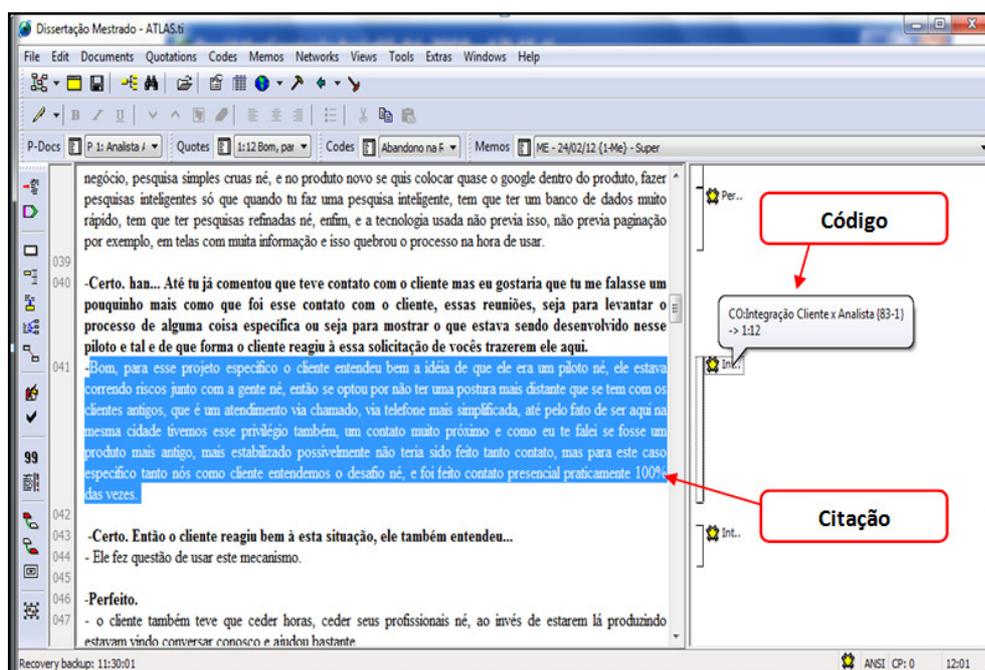
FIGURA 13 - Categorias e suas sub-categorias



Fonte: Desenvolvido pelo autor.

A partir daí iniciou-se a codificação, processo que identifica nas falas, uma representação do conteúdo capaz de evidenciar para o pesquisador características presentes no material analisado. Nesse momento uma nova leitura das entrevistas se fez necessária para que cada pedaço de frase fosse encaixado em sua respectiva categoria proporcionando um melhor entendimento das respostas no contexto do trabalho. Um exemplo de codificação pode ser observado na Figura 14:

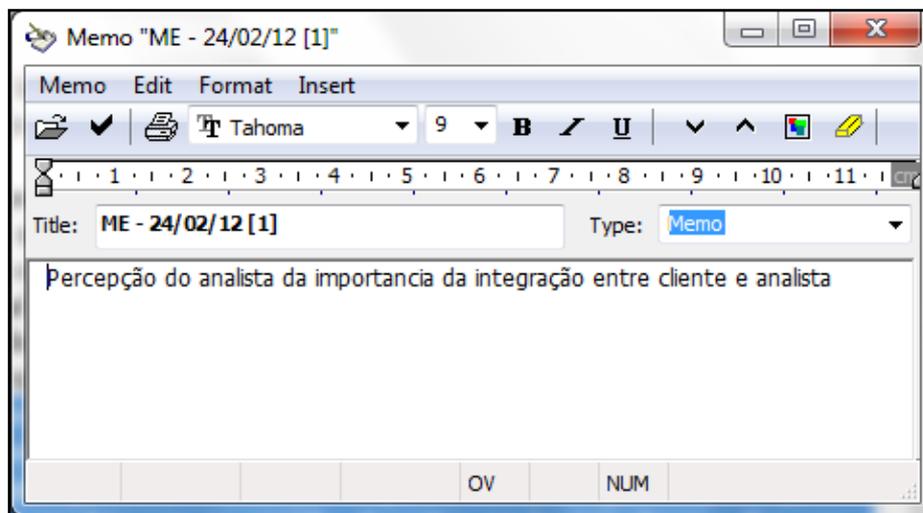
FIGURA 14 - Exemplo de codificação



Fonte: Desenvolvido pelo autor.

Na Figura 14 percebe-se que é atribuída uma codificação para trechos das falas dos entrevistados, possibilitando rever posteriormente todas as falas de todos os entrevistados de maneira fácil e unificada. No decorrer de todo o processo, foram criados comentários para as entrevistas, sempre que se julgava relevante. Essa ação foi tomada com a finalidade de registrar informações importantes para consulta futura, pois um comentário, uma síntese de resultados podem ser essenciais para o pesquisador no andamento das análises como pode ser observado na Figura 15.

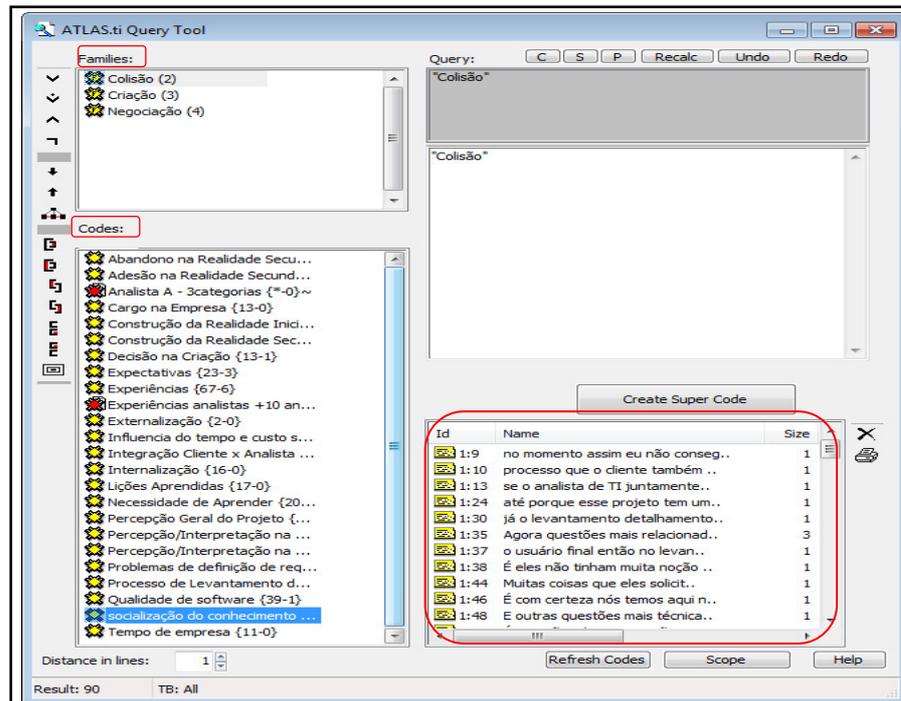
FIGURA 15 - Exemplo comentários



Fonte: Desenvolvido pelo autor.

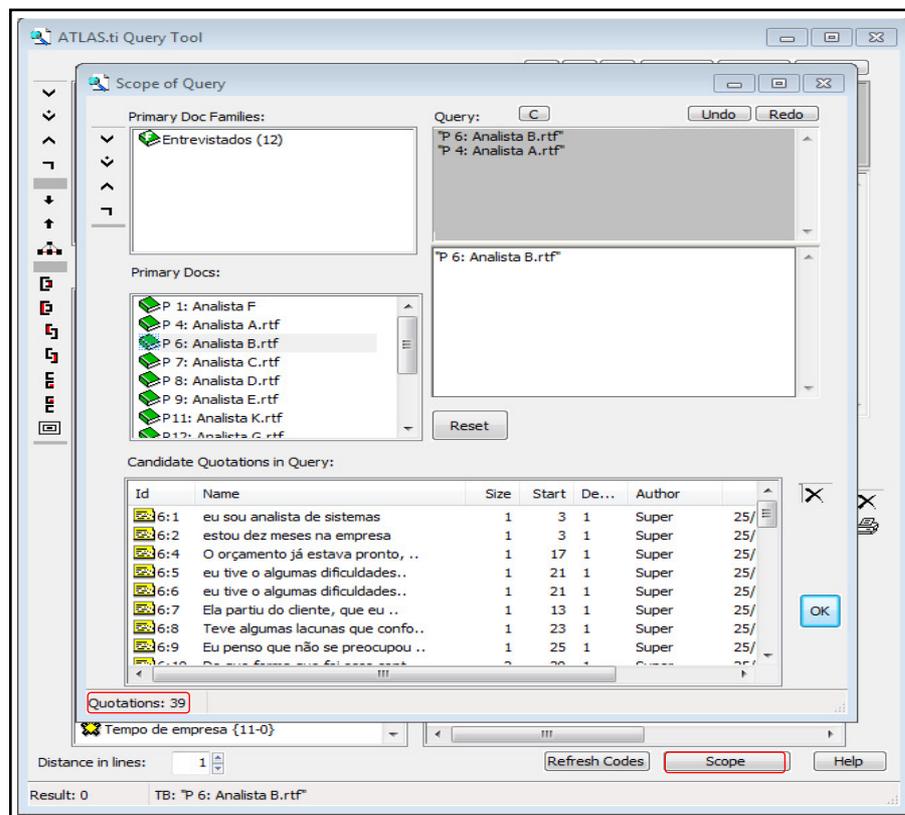
Como dito anteriormente o Atlas/Ti age como uma ferramenta que facilita e agiliza a identificação dos conceitos teóricos nos documentos coletados na pesquisa, tornando mais fácil e ágil a visualização dos achados como pode-se verificar nas figuras 16 e 17. O *software* possui uma ferramenta de consulta a qual permite que combinações sejam realizadas rapidamente, por exemplo, listando todas as codificações de determinada família, todas as falas de determinadas categorias ou ainda todas as famílias de determinada entrevista.

FIGURA 16 - Ferramenta de consulta 1



Fonte: Desenvolvido pelo autor.

FIGURA 17 - Ferramenta de consulta 2

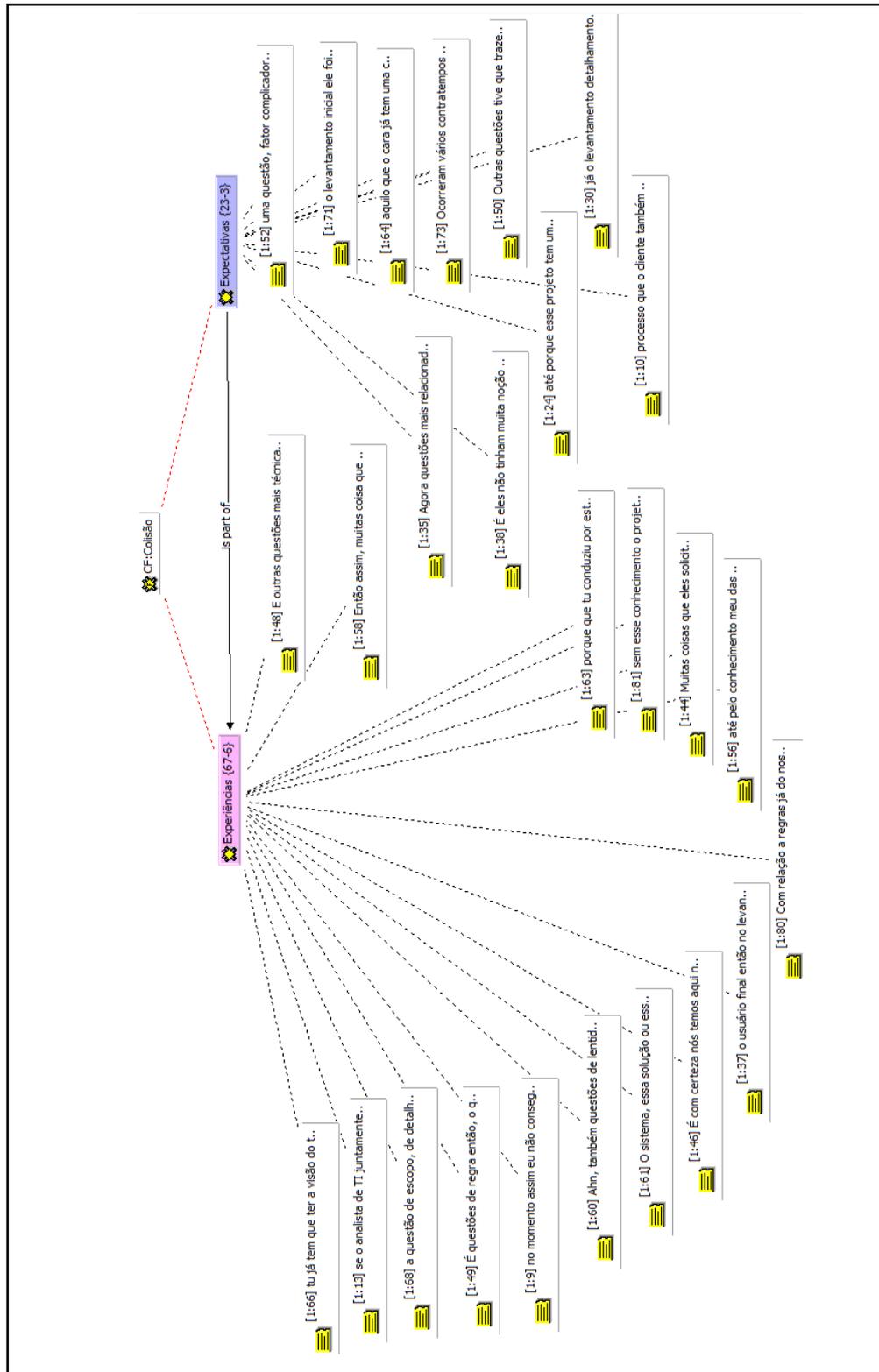


Fonte: Desenvolvido pelo autor.

O Atlas/Ti também possibilita gerar esquemas gráficos (networks) que podem ser visualizações de um elemento chave e suas conexões com a teoria. Na Figura 18 é possível

observar todos os trechos da entrevista de um determinado analista. Nesta visualização percebe-se a família “Colisão” referenciada na teoria com suas sub-categorias, as experiências e expectativas, todas relacionadas com a entrevista do analista F neste caso.

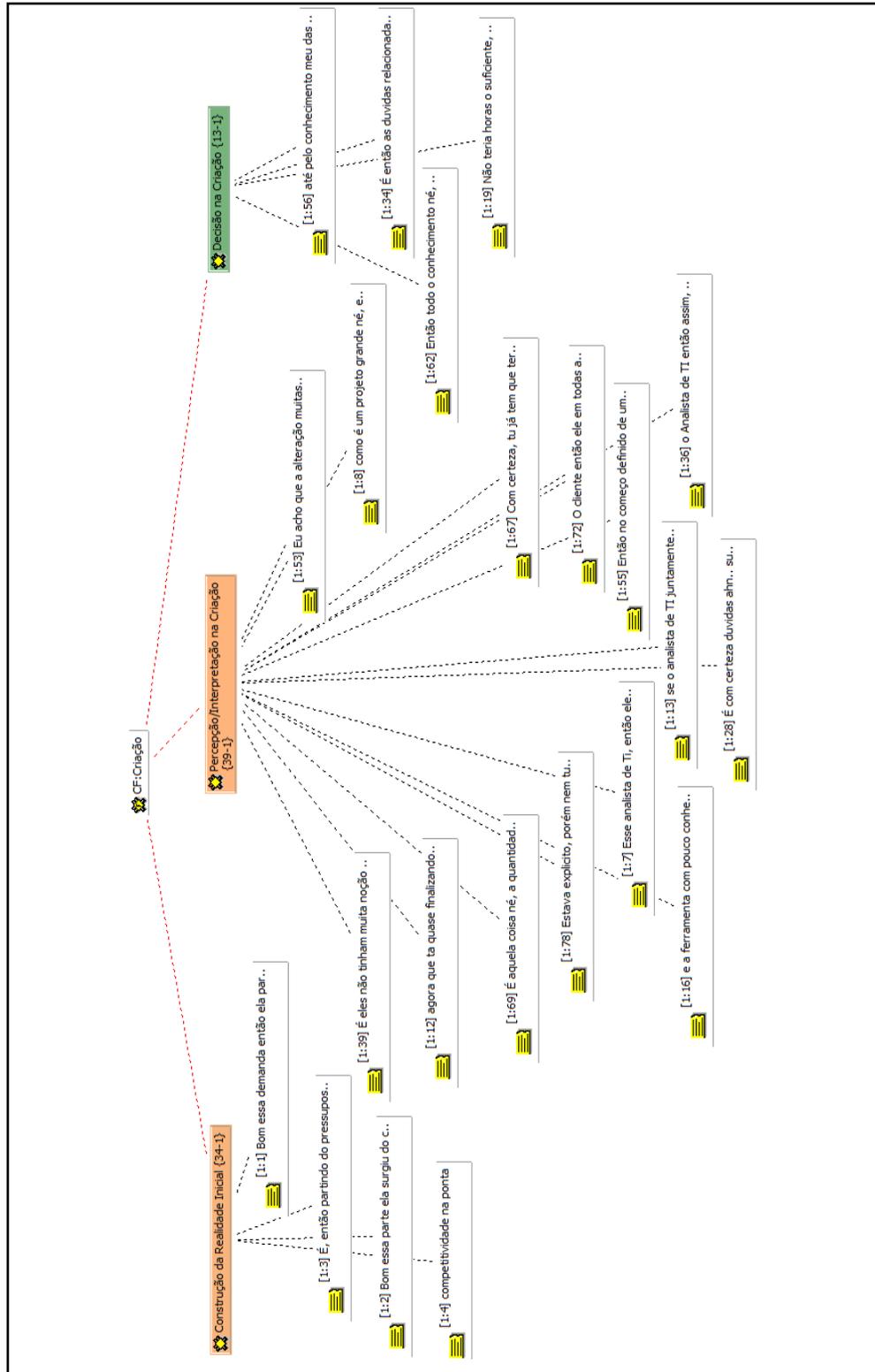
FIGURA 18 - Família colisão



Fonte: Desenvolvido pelo autor.

Da mesma forma acontece com a família “Criação”, outro exemplo destes esquemas gráficos que facilitam a visualização e dão uma ideia mais clara da quantidade de vezes que suas sub-categorias são identificadas nas falas dos entrevistados.

FIGURA 19 - Família criação



Fonte: Desenvolvido pelo autor.

Assim, foi possível viabilizar a utilização dos mapas de associação de ideias, pois com as entrevistas devidamente classificadas e os filtros criados a fase seguinte foi a transferência das falas para dentro do mapa, ou seja, retomando o sistema de análise dos dados a etapa seguinte realizada foi a elaboração do mapa de associação de ideias com a inclusão dos trechos das entrevistas de forma sequencial separados em categorias. O teor das transcrições foi organizado com o caráter dialógico e o sentido das entrevistas preservado. Ao final de cada resposta dos entrevistados foi incluída uma abreviação como, por exemplo, “p01” onde “p” significa a pergunta e “01” significa o número da mesma. Estas perguntas fazem parte do roteiro utilizado nas entrevistas com os analistas e estão contidas no Apêndice A deste trabalho.

Uma nova tabela então foi estabelecida com a quantidade de colunas de acordo com o que foi definido no tópico 3.3 deste trabalho, categorias de análise, ou seja, criação, colisão e negociação (KJÆRGAARD, KAUTZ e NIELSEN, 2007).

5 ANÁLISE E DISCUSSÃO

Iniciando o processo de análise dos resultados verificou-se no Mapa 1- Criação, disponível no apêndice B, as diversas formas de implementações feitas pelos analistas: implementações, melhorias em funcionalidades existentes, continuações de projetos que já estavam em andamento e correções no produto existente.

Nessa realidade identifica-se que analistas com menos tempo de empresa encontraram dificuldades na definição dos requisitos em virtude da falta de conhecimento da regra de negócio do sistema, fato esse que não se observa nos discursos de analistas mais experientes. Essa dificuldade acaba se refletindo no projeto, pois segundo os próprios analistas pode acabar trazendo certo constrangimento e com isso dificultando o contato com o cliente na busca de refinar os requisitos. Como se pode ver na declaração abaixo esse tipo de problema passa pela convivência da coordenação indicando uma falha de processo interno, visto que, uma continuação de um projeto não pode ser realizada por outro analista quando o mesmo está iniciando sua carreira na organização.

“[...] eu tive algumas dificuldades para entender, porque como ele era uma continuidade faltou um pouco de entendimento meu no início.”

“[...] então eu acho que nisso assim eu comecei a cuidar mais o que eu perguntava, ir mais atrás da pessoa que fez o orçamento e não do cliente, porque senão eu ia acabar estressando o cliente porque ele deve ter sentido que...*como botaram uma outra pessoa para fazer, sendo que eu já tinha explicado tudo para aquela pessoa?*”

Pesquisador: “Você em algum momento se sentiu constrangido em estar perguntando isso?”

Entrevistado: “É, porque talvez não caberia a mim fazer esse projeto.”

Em outras oportunidades verifica-se que o cliente facilita o trabalho dos analistas por diversas vezes quando cria documentos e fluxos tornando o processo mais visual sempre no intuito de auxiliar a fase de definição dos requisitos para que ele mesmo receba um produto com maior qualidade. Esse processo faz parte da construção da realidade inicial quando o cliente utiliza inclusive operações cognitivas transformando suas percepções do dia-a-dia, nesse caso as suas dificuldades, em uma nova demanda indo de encontro ao que diz Isabella (1990). Essa etapa também é conhecida dentro da perspectiva do *Sensemaking* como o primeiro vértice do triângulo, a situação.

No Mapa 2 – Colisão, que se encontra no apêndice B é possível observar que em praticamente todas as entrevistas os analistas se utilizam de experiências e de suas próprias interpretações da realidade para tornar um requisito de *software* ou até o desenvolvimento do

projeto o mais eficiente possível. Essa interpretação da realidade seria uma espécie de quebra-cabeça o qual precisa ser montado pelas percepções coletivas de como se comportar na organização (KJÆRGAARD, KAUTZ e NIELSEN 2007) utilizando normas, técnicas e processos juntamente com informações e crenças do seu dia-a-dia (ISABELLA, 1990).

Através destas percepções do próprio cliente são criadas as expectativas em torno do projeto, visto que, em parte das entrevistas é possível identificar que os analistas percebem esse sentimento quando da existência de integração entre eles. O cliente passa de uma forma tácita tudo aquilo que não conseguiu expressar no papel, principalmente detalhes da usabilidade, possibilitando ao analista recolher mais informações sobre a solicitação. O trecho abaixo retirado da fala de alguns analistas confirma o que foi dito acima.

“Teve, uma situação em que foi pedido um campo no cadastro, só que este campo não ficou muito bem definido então eu entendi uma coisa, mas depois não era bem isso, e o usuário não sabia muito bem o que era, chegamos a ter de conversar por telefone, ligamos para a área solicitante e aí se definiu o que era[...]”

“Então teve questões que ele solicitou que digamos para tu alterar o registro no sistema, tu tinha que dar 40 "enters" para chegar até na informação que tu queria alterar[...]”

“[...]eles pediram para lá no botão da consulta ter uma nova função que quando eu clicasse lá já fosse direto para aquela informação.”

No mapa 2 um fato curioso chamou a atenção, pois o cliente em determinado projeto solicitou que o levantamento de requisito fosse feito por um analista experiente. Isso mostra que implicitamente o cliente entende que a integração cliente analista é necessária, algo que ocorre principalmente com profissionais com esse nível de conhecimento os quais terão mais bagagem para poder interagir, questionar e discutir com o cliente.

“[...]O cliente, ele entendia que era um projeto complexo então ele solicitou alguém que tivesse experiência no sistema para estar acompanhando o próprio levantamento dos requisitos.”

Dando sequencia nesse raciocínio, analistas que provocam a colisão segundo o modelo de Kjærgaard, Kautz e Nielsen (2007) na tentativa de definir da melhor forma os requisitos acabam encontrando alguns problemas. Problemas estes que muitas vezes são criados pelos próprios clientes como no trecho abaixo, onde o cliente se esquece de mencionar um processo interno seu. Essa identificação só foi possível através da interação existente entre cliente e analista e devido a proximidade entre os dois também pode ser sanada de forma mais rápida.

“[...]na hora que eles colocaram para nós o projeto eles não lembravam que tinham aqueles três ou quatro casos lá que era uma outra situação.[...]”

“[...]Todas as dúvidas eram sanadas de imediato.”

Na análise do Mapa 3 Negociação, também disponível no apêndice B, pode-se perceber que os processos de negociação são desencadeados em sua grande maioria pela colisão das expectativas normalmente dos clientes com as experiências dos analistas. Isso é decorrência da busca incessante em sempre melhorar ao máximo um requisito.

Porém, existem aquelas situações onde a interpretação também pode gerar a negociação. Nesses casos isso ocorre pela forma com que foram escritos os requisitos, tanto pelo cliente como pelo analista ocasionando discussões devido a sua dupla interpretação.

A fase da construção da realidade secundária ocorre após a negociação entre cliente e analista. É possível identificar essa fase na compilação de trechos das falas abaixo onde os analistas procuram passar para o cliente como o sistema deve atuar mesmo que em alguns casos se necessite de outros caminhos para chegar ao mesmo resultado. Isso deixa claro a existência da preocupação por parte dos analistas em não fugir da solicitação inicial. As situações em que se obriga a execução de um processo de determinada forma são menos comuns, mas também fica claro que elas existem.

“[...]e aí se buscou caminhos alternativos que alguns contentaram o cliente e outros foi meio que empurrado, ó é assim ou não tem outro jeito e aí começou um pouco de desgaste em alguns momentos.”

“[...]Se tentou, se dialogou no início, mas nós acabamos meio que impondo uma solução porque foi o que tecnicamente era viável e ficou dentro...”

“Neste caso a negociação foi tranquila. Sempre me preocupo em deixar claro para o cliente que se o que o ele quer é viável para o sistema, é aceito exatamente como ele coloca. Caso não seja viável, é exposto para o cliente e oferecido outra forma pra atendê-lo. Tudo é negociável para que fique bom para ambos.”

Outras fases do modelo de Kjærgaard, Kautz e Nielsen (2007) também são identificadas nas entrevistas. A interpretação e percepção da análise secundária são processos cognitivos presentes no dia-a-dia destes profissionais e que auxiliam nas suas tomadas de decisão. Quando se recorre às experiências para solucionar um problema ou indefinição no processo de definição dos requisitos ocorre a adesão dessa nova realidade criada. Entretanto uma fase pouco vista nas conversas é o abandono da realidade secundária, exatamente o oposto da adesão, quando as experiências são ignoradas e o processo sugerido é aceito.

“[...]mas para não inviabilizar o projeto agente construiu uma outra um pouco fora do padrão para atender aquela necessidade do cliente[...].”

Isso pode ser percebido em apenas uma entrevista na qual o analista confessa ter uma visão bastante técnica, acatando a solicitação do cliente tornando o sistema mais usual.

“Na realidade eu ia manter dessa forma, mas eles pediram para lá no botão da consulta ter uma nova função que quando eu clicasse lá já fosse direto para aquela informação[...].”

Saindo um pouco das categorias de análise e dos mapas de associação de ideias, outras revelações são percebidas nas entrevistas. O processo de definição de requisitos registrado pelas entrevistas indica que analistas com mais de um ano de empresa buscam o diálogo com o cliente como uma forma de entender melhor a sua solicitação criando um requisito mais adequado e tornando seu trabalho mais eficiente e assertivo. Isso acontece devido à experiência adquirida nesse período a qual perpassa maior segurança ao trabalho desempenhado.

Quando perguntado para estes respondentes que pensassem em um projeto que os marcou, foi possível observar que analistas com mais tempo de empresa falaram de projetos onde ocorreu essa interação. Identifica-se também que a maioria desses projetos na visão dos analistas ficaram de acordo com a especificação do requisito atendendo a necessidade do cliente.

Alguns problemas na fase de definição dos requisitos citados por Brooks (1987), foram visualizados nas entrevistas no decorrer deste trabalho. O mais evidente de todos é o fato de não existir nenhum tipo de metodologia ou padrão para realizar tal processo. Independentemente de cliente, módulo e tamanho do projeto cada analista experiente ou não utiliza apenas seu conhecimento para realizar essa etapa tão crítica em um projeto de *software*. Alguns dos problemas mais comuns são destacados abaixo nas próprias declarações dos entrevistados.

“eu tive algumas dificuldades para entender porque como ele era uma continuidade, faltou um pouco de entendimento meu no início... eu tive que buscar com a pessoa que fez para mim poder fazer, senão eu não conseguiria. Eu lendo, assim uma pessoa de fora que pega o projeto fosse ler para entender não iria entender por que ele é uma continuidade.”

“100% não, porque no decorrer do desenvolvimento a gente teve que realmente adequar algumas coisas que como a gente não tinha analisado no total ou até o cliente não sabia que ele tinha aquele processo em funcionamento.”

“Teve algumas contradições pequenas no final alí com a definição inicial que havia sido aprovada e com a... e depois na hora de homologar o cliente já..opa! isso aqui não... mas tu pediu daquela forma. Então teve algumas contradições também a nível de usuário final”

Leite (1994) afirma que tais problemas ocorrem por dois principais motivos. Ou pela falta de conhecimento das técnicas por parte de analistas ou pelo excesso de intimidade com o cliente. Fica claro que por mais que os projetos tenham sido entregues de acordo com os requisitos na opinião dos próprios analistas, em todos eles algum tipo de problema na etapa de definição ocorreu. Tais problemas ocorreram pelo primeiro motivo confirmando assim o que diz o autor.

Uma explicação para isso pode ser encontrada na afirmação de Pressman (2006), onde o autor relata que o pouco tempo investido para se verificar o que realmente foi registrado na etapa de definição apesar de não ser inconcebível causa dificuldades no decorrer do projeto. O pouco tempo acaba forçando a realização de uma definição mais macro com requisitos em alguns casos incompletos e conseqüentemente a falta de verificação de viabilidade da implementação.

“[...]é então com certeza isso também ocorreu, então no levantamento inicial por ter sido feito de uma forma muito macro[...]”

“[...]ocorreu em alguns casos sim. que deu discussão de os dois lerem o mesmo texto, é como interpretar uma lei né, às vezes tem só um parágrafo ali eu entendo que é uma coisa e tu entende que é outra diferente e aí cai em discussão.”

“É teve coisas que na hora da conversa se falou, cliente beleza, vou fazer e na hora de botar a mão na massa se viu que a tecnologia não permitia”

Outro fato que deve-se considerar é a prática do levantamento de requisitos por pessoas pouco experientes e com pouco tempo de empresa. Apesar de todos os analistas deixarem claro que sempre foi necessário aprender algo no decorrer do projeto foi possível identificar que essa necessidade estava acentuada nos analistas classificados como novatos.

Uma forma de contornar tal dificuldade que pode ser ampliada com uma rotatividade elevada seria a utilização de tutoria entre pares de funcionários, como propõem Scott (2008) em seu estudo. A ideia de criar e compartilhar conhecimento de forma mais eficaz realizada por funcionários com mais experiência repassando o conhecimento tácito e explícito sobre os processos, sistemas, regras de negócio, etc., seria um mecanismo para evitar requisitos incompletos. Nos trechos abaixo se pode identificar tal dificuldade.

“[...]eu não tenho muito conhecimento do negócio, para estar questionando assim... não cheguei a questionar, ah tem que fazer de outra forma. Mudar mesmo o que ele(cliente) pediu assim nunca cheguei a mudar.”

Pesquisador: “Entendi, e se o requisito tivesse alguma coisa diferente do que o cliente solicitou? você fez baseado naquele requisito, então se por acaso tiver alguma coisa que não estava ali o cliente não iria receber?”

Entrevistado: “Não vai receber.”

Pontos positivos também são notados nas entrevistas deste estudo. É o caso da internalização de conhecimento que segundo Nonaka e Takeuchi (1998) ocorre quando da incorporação do conhecimento explícito em tácito. O exemplo mais claro e também comum em um Fábrica de *Software* é a observação do código fonte de um programa para aprender o que o mesmo faz.

“Esses casos já aconteceram em outros projetos, que eu recorri a códigos fontes[...].”

“Os outros 10% eu basicamente rodava o sistema e ia por baixo do fonte e mais ou menos olhava...ah ele faz uma busca aqui, traz aqui e ali, cria ali e eu conseguia matar a charada.”

“[...]o resto foi em busca dentro dos fontes do sistema.... teve um ou dois manuais que existiam dentro do sistema que falavam de alguma coisa relacionada ao processo, aquela regra que eu dei uma olhada mas não entendi muito bem....”

“A gente teve que ir atrás de outros códigos fontes que tinham uma regra semelhante para poder implementar também[...].”

Exceto o código fonte que também é o produto de uma empresa de *software* e por isso da sua existência, as demais documentações deixam a desejar segundo os próprios analistas. Segundo eles a maioria do aprendizado vem da socialização do conhecimento. Os motivos principais para a utilização dessa forma de transmissão de conhecimento vai ao encontro com que Nonaka e Takeuchi (1998) dizem, afirmando ser uma forma mais eficaz de adquirir conhecimento. Essa socialização também auxiliou muito na definição do requisito sendo o fator mais importante na opinião dos analistas para que o projeto ficasse de acordo com a solicitação do cliente. Esse processo traz inclusive benefícios para a organização, pois dentro do projeto a socialização do conhecimento de analistas com suas equipes sempre ocorreram. Abaixo uma compilação do que foi dito acima nas palavras dos entrevistados.

“Eu julgo que tá muito precária essa documentação, o conhecimento está na cabeça de poucos e quem quer aprender tem que chegar nessas pessoas.”

“Acontece, eu até prefiro falar com os outros colegas que é uma maneira mais rápida de tu conseguir uma informação. Tu pode muito bem aprender por si só, ir lá e abrir o código fonte e aprender, mas de repente o que o cara vai te falar em cinco minutos tu vai levar uma manhã para...”

“Muitas vezes se eu pegar sozinho, vou fazer assim, mas não lembrei de verificar que impactava lá em outro processo, então eu acho que o contato é essencial.”

“Olha, esse projeto teve esse lado especial, porque na época ele era o módulo que tinha pouca movimentação, ele estava estável não dava muita solicitação de mudança nem correções, então estava muito tácito e estava muito na minha cabeça, e esse projeto como era para replicar esse produto para uma tecnologia nova foi uma oportunidade de ouro para os mais novos, que participavam da equipe. Eles acabaram aprendendo na marra para fazer o produto, aprenderam a regra de negócio que tinha muita coisa que estava presa na minha cabeça. Hoje passados alguns anos eu até não domino mais esse modulo, mas tenho maior orgulho de levantar o telefone e ligar para o cara que participou do projeto e pedir como que faz algum detalhe, isso foi muito bom para disseminar esse conhecimento.”

“Houve esse repasse, até os outros projetos que passaram que ainda complementam esse, todos teve a minha colaboração em cima deles, então é um conhecimento que hoje está nessa questão naquele ponto específico ali eu tenho bastante então o pessoal vem e me pede então a consigo está passando isso sim”

“Eu procuro sempre passar o máximo de conhecimento possível em relação ao tempo que se tem para realizar aquela tarefa[...].”

“Primeiramente na qualidade do desenvolvimento, aí tu evita retrabalho, você tem um melhor desempenho do *software* porque muitas vezes tu falando com um

analista mais experiente ele tem um caminho que seja mais, que seja mais correto eu não diria mas. que o desempenho seja melhor do *software*[...]"

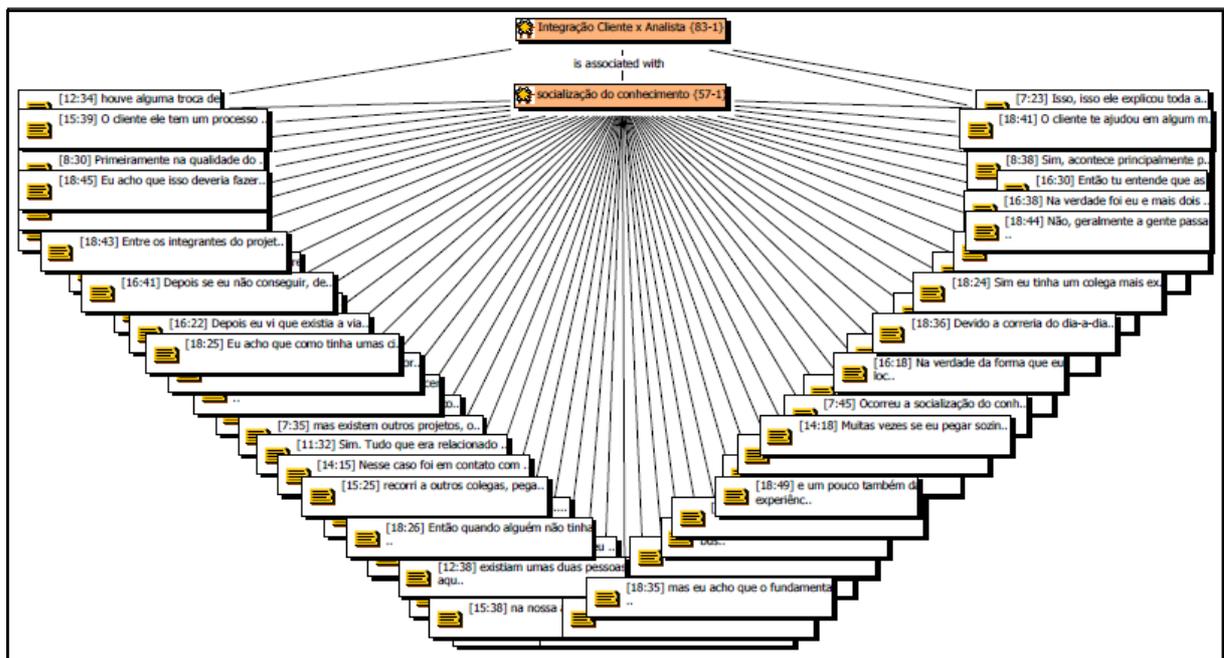
"Sim, acontece principalmente passar para o programador, porque tu tem que explicar para ele como que funciona e aonde ele vai colocar o código, então tu vai ter que indicar. Principalmente para o programador e claro um outro analista no caso se ele venha a fazer um projeto parecido ou que usem o mesmo código, provavelmente eu consiga explicar para ele como foi feito o meu e como funciona aquele programa."

"Sim, para que as demais pessoas tanto analistas quanto programadores que estavam envolvidas no projeto pudessem integrar o projeto, interagir com o usuário, auxiliar no levantamento de requisitos foi necessário um prévio repasse digamos assim de conhecimento e experiência de trabalho no cliente. Também durante o andamento do projeto houve a socialização com a equipe, sempre na medida em que os requisitos estavam sendo identificados."

"De forma formal não, mas informalmente sim... conversas ali dentro da empresa, a gente acabou repassando algumas experiências mas formalmente não tem nem um material, não se chegou a montar nada de histórico disso."

Na Figura 20 gerada pelo *software* Atlas/Ti é possível perceber a quantidade de vezes que a socialização do conhecimento foi identificada nas entrevistas.

FIGURA 20 - Socialização do conhecimento



Fonte: Desenvolvido pelo autor.

Tal integração e socialização de conhecimento entre analistas e clientes se refletem na qualidade do *software*. Esse contato completa lacunas no momento da definição de um requisito tornando o mesmo rico em detalhes eliminando ambiguidades e incompletude. Assim como consideram Chiu (2011) e Barney (2012), essa interação na elicitação dos requisitos proporcionam grandes ganhos na qualidade final do *software* resultando em um produto aderente as necessidades do cliente.

Como dito anteriormente a maior parte dos analistas afirmaram existir em seus projetos mais relevantes, citados nas entrevistas, interação com o cliente e socialização do conhecimento. Nestes projetos todos analistas iteraram que atenderam as necessidades dos clientes com um requisito adequado a sua solicitação, mas com alguns ajustes a serem realizados no decorrer destes projetos.

"Eu queria que tivesse sido muito bom, mas eu digo que foi bom, foi mais que aceitável. O cliente virou ficou satisfeito."

"Sim no final de tudo ele ficou como o cliente solicitou, mas que eu vejo que Bah né... e aí que tá né... deveria também ser a nossa parte de estar talvez questionando mais ali no início para tentar fechar tudo que precisava, que daí... mesmo o que não estava no escopo acaba estressando o cliente, então eu vejo que isso poderia estar com certeza ajudando."

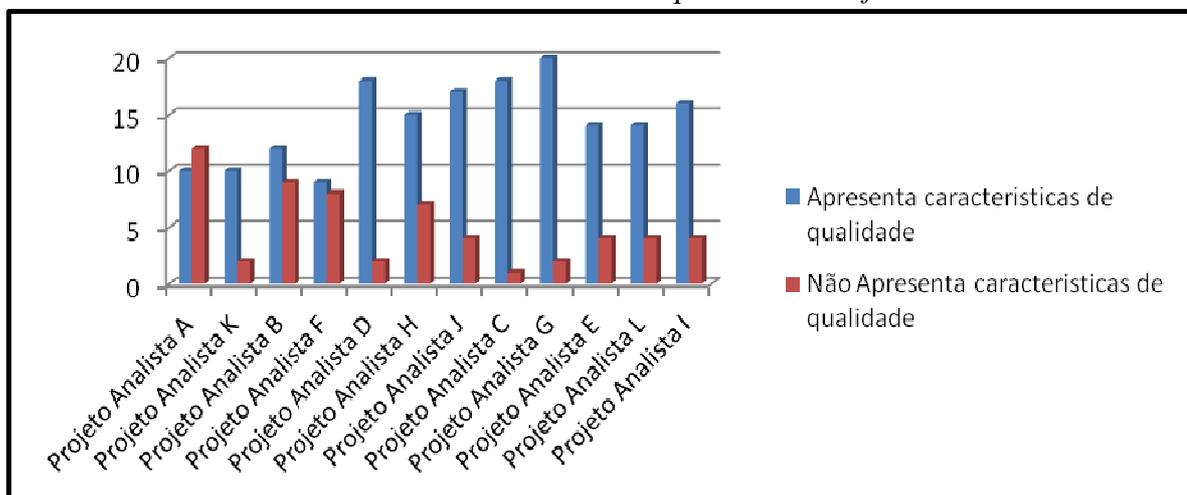
"O desfecho foi bom, não julgo ele ótimo ahn.... porque por algumas situações que ocorreram no momento do desenvolvimento...se perdeu um tempo que não impactou no termino do projeto, mas em algumas situações deixou um pouquinho mais apertado, mas nada que impactasse no projeto e na entrega..."

"Na minha opinião, em termos de engenharia, funcionalidade ele atende, estaria ok. Em termos de prazo ele estourou, ele passou do prazo né, teve que negociar o prazo com o cliente e em termos de qualidade ficou também ok"

"Foi positivo. Por ser um projeto de grande importância para o cliente, senti que o ele estava satisfeito e principalmente tranquilo no que foi realizado pela nossa empresa."

Isso pode ser comprovado pelo Figura 21 abaixo onde em pesquisa realizada com os clientes, mais precisamente com pessoas que exercem função de analista no cliente e são responsáveis pelos projetos solicitados junto à fábrica, foi possível perceber que com exceção de um caso as características de qualidade de *software* são maioria nos projetos. Esta pesquisa teve como finalidade verificar se a expectativa que o analista tinha se comprovava a partir das respostas dadas pelo cliente.

FIGURA 21 - Características da qualidade de *software*



Fonte: Desenvolvido pelo autor.

Um último fato observado, mas não menos importante, são as lições aprendidas pelos analistas. Todos, sem exceção, afirmam que aprenderam algo durante a execução dos projetos julgando um fator importante e necessário nas suas atividades rotineiras. Para eles o fato de aprender possibilita um crescimento dentro da organização e como consequência o desenvolvimento de produtos com maior qualidade de *software*. A compilação das falas abaixo relata um pouco do que foi dito.

“Sim, precisava entender melhor aonde que aquilo ia ser aplicado no final. ahn... era me explicado ah é por causa de tal coisa que é uma lei que exige e tal... mas assim onde que está exatamente aplicabilidade daquilo que me faltou, que eu pude ver depois lá no final a hora que eu fui falar com o usuário final.”

“Sim, aprendi a regra de negócio do cliente, aprendi em relação ao que um levantamento de requisitos pode ter de problemas ele sendo bem ou mal feito.”

“Tudo não, algumas regras não eram de meu conhecimento. Tive que buscar ajuda com o próprio cliente para aprender as regras utilizadas pelo usuário do financeiro. Por exemplo: as regras para o cálculo da devolução dos valores.”

“Acho que agente nunca sabe tudo né, e nesse caso não foi diferente. Eu conhecia digamos o processo macro, mas precisei aprender sim mais detalhes, ou seja, o micro da regra que o projeto envolvia, além de ter que identificar onde aquela alteração que eu estava fazendo ia impactar sempre pensando na qualidade.”

Com base nos resultados obtidos através das técnicas e ferramentas utilizadas foi possível fazer uma análise criteriosa sobre a importância do resgate do conhecimento tácito na definição dos requisitos. Os dados mostram que devido ao fato de uma fábrica de software estar contextualizada em uma área intensiva em conhecimento a utilização e transferência do mesmo é fundamental para os processos do cotidiano principalmente ao utilizar o Sensemaking como instrumento para superar problemas até então desconhecidos na definição de requisitos assim como afirma Dervin (1992).

Fazer emergir o conhecimento tácito e transformá-lo em explícito é uma tarefa essencial como já diriam Nonaka e Takeuchi (1998) e pode ser facilitada com o processo de interação direta entre os polos cliente e analista. Porém, absorver essa gama de dados e informações afloradas nesse momento e relacionar com as técnicas de engenharia de requisitos passa a ser uma atividade dura a ser desempenhada por um analista. Descobrir ou instigar essa habilidade nestas pessoas significa entregar projetos com mais qualidade e consequentemente potencializar a vantagem competitiva da organização.

6 CONSIDERAÇÕES FINAIS

O objetivo desta pesquisa foi verificar em uma Fábrica de *Software* a existência da dimensão tácita do conhecimento na definição de requisitos e sua relação com qualidade do *software*. Foi possível perceber que sim existe uma relação e tal é positiva para a organização. Identificou-se que os processos cognitivos dos analistas como experiência, interpretação, memória, aprendizagem, etc. foram fundamentais para que projetos fossem entregues com a qualidade esperada.

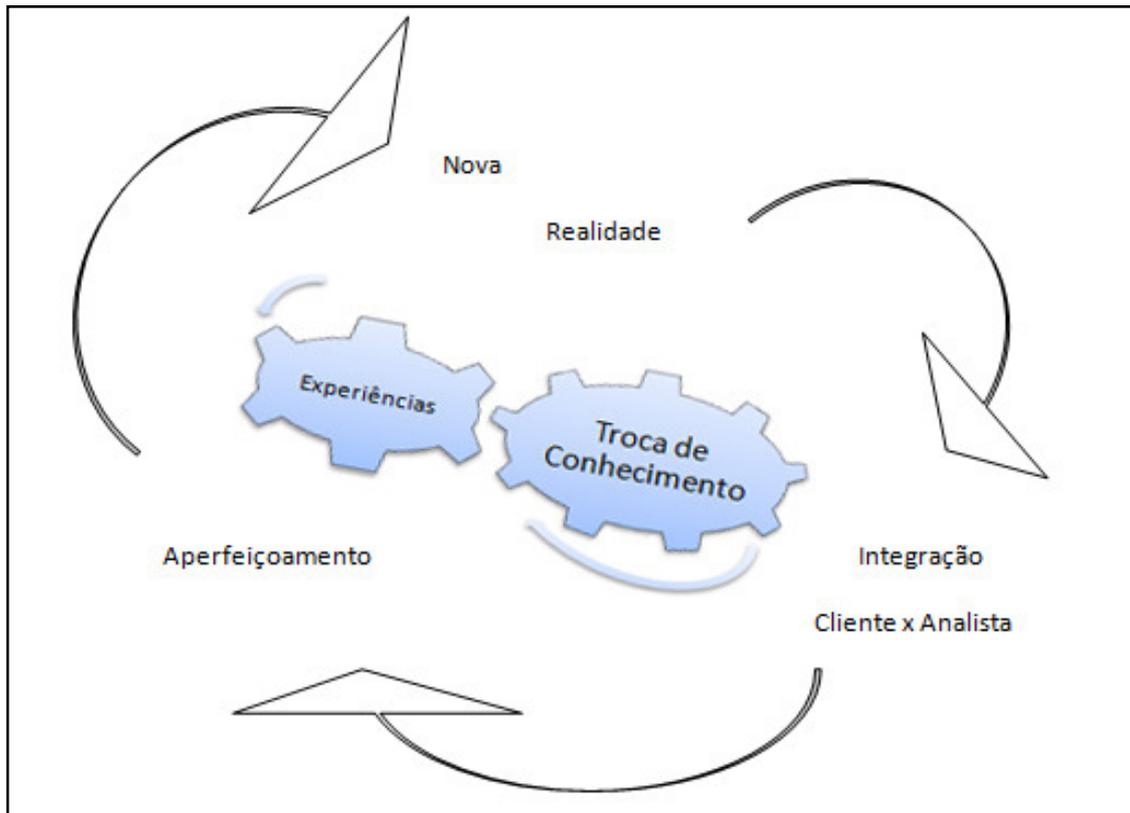
Estes processos que possuem relação direta com o conhecimento tácito foram amplamente identificados na categoria colisão segundo o modelo dos autores Kjærgaard, Kautz e Nielsen (2007) onde percebe-se que a mesma existiu devido às experiências e conhecimento por parte de analistas com mais tempo de empresa. O reflexo dessa colisão na fase de levantamento de requisitos foi extremamente positivo, pois a interação das pontas cliente versus analista possibilitou que as necessidades do cliente estivessem contempladas de forma mais clara e objetiva minimizando o risco de se iniciar o desenvolvimento de um produto fadado ao fracasso.

Essa com certeza é uma das principais funções de um levantamento de requisitos e que também é percebida pelos analistas, pois nas suas entrevistas, sem exceção, todos deixaram claro que essa interação na elicitação dos requisitos é fundamental apesar de nem todos adotarem tal prática. Também fica claro que se por um lado analistas com mais tempo de casa e conseqüentemente mais experiência buscavam sempre essa interação, por outro, analistas considerados novatos tinham certo receio de criar esse contato, justamente pela falta de experiência e conhecimento da regra de negócio.

O problema que um requisito mal elaborado pode causar também é de conhecimento de grande parte dos entrevistados, pois muitos já sofreram com isso. Requisitos mal estimados e com poucos detalhes exigem maior esforço, principalmente de pessoas com menos experiência, o que acaba impactando diretamente na quantidade de horas diminuindo a qualidade final do *software*, gerando maior retrabalho e tendo como resultado menor lucro para a empresa.

Baseado nos fatos evidenciados na pesquisa e relacionando com a visão de Nonaka e Takeuchi (1998) onde afirmam que para inovar é necessário processar informações de fora para dentro criando novos conhecimentos e informações de dentro para fora recriando seu meio chegamos ao esquema observado na figura 22, o qual poderá ser utilizado como guia em processos de definição de requisitos.

FIGURA 22 - Guia para processos de definição de requisitos



Fonte: Desenvolvido pelo autor.

Nesse esquema o formato é circular dando ideia de continuidade, onde o processo permaneça evoluindo até que fique ajustado.

Tudo começa na construção da realidade inicial, ou seja, na necessidade vista pelo cliente, fato este que o faz procurar a fábrica solicitando a demanda. A partir daí é fundamental que ocorra a integração entre as extremidades cliente e analista, momento em que os processos cognitivos como experiência e conhecimento fazem toda a diferença na elaboração de um requisito de *software*, visto que, funcionam como uma espécie de engrenagem que ajusta e refina o documento a partir da dinâmica do conhecimento tácito. Como consequência uma nova realidade é criada o mais próximo da exatidão, mas que se ainda assim não for a ideal dispara novamente o processo fazendo com que este se repita até que se chegue a um requisito 100% aderente.

O efeito do modelo é ter um requisito que atenda o que o cliente necessita, tornando-se a chave para diminuição de um problema existente em praticamente todas as fábricas de *software* que é o retrabalho.

Espera-se que este modelo possa servir como referência para fábricas de *software*, pois na busca do entendimento dos objetivos e ambições de um projeto analistas alcançam uma expertise ímpar desenvolvida a partir do compartilhamento de experiências com seus

clientes. Essa experiência adquirida precisa ser repassada para o resto da equipe algo que já acontece atualmente dentro da fábrica estudada, porém, de uma forma um tanto limitada, alcançando apenas as pessoas envolvidas com o projeto. Esse repasse de conhecimento para as demais equipes pode ser a maneira que uma empresa tem de preparar seus colaboradores sem que eles precisem sofrer na prática para adquirir tal conhecimento evitando, sobretudo, problemas com qualidade e desgaste do cliente.

Como todos analistas deixaram claro que aprenderam com seus projetos, identificar uma maneira eficiente e que seja adequada a uma Fábrica de *Software* de transferir esse conhecimento após o término dos projetos pode ser tratado como um trabalho futuro. Além disso, propor novos fluxos assim como a reestruturação dos processos de definição dos requisitos de modo que permaneçam alinhados com as descobertas deste estudo podem ser alvo de pesquisas futuras.

REFERÊNCIAS

- ABCKER, A. et al. Toward a Technology for Organizational Memories. **Revista IEEE Intelligent Systems**, Los Angeles, v. 13, n. 3, p. 40-48, 1998.
- ALSHAYEB M. Empirical investigation of refactoring effect on *software* quality. **Information and Software Technology**, v. 51, n. 9, p. 1319–1326, set. 2009. Disponível em: http://mail.im.tku.edu.tw/Prof_Shayur/Software%20Project%20Management/99_2/paper/P_5.pdf. Acesso em: 11 out. 2010.
- ASSOCIAÇÃO BRASILEIRA DAS EMPRESAS DE *SOFTWARE*. **Mercado Brasileiro de Software, Panorama e Tendências, 2010**. São Paulo: 2010. Disponível em http://www.abes.org.br/UserFiles/Image/PDFs/Mercado_BR2010.pdf. Acesso em: 11 out. 2010.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR –ISO/IEC 12207**:Tecnologia de Informação - Processos de ciclo de vida de *software*. Rio de Janeiro, 1998.
- BANDEIRA-DE-MELLO, R. In: GODOI, C. K.; BANDEIRA-DE-MELLO, R.; SILVA, A. B. da. (Org). **Pesquisa qualitativa em estudos organizacionais**: paradigmas, estratégias e métodos. São Paulo: Saraiva, 2006.
- BARNEY, S.; et al. *Software* Quality Trade-Offs: A Systematic Map, Information and *Software* Technology. **ScienceDirect**, v. 54, n. 7, p. 651–662, jul. 2012. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0950584912000195>>. Acesso em: 14 nov. 2011.
- BASILI, V.; LINDVALL, M.; COSTA, P. **Implementing the Experience Factory concepts as a set of Experiences Bases**. In: Proc. of the Int. Conf. on *Software* Engineering and Knowledge Engineering, Buenos Aires, Argentina, 2001, p. 102-109.
- BOEHM, B. W. Verifying and validating *software* requirements and design specification. **IEEE Software**, Los Angeles, v. 1, n.1 p. 75-88, 1984. Disponível em: <<http://www.sroga.com/school/cs8263/Verifying%20and%20Validating%20by%20Boehm.pdf>>. Acesso em: 23 fev. 2011.
- BOISOT, M. **Knowledge Assets**: Securing Competitive Advantage in the Information Economy. New York: Oxford University Press, 1998.
- BORGES, L.S.M.; FALBO, R.A. Gerência de Conhecimento sobre Processos de *Software*. In: VIII WORKSHOP DE QUALIDADE DE SOFTWARE e XV SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 2001, Rio de Janeiro. **Anais...** Rio de Janeiro: out. 2001, p. 27-38.
- BOWEN, H. K. et al. The perceptual enterprise machine - seven keys to corporate renewal through successful product and process development. New York: Oxford University Press, 1994.

BRITO, J. **Metodologia para Gestão do Processo de Qualidade de *Software* para Incremento da Competitividade da Mobile**. Disponível em <<http://www.mct.gov.br/temas>>. Acesso em: 19 abr. 2010.

BROOKS, F. No Silver Bullet: Essence and accidents of *software* engineering. **IEEE Computer**, Los Angeles, v.20 , n.4 p. 10-19, 1987

CARDOSO, S.H. **Memória**: o que é e como melhorá-la. Disponível em: <http://www.cerebromente.org.br>. Acesso em: 01 abr. 2011.

CARVALHO, H. G. de C. **Gestão da Informação tecnológica**. Curitiba: CEFETPR, 2000. Disponível em: <www.ppgte.cefetpr.br>. Acesso em: 09 mar. 2001.

CASTELLS, M. **A era da informação**: economia, sociedade e cultura. São Paulo: Paz e Terra, 1999.

CASTELLS, M. **A era da informação**: economia, sociedade e cultura. A sociedade em rede. 9. ed. São Paulo: Paz e Terra, 2006.

CASTOR, E. M. **Fábrica de *Software***: Passado, Presente e Futuro. 2004. 00 f. Dissertação (Mestrado) - União dos Institutos Brasileiros de Tecnologia, Programa de Pós-Graduação Lato Sensu em Tecnologia da Informação, 2004.

CECEZ-KECMANOVIC, D.; JERRAM, C. A Sensemaking Model of Knowledge Management in Organisations. **The European Conference on Information Systems**, Gdansk, Poland, 6-8 un. p. 894-904, 2002. CD-ROM.

CESAR, R. **Fábrica de *Software***: uma vocação nacional?. Disponível em: <<http://www.siscorp.com.br/imprensa/computerworld02.htm?documento=24655&Area=51>>. Acesso em: 28 set. 2010.

CHIU, N.H. Combining techniques for *software* quality classification: An integrated decision network approach. **Journal Expert Systems with Applications**: An International Journal, v. 38, n 4, p. 4618–4625, abr. 2011.

COSTA, V.; ROCHA, A.R. Um ambiente de gestão do conhecimento para apoio ao processo de fornecimento de *software*. In: TERRA, J. C. C.; KRUGLIANSKAS, I. (Org.). **Gestão do conhecimento em pequenas e médias empresas**. 2 ed. Rio de Janeiro: Campus, 2003.

CUSUMANO, M. A. **Japan's *Software* Factories**: A Challenge to US Management. New York: Oxford University Press, 1991.

CUSUMANO, M. A. The puzzle of Japanese *software*. **Communications of the ACM**, v. 48. n.7, p. 26-27, jul. 2005

DAFT, R. L.; WEICK, K. E. Toward a model of organizations as interpretations systems. **Academy of Management Review**, v. 9, n. 2, p.284-295, abr. 1984.

DAMM, D.; SCHINDLER, M. **Security issues of a knowledge médium for distributed projects work**. International. Journal. of Project Management, v. 20, n. 1, p. 37-47, jan. 2002.

DAUGULIS, A., Time aspects in requirements engineering: or every cloud has a silver lining. **Requirements Engineering**, v. 5, n. 3 p.137— 143, 2000. Disponível em: <<http://www.springerlink.com/content/ec41bbah8tfg5rlx/>>. Acesso em: 23 out. 2011.

DAVENPORT, T. H.; PRUSAK, L. **Conhecimento empresarial**. Rio de Janeiro: Campus, 1998.

DAVIS, A. M. **Software requirement: objects, functions and states**. Englewood Cliffs: Prentice Hall, 1993.

DEMIR, A. **Comparison of model-driven architecture and software factories in the context of model-driven development**. 2006 In: FOURTH INTERNATIONAL WORKSHOP ON MODEL-BASED DEVELOPMENT OF COMPUTER-BASED SYSTEMS AND THIRD MODEL-BASED METHODOLOGIES FOR PERVASIVE AND EMBEDDED SOFTWARE (MBD/MOMPES 2006). Universidade de Munchen, 2006. Disponível em: <<http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1604767&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F10681%2F33715%2F01604767.pdf%3Farnumber%3D1604767>>. Acesso em: 23 out. 2011.

DERVIN, B. **An overview of Sense-Making research: concepts, methods and results to date**. In: INTERNATIONAL COMMUNICATIONS ASSOCIATION ANNUAL MEETING, 1983. Dallas, Texas, mai. 1983 Disponível em: <http://www.eca.usp.br/prof/sueli/sm1983_4.html>. Acesso em: 11 mai. 2011.

DERVIN, B. From the Mind's Eye of the User: The Sense-Making Qualitative- Quantitative Methodology. 1992. In: GLAZIER, J.; POWELL, R. **Qualitative Research in Information Management**. Englewood: Libraries Unlimited, 1992. Disponível em: <http://communication.sbs.ohio-state.edu/sensemaking/art/artabsdervin92powell.html>. Acesso em: 05 jan. 2011.

DERVIN, B. **On Studying Information Seeking Methodologically: The Implications of Connecting Metatheory to Method**. Information Processing and Management. 35.6, p. 727-750, 1999.

DERVIN, B. **What we know about information seeking and use and how research discourse community makes a difference in our knowing**. 2001. Disponível em <<http://communication.sbs.ohio-state.edu/sense-making/art/artabsdervin01nlm.html>>. Acesso em: 13 mai. 2011.

DESOUZA, K.C. Barriers to Effective Use of Knowledge Management Systems in *Software Engineering*. In: Communications of the ACM, v. 46, n.1. 2003.

DIAS, A. S. **Uso de conhecimentos teóricos e de especialista para Previsão de Demanda**. 2004. 181 f. Dissertação (Mestrado) – Universidade Federal de São Carlos. Programa de Pós-Graduação em Engenharia de Produção, 2004.

DRUCKER, P. F. **A sociedade pós-capitalista**. 3 ed. São Paulo: Pioneira, 1993.

DWORKIN, M.; FOREMAN-WERNET, L.; DERVIN, B. Sense-Making and television news: an inquiry into audience interpretations. **The Electronic Journal of Communication**, v.9, n. 2, p.3-4. 1999. Disponível em:<<http://www.cios.org/EJCPUBLIC/009/2/009217.html>>. Acesso em: 24 jul. 2011.

EVANS, B.M.; CHI, E.H. **Towards a Model of Understanding Social Search**. San Diego: ACM, 2008.

FABRI, J. A. et al. Desenvolvimento e replicação de uma Fábrica de *Software*. In: VI SIMPÓSIO INTERNACIONAL DE MELHORIA DE PROCESSOS DE SOFTWARE, São Paulo. **Anais...** São Paulo: 2004, p. 1-50. Disponível em: <www.simpros.com.br>. Acesso em 21 jan. 2011.

FERNANDES, A. A.; TEIXEIRA, D. de S. **Fábrica de Software: implantação e gestão de operações**. São Paulo: Atlas, 2004.

FERREIRA, S. M. S. P. **Estudos de Necessidades de Informação: dos paradigmas tradicionais à abordagem Sense-Making**, ABEBD, 1997. Disponível em: <http://www.eca.usp.br/nucleos/sense/textos/sumar.htm>. Acesso em: 10 jan. 2011.

FISCHER, G.; OSTWALD, J. Knowledge Management: Problems, Promises, and Challenges. In: **IEEE Intelligent Systems**. 2001

FURNAS, G. W.; RUSSELL, D.M. **Making Sense of Sensemaking**. . In: CHI '05 EXTENDED ABSTRACTS ON HUMAN FACTORS IN COMPUTING SYSTEMS, Portland, 2005. **Anais...** Portland, OR, USA, abr.l 2 - 7, 2005, p. 2115-2116.

GIL, A. C. **Como elaborar projetos de pesquisa**. São Paulo: Atlas, 1991.

GIOIA, D. A.; MEHRA, A. Sensemaking in organizations In: WEICK, KARL E. Thousand Oaks (CA): Sage. Academy of Management. **The Academy of Management**. v. 21, n. 4, p. 1226-1240, out. 1996.

GOMEL, M. M. **O papel da capacitação tecnológica no desempenho exportador da indústria brasileira de software**. 2005. Dissertação (Doutorado) – Universidade de São Paulo, Faculdade de Economia e Administração, 2005.

GRANT, R. Toward a Knowledge: based Theory of the Firm. **Strategic Management Journal**. Washington, v.17, n17, p. 109-122. 1996.

GREENFIELD, J.; SHORT, K. *Software factories assembling applications with patterns, models, frameworks and tools*. In: CONFERENCE ON OBJECT ORIENTED PROGRAMMING SYSTEMS LANGUAGES AND APPLICATIONS, 2003. **Anais...**, USA: 2003.

HALL, R.; ANDRIANI, P. Managing knowledge associated with innovation. **Journal of Business Research**. Durham v. 56, n.2, p. 145-152, fev.2003. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0148296301002879>> Acesso em: 21 mai. 2011.

HARVEY, D. **Condição pós moderna**. São Paulo: Edições Loyola, 1994.

HAYATI, D; KARAMI, E.; SLEE, B. Combining qualitative and quantitative methods in the measurement of rural poverty. **Social Indicators Research**. Netherlands. v.75, n. 3, p.361-394, 2006. Disponível em: <<http://www.jstor.org/discover/10.2307/27522539?uid=3737664&uid=2129&uid=2&uid=70&uid=4&sid=21100837751061>>. Acesso em: 21 mai. 2011.

HUBER, G. P.; DAFT, R. L. The information environments of organizations. In: JABLIN, F. M. et al. **Handbook of Organizational Communication**. Newbury Park, CA: Sage, 1987.

HUMPHREY, W. S. **Managing the software process**. Canada: Addison-Wesley, 1989.

INTERNATIONAL DATA CORPORATION. **Black Book**, 2010. Disponível em <http://www.idc.com/>. Acesso em: 01 ago. 2010.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **INFORMATION TECHNOLOGY: Guideline for the Evaluation and Selection of CASE Tools**. [ISO/IEC 14102:1995(E)]. Geneva, Switzerland: International Organization for Standardization, 1995

ISABELLA, L. A. Evolving interpretations as a change unfolds: How managers construe key organizational events. **Academy of Management Journal**. v33, n.1, p.7-41. 1990.

KÄSER, P. A. W.; MILES, R. E. **Understanding knowledge activists' successes and failures**. Long Range Planning, 2002.

KIKOSKI, C.K.; KIKOSKI, J.F. **The Inquiring organization: Tacit knowledge, conversation, and knowledge creation skills for 21st-Century organizations**. London: Praeger, 2004.

Kitchenham, B. Empirical paradigm :the role of experiments. In: Proceedings of the 2006 international conference on Empirical software engineering issues: criticalassessmentand future directions.2006, Berlim. **Anais...Berlim: 2006**. p. 25–32.

KJÆRGAARD A.; KAUTZ K.; NIELSEN P. A. Making Sense of Project Management - A Case of Knowledge Management in *Software* Development. In: PROCEEDINGS OF THE FIFTEENTH EUROPEAN CONFERENCE ON INFORMATION SYSTEMS, St. Gallen. **Anais...University of St. Gallen, 2007**. Disponível em: <<http://www.mendeley.com/research/knowledge-management-practicethe-case-agile-software-development-3/>>. Acesso em 15 jul. 2011.

KOGUT, B.; ZANDER, U. Knowledge of the Firm, Combinative Capabilities, and the Replication of Technology. *Organization Science*, 1992.

KOTONYA, Gerald; SOMMERVILLE, Ian. Requirements engineering (Processes and techniques). England: J. Wiley & Sons, 1998. p. 282. A dynamic theory of organizational knowledge creation. **Organization Science**. v. 5, n. 1, p. 14-37, fev.1994.

KRAKOWIAK, S. Principes des systèmes d'exploitation des ordinateurs. Paris: Bordas, 1985.

KUDIKYALA, U.K.; VAUGHN, R.B. *Software Requirement Understanding Using Pathfinder Networks: Discovering and evaluating mental models*. **Journal of systems and software**. v. 74, n. 1, p.101-108, jan. 2005.

LANDES, D.; SCHNEIDER, K.; HOUDEK, F. Organizational Learning and Experience Documentation in Industrial *Software* Projects.: **International Journal of Human-Computer Studies**, v. 51, n. 3, p. 643-661, set. 1999. Disponível: <<http://www.sciencedirect.com/science/article/pii/S107158199990280X>>. Acesso em: 15 jul. 2011.

LARMAN, C. **Utilizando UML e padrões:** uma introdução à análise e ao projeto orientados a objetos e ao processo unificado. São Paulo: Bookman, 2004.

LEFFINGWELL, D. Calculating the Return on Investment from More Effective Requirements Management. *American Programmer*, v.10, n.4, p. 13-16, abr. 1997.

LEIDERMAN, S.M. The World Problem of Environmental Emigration from Polluted Regions. **The Nato Advanced Research**. Ucrânia, v.5, n.7,set. 2002. Tipo de obra?

LEITE, J. Documentação de *Software*. In: ROCHA, A.R.C.; MALDONADO, J.C.; WEBER K. **Qualidade de software:** teoria e prática. São Paulo: Prentice Hall, 2001.

LINDVALL, M. et al., Lessons Learned about Structuring and Describing Experience for Three Experience Bases. Berlin: LNCS, 2001.

LONGCHAMP, J. A Structured Conceptual and Terminological Framework for the *Software* Process Engineering. In: INTERNATIONAL CONFERENCE ON THE SOFTWARE PROCESS. Anais... Berlin, 1993. P. 41-53.

LOVE, P. E. D.; EDUM-FOTWE, F.; IRANI, Z. Management of knowledge in project environments. **International Journal of Project Management**. v. 21, n. 3, p. 155-156. 2003.

MACHADO, C. Documentação de *Software*. In: ROCHA, A.R.C.; MALDONADO, J.C.; WEBER K. **Qualidade de software:** teoria e prática. São Paulo: Prentice Hall, 2001.

MANIS, M. **Processos Cognitivos**. São Paulo: Herder, 1973.

MURPHY, T. **Achieving business value from technology:** a practical guide for today's executive. New Jersey: John Wiley & Sons, 2002.

NATALI, A. C. Uma Infraestrutura para Gerência de Conhecimento em um Ambiente de Desenvolvimento de *Software*. Dissertação (Mestrado) Universidade Federal do Espírito Santo, 2003.

NONAKA, I.; TAKEUCHI, H. **Criação de conhecimento na empresa**. 16 ed. Rio de Janeiro: Campus, 1997.

NONAKA, I.; TAKEUCHI, H. **Gestão do conhecimento**. Tradução: Ana Thorell. Porto Alegre: Bookman, 2008.

NONAKA, I.; TOYAMA, R.; KONNO, N. S. **ba and leadership:** a unified model of dynamic knowledge creation. **Long Range Planning**. London, v. 33, n.1, p. 5-34, fev.2000.

O'LEARY, D. E. **Knowledge Management Systems:** Converting and Connectiong, IEEE Computer, 1998.

OLIVEIRA, D.; NETO, A. Fábrica de *Software*: Promovendo a Criação de Empresas Competitivas em Tecnologia da Informação. In: ENEGEP, Ouro Preto. **Anais Eletrônicos**. ABEPRO. 1 CD-ROM. Minas Gerais.

- OSTERLOH, M.; FREY, B. S. Motivation, knowledge transfer, and organizational forms. **Organization Science**, v. 11, n. 5, p. 538–550, set/out. 2000. Disponível em: <<http://links.jstor.org/sici?sici=1047-7039%28200009%2F10%2911%3A5%3C538%3AMKTAOF%3E2.0.CO%3B2-C>>. Acesso em: 23 jan. 2011.
- PARREIRAS, F.S.; BAX, M.P. A gestão de conteúdo no apoio à engenharia de *software*. In: CONGRESSO BRASILEIRO DE GESTÃO DO CONHECIMENTO, **Anais...** São Paulo: KMBRASIL, 2003.
- PESSOA, M. S. P. **Introdução ao CMM - Modelo de Maturidade da Capacidade de Processo de Software**. 2003. 77f. Dissertação (Mestrado) – Universidade Federal de Lavras. Curso de Pós-graduação “Lato Sensu” à Distância: Melhoria de Processo de Software, Fundação de Apoio ao Ensino, Pesquisa e Extensão, 2003.
- PFLEEGER, S. L. **Engenharia de software: teoria e prática**. 2. ed. Rio de Janeiro: Pearson, 2004.
- PFLEEGER, S.L. **Software Engineering: Theory and Practice**. New Jersey: Prentice Hall, 1998.
- PLANT R.T., Factors in *software* quality for knowledge: based systems. **Journal Information and Software Technology**. Miami. v. 33 n. 7. p.527 – 536, 1991.
- POLANYI, M. **The Tacit dimension**. London: Routledge & Keagan Paul, 1966.
- PRESSMAN, Roger S. **Engenharia de software**. 6.ed. São Paulo: McGraw-Hill, 2006.
- QIU, Y.F.; CHUI, Y.P.; HELANDER, M.G. A cognitive approach to understanding knowledge-based virtual team decision making. **Product Design, Int. J. Intelligent Enterprise**. v. 1, n. 1, p. 45 – 64, 2007.
- RATCHEV, S.M. et al. Knowledge based requirement engineering for one-of-a-kind complex systems. **Journal of Knowledge Based Systems**. v.16, n. 1, p.1-5, 2003.
- ROCHA, A.R.C; MALDONADO, J.C; WEBER, K. **Qualidade de software: teoria e prática**. São Paulo: Prentice Hall, 2001.
- ROSELINO, J. E. **A indústria de software: o modelo brasileiro em perspectiva comparada**. 2006. 222f. Dissertação (Doutorado) - Universidade Estadual de Campinas, Instituto de Economia, 2006.
- RUSSELL, D.M. et al. Cost structure of sensemaking. In: THE INTERCHI '93 CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 1993. Amsterdã. **Anais...** Palo Alto: California, 1993. Disponível em: <<http://www2.parc.com/istl/groups/uir/publications/items/UIR-1993-10-Russell.pdf>>. Acesso em: 12 mai. 2011.
- SAMPAIO, A.L.S. et al. Método para Definição de Requisitos de *Software* de um Sistema a Partir das Necessidades dos seus Stakeholders In: VII SIMPÓSIO INTERNACIONAL DE MELHORIA DE PROCESSOS DE SOFTWARE, 2005, São Paulo. **Anais...** São Paulo: 2005.

SAMPAIO, S. E. K. **O desenvolvimento da aglomeração produtiva de *software* de Curitiba**. 2006. 158f. Dissertação (Mestrado) – Universidade Federal do Paraná, 2006.

SANCHES, R. Processos do ciclo de vida: processos de apoio, documentação de software. In: DA ROCHA, A.R.C.; MALDONADO, J.C.; WEBER, K.C. (Orgs.). **Qualidade de software: teoria e prática**. São Paulo: Prentice Hall, 2001.

SCOTT, E.B. Impact of Peer Mentor Training On Creating and Sharing Organizational Knowledge. **Journal of Managerial Issues**. Pittsburg, v. 20, n. 1, p.11-29, abr. 2008.

SHAW, M.L.G.; GAINS, B.R. Integrated knowledge acquisition architectures. **Journal of Intelligent Information Systems**. v.1, p. 22, 1992. Disponível em: <<http://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CEsQFjAA&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.47.4916%26rep%3Drep1%26type%3Dpdf&ei=YSHRT5X>>. Acesso em: 12 mai. 2011.

SILVA FILHO, A. M. Produzir e exportar *software*: necessidades do Brasil. **Espaço Acadêmico**. n. 31. dez., 2003. Disponível em: <<http://www.espacoacademico.com.br/031/31amsf.htm>>. Acesso em: 12 mai. 2011.

SILVA, E. L. da; MENEZES, E. M. **Metodologia da pesquisa e elaboração de dissertação**. 4. ed. Florianópolis: UFSC, 2005.

SIY, H, P. et al. Making The *Software* Factory Work: Lessons from a decade of experience. In: SEVENTH INTERNATIONAL SYMPOSIUM ON *SOFTWARE* METRICS. 2001, London. **Anais...** London: England, 2001.

SOCIEDADE PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO - **Relatório anual 2002**. Disponível em: http://www.softex.br/_asoftex/RelatoriosAnuais/Relat%C3%B3rio%20Anual%202002.pdf. Acesso em: 10 out. 2010.

SOFTEX. **Relatório Anual**. 2002. Disponível em: <http://www.softex.br/_asoftex/RelatoriosAnuais/Relat%C3%B3rio%20Anual%202002.pdf>. Acesso em: 11 jan. 2012.

SOMMERVILLE, I. **Engenharia de *software***. 6.ed. São Paulo: Addison-Wesley, 2003.

SOMMERVILLE, I. **Engenharia de *Software***. 8 ed. São Paulo: Pearson Education do Brasil, 2007.

SOMMERVILLE, I. ***Software Engineering***: International Computer Science Series. 5 ed. Califórnia: Addison-Wesley, 1995.

SOMMERVILLE, I.; SAWYER, P. **Requirements Engineering a Good Practice Guide**. England: John Wiley & Sons, 1997.

SOUZA, A. D. Estudo e avaliação da área de processo de planejamento de projeto de acordo com o modelo CMMI-SW Nível 2 na empresa SW Quality situada em Lavras-MG. 2004. 170f. Monografia (Graduação) – Universidade Federal de Lavras, Graduação em Ciência da Computação, 2004. Disponível em: <<http://www.comp.ufla.br/monografias/ano2004/ano2004.htm>>. Acesso em: 06 out. 2010.

STANDARD GLOSSARY OF *SOFTWARE* ENGINEERING TERMINOLOGY. Disponível em: <http://standards.ieee.org/findstds/standard/610.12-1990.html>. Acesso em: 21 fev. 2011.

SVEIBY, K. E. **A nova riqueza das organizações**. Rio de Janeiro: Campus, 1998.

TAURION, C. **Software livre: potencialidades e modelos de negócio**. Rio de Janeiro: Brasport, 2004.

TEECE, D. J., **Capturing Value from Knowledge Assets: The New Economy, Markets for Know-how, and Intangible Assets**. **California Management Review**. Califórnia, v. 40, n.55, p.79, 1998.

TERRA, J.C.C. **Gestão do Conhecimento: o grande desafio empresarial**. São Paulo: Negócio, 2000.

THAYER, Richard; DORFMAN, Merlin. **System and Software Requirements Engineering**. IEEE Computer Society Press Tutorial. 2000.

THOMAS, J. B.; CLARK, S. M.; GIOIA, D. A. Strategic sensemaking and organizational performance: linkages among scanning, interpretation, action, and outcomes. **Academy of Management Journal**. Pensilvânia, v. 36, n. 2, p. 239-270, 1993.

TOTVS. Disponível em: <<http://www.totvs.br>>. Acesso em: 11 jan. 2012.

VALENTIM, M.L.P. Formação: competências e habilidades do profissional da informação. In: VALENTIM, M. L. P. (Org.). **Formação do profissional da informação**. São Paulo: Polis, 2002.

VERGARA, S. C. **Métodos de pesquisa em Administração**. São Paulo: Atlas, 2005.

WEBER, K. C.; ROCHA, A. R. C.; NASCIMENTO, C. J. **Qualidade e Produtividade em Software**. 4 ed. São Paulo: Makron Books, 2001.

WEICK, K. E. **Sensemaking in organizations**. Thousand Oaks: Sage, 1995.

WOODWARD, A. Winning success from Babbage's failure. **IEE Computing & Control Engineering**, v. 16, n. 5, p.18-21, 2005.

YANG, S.C.; FARN, C.K. Social capital, behavioural control, and tacit knowledge sharing: a multi-informant design. **International Journal of Information Management**. v. 29, n. 3, p. 210-218, jun. 2009.

YIN, R. K. **Estudo de Caso: planejamento e método**. Porto Alegre: Bookman. 2005.

YOURDON, E. **Análise Estruturada Moderna**. 3 ed. Tradução Dalton C. de Alencar. Rio de Janeiro: Campus, 1990.

ANEXO A - Termo de Consentimento



Caxias do Sul, 15 de junho de 2011.

TERMO DE CONSENTIMENTO

Vimos, por meio desta, autorizar o aluno de mestrado da Universidade de Caxias do Sul, Rafael Poltronieri Panozzo, CPF: 0003.102.640-06, a realizar uma pesquisa com fim científico sobre o tema: Estudar como a interação cognitiva cliente/analista pode promover a socialização de conhecimentos tácitos no processo de análise de requisitos contribuindo para melhorar a qualidade de software, em nossa empresa inscrita sob o CNPJ 07.428.629/0001-86, incluindo a divulgação dos resultados.

Atenciosamente,



Sabrina Rugeri
Coord de Recursos Humanos
Empresa: Seventeen Tec da Informação (TOTVS)

Drª. Ana Cristina Fachinelli
Professora Orientadora
PPGA/UCS

APÊNDICE A – Roteiro de Entrevistas

1º Passo: narração de eventos ordenados no tempo ou sem ordenar

1. Dentre todos os projetos que já realizou, escolha um com o critério de: mais importante, mais problemático, mais lembrado ou simplesmente o mais recente. (a ideia é obter dados sobre situações memoráveis ou mais significativas)
2. A solicitação do cliente surgiu a partir de algum problema?
3. Você iniciou o processo de levantamento de requisitos? Como o mesmo foi realizado?
4. Você entende que a definição dos requisitos está clara e adequada a solicitação do cliente?
5. Alguma atitude foi tomada pensando na qualidade final do *software*?
6. Você realizou algum contato com o cliente? De que forma foi? Como o cliente reagiu a esta sua solicitação?
7. Você julga importante este tipo de atividade por parte dos analistas?

2º Passo: expressão de confusões, ideias, emoções, sentimentos, perguntas e conclusões.

8. Existiram dúvidas na elicitación dos requisitos?
9. No seu entendimento, algum requisito está ambíguo, incompleto ou até contraditório com a solicitação feita pelo cliente?
10. Tudo o que o cliente solicitou é de seu conhecimento? Você precisava aprender algo mais?
11. O cliente sabia exatamente o que queria? Se sabia, soube explicar?
12. Existe uma forma viável de implementar a demanda?
13. Foi necessário contatar o cliente para mudar a definição do escopo e os requisitos?
14. O cliente entendeu? De que forma ele reagiu?
15. Na negociação com o cliente, sempre se aceita o que o cliente quer?

3º Passo: Respostas às perguntas

16. O que foi feito no momento de confusão ou dúvida?
17. Porque você conduziu por este caminho? O que foi feito para elucidar os requisitos?
18. Você acredita que suas atitudes podem gerar impactos na qualidade do *software*?
Positivamente ou Negativamente?
19. As horas reduzidas ou estouradas geram que tipo de impacto no projeto?
20. Experiências de projetos passados lhe ajudaram em algum sentido?
21. Você aprendeu com esse projeto?
22. O cliente o ajudou em algum momento? Como?
23. No final ficou como havia previsto o requisito?
24. Ocorreu socialização do conhecimento tácito com a equipe?
25. Quando encontra solução como ele fez isso, inspiração, experiências anteriores?
26. Como adquiriu esse conhecimento tácito? Veio através do explícito?
27. Esse conhecimento teve influencia no projeto
28. Qual o desfecho do projeto p você? foi bom?

APÊNDICE B – Mapas de Associação de ideias

Mapa 1 – Criação

Entrevistado: Analista A		
Construção da Realidade Inicial	Percepção/Interpretação na Construção da Realidade Inicial	Decisão na Construção da Realidade Inicial
<p>Na verdade foi uma demanda da empresa de modernizar o produto fazer uma cara toda nova e foi eleito um cliente para ser o piloto e aí alguma coisa foi customizada com a cara dele.(p02)</p> <p>Não, na verdade a necessidade foi de mercado, o nosso produto estava obsoleto e a empresa optou por fazer um produto tudo novo então a empresa optou por um cliente em implantação para ser o piloto, não é um cliente que tinha problemas, e ele tinha problemas com o produto antigo não era o nosso, e aí ele optou para migrar para o nosso e aí foi colocado o produto novo. (p02)</p> <p>Bom a demanda interna foi basicamente de modernizar de pegar o produto que já existia, e identificar em uma análise grande, pontos falhos, falhos não, mas que davam muito esforço para usuários ou a usabilidade não era boa e montar uma versão mais</p>		

<p>otimizada pro produto, diminuir a carga, o esforço de treinamento do usuário final, o esforço de aprendizado, tornar o <i>software</i> mais rápido, e o cliente final perder menos tempo com ações pequenas. (p03)</p>	<p>Olha para o cliente teve muita coisa nova... ele não esperava vamos dizer assim, ele esperava menos ele esperava algo mais simples que era realidade antiga deles né, mas aí quando eles viram para que lado nosso produto estava andando eles compraram a idéia e começaram a investir por esse lado de otimização dos processos deles também e aí tornou algumas conversas bem interessantes. (p04)</p>	<p>Muita coisa nova entrou no produto muitas regras novas, funcionalidades novas também. (p04)</p>
---	--	--

Entrevistado: Analista B		
Construção da Realidade Inicial	Percepção/Interpretação na Construção da Realidade Inicial	Decisão na Construção da Realidade Inicial
<p>Ela partiu do cliente, que eu me lembro do projeto que eu fiz que eu acho que foi o mais interessante e o primeiro que me veio na cabeça, ahn... partiu de uma solicitação do cliente e ele foi uma complementação de um projeto que já havia sido desenvolvido por outro analista, então foi um complemento esse projeto. (p02)</p>	<p>Ahn... eu tive algumas dificuldades para entender porque como ele era o uma continuidade né, faltou um pouco de entendimento meu né no início. (p04)</p> <p>[...] então eu acho que nisso assim eu comecei a cuidar mais o que eu perguntava né, ir mais atrás da pessoa que fez a o orçamento e não do cliente, porque senão eu ia acabar estressando o cliente porque ele bah! ele deve ter sentido que...como botaram uma outra</p>	

	<p>pessoa para fazer, sendo que eu já tinha explicado tudo para aquela pessoa? (p06)</p> <p>É, porque talvez não caberia a mim fazer esse projeto. (p06)</p> <p>Acredito que sim, porque no final ali acabou dando uns problemas e eu tive que recorrer a ajuda daquela pessoa que fez a orçamentação, sentar do lado para que eu pudesse conseguir resolver né. (p06)</p>	
--	--	--

Entrevistado: Analista C		
Construção da Realidade Inicial	Percepção/Interpretação na Construção da Realidade Inicial	Decisão na Construção da Realidade Inicial
<p>Sim, o cliente tinha problema, de um processo onde eles realizavam ele todo manual e queriam de alguma forma passar para dentro do sistema essas informações tanto de forma automática quanto para guardar informações que anteriormente eles não tinham. (p02)</p> <p>Nessa situação foi o cliente. O analista de negócio do cliente fez um pré-levantamento e eu fui validar esse pré-levantamento ahn... na intenção de levantar os requisitos mesmo, definir os requisitos... eu fui em umas duas ou três reuniões não consigo me lembrar. (p02)</p>	<p>[...] por diversas vezes o requisito, a solicitação do cliente implicava em outras áreas do sistema e precisaram ser modificadas ou adequadas [...].(p02)</p> <p>Depois dos requisitos montados para aí sim iniciar o projeto não. Não porque eu já tinha o conhecimento do todo... (p03)</p> <p>Sim, ele montou documento os fluxos que me auxiliaram a não ter divergências e ambiguidade. [...] nem falhas no processo, o processo estava todo desenhado... o fluxo do processo estava todo desenhado. (p04)</p>	

		[...] nós definíamos o que ia ser feito e como poderia ser feito todo o fluxo de informações dentro do sistema[...].(p04)
--	--	---

Entrevistado: Analista D		
Construção da Realidade Inicial	Percepção/Interpretação na Construção da Realidade Inicial	Decisão na Construção da Realidade Inicial
<p>Sim ela partiu do cliente, para atender uma necessidade dele. (p02)</p> <p>Para ele era uma melhoria. (p02)</p> <p>Estava claro o que ele queria fazer, agora o modo como ele queria fazer né, não estava bem definido. (p04)</p>	<p>É foi bem dessa forma mesmo, ahn... o que estava proposto no orçamento, as funcionalidades foram entregues. Daí essa questão do entendimento você já elimina essa possibilidade quando entrar em contato com o cliente porque você geralmente vai ligar quando você não entendeu algum requisito ou uma funcionalidade ou quando ocorre de na análise você perceber que aquilo não é viável ou quando tenha alguma alteração na funcionalidade[...].(p06)</p> <p>Eu não tive dúvidas depois de ter lido o requisito, então essa parte da engenharia estava tranquilo, não era muito complicada e era uma implementação no sistema, era uma melhoria que não impactava em outras funcionalidades do sistema, então era só uma pequena alteração no produto que já existia e aí não tinha tanto impacto? (p06)</p> <p>Sim, a proposta como colocado no orçamento. (p06)</p>	<p>A funcionalidade eu mantive com o que estava no orçamento, agora o "como vai ser feito" ficou diferente. (p04)</p>

Entrevistado: Analista E		
Construção da Realidade Inicial	Percepção/Interpretação na Construção da Realidade Inicial	Decisão na Construção da Realidade Inicial
<p>Sim, partiu do cliente e era um problema. O cliente sempre reajustou os valores de mensalidades na troca de faixa etária dos beneficiários que completam 60 ou 70 anos. Este processo teve que ser revisto por parte da operadora em virtude do ingresso de uma Ação Civil Pública, em desfavor do cliente, questionando os reajustes por faixa etária em relação aos idosos. E daí como não existe nenhum processo no sistema que atenda o acordo com a defensoria pública sobre o reenquadramento etário, foi necessário a criação deste projeto. (p02)</p> <p>Sim. Primeiramente o cliente descreveu a sua necessidade e o que queria do sistema para atender esta necessidade. [...] baseado no detalhamento desse novo processo que foi descrito de forma clara e detalhada pelo cliente, ficou muito mais fácil especificar os requisitos e descrever de forma macro o que o sistema iria fazer para atender esses requisitos né. Após este processo, foi feita uma nova reunião com todos os envolvidos no projeto para a leitura dos requisitos levantados. Depois dessa reunião então se iniciou de fato o desenvolvimento do orçamento. (p03)</p>		

	<p>Sim, a maioria das dúvidas estavam relacionadas ao processo de como chegar a uma determinada informação e não do que era a necessidade do cliente. Por exemplo: Para os usuários reenquadrados era necessário devolver o valor cobrado indevido.</p> <p>A necessidade é clara, como deverá ocorrer a devolução está claro, o que o sistema deverá fazer está claro, agora, como chegar no valor que era preciso devolver? essa dúvida não estava clara, aí eu tive que pedir para o usuário descrever passo a passo o cálculo. (p06)</p>	
--	---	--

Entrevistado: Analista F		
Construção da Realidade Inicial	Percepção/Interpretação na Construção da Realidade Inicial	Decisão na Construção da Realidade Inicial
<p>Bom essa demanda então ela partiu do cliente certo, basicamente ficaram em negociação ali o cliente e empresa um ano com essa parte do levantamento inicial, essa negociação, mas sempre partiu da necessidade do cliente. (p02)</p> <p>Bom essa parte ela surgiu do cliente porque existia um processo totalmente manual né, então nada informatizado, fora do sistema que era o processo de vendas de plano. (p02)</p> <p>Então partindo do pressuposto que eles queriam otimizar esse processo eles iam ganhar agilidade na venda né, podendo ahan... abranger uma área maior de negociação digamos assim. (p02)</p> <p>[...]competitividade na ponta. (p02)</p>	<p>[...]esse analista de Ti, então ele tendo aquele conhecimento ahan... básico né digamos assim,aquele conhecimento fundamental para tomar os rumos né, ele convocava as pessoas responsáveis por cada área, os responsáveis pra direcionar esse detalhamento. (p03)</p> <p>[...] como é um projeto grande né, então ele envolve muitas regras coisas assim, então muitas vezes eu também não, não não... no momento assim eu não consegui sanar todas as</p>	

	<p>dúvidas. (p03)</p> <p>Agora que ta quase finalizando notei que não foi um levantamento full assim né. (p03)</p> <p>Poderia ser um pouco mais aprofundado. (p03)</p> <p>se o analista de TI juntamente com o usuário já tivessem alinhado o que eles queriam, detalhado o que eles queriam melhor, ou seja, bom vamos alinhar, então no levantamento ocorreram muitos conflitos. (p03)</p> <p>[...] e a ferramenta também... com pouco conhecimento que eu tinha da ferramenta. (p03)</p> <p>É com certeza duvidas ahn.. surgiram nesse processo, muitas delas relacionadas as novas regras, regras totalmente novas não contempladas pelo sistema e também duvidas relacionadas a tecnologia ou seja se aquilo estava sendo solicitado ahn.. poderia ser atendido através da ferramenta que eu tinha disponível. (p03)</p> <p>[...] o Analista de TI então assim, ele tinha assim o conhecimento inicial e que por sua vez não era o conhecimento completo então ele próprio tinha muitas duvidas por isso da solicitação da participação do usuário final. (p04)</p> <p>É eles não tinham muita noção tá. (p04)</p> <p>Eu acho que a alteração muitas vezes ocorreu por dúvidas do cliente, no processo. No começo foi definido de uma forma, então durante uma reunião muita coisa foi mudada, então o escopo foi mudado várias vezes nesse sentido, a regra que era, que foi definida no inicio não era a mesma que foi concluída, dada como certa, então acho que esse foi um dos fatores também que levou a mudança e outro com relação ao levantamento que foi muito macro. (p04)</p> <p>Com certeza, tu já tem que ter a visão do todo pra tu ver naquilo que impacta, muitas vezes aquilo que o cliente não tem, que eu acho que é principal função do analista de sistemas aqui é ter a visão mais geral, do todo né, daquilo que possa impactar e com certeza vai da mais qualidade pro sistema né.(p05)</p> <p>Estava explicito, porém nem tudo aquilo que estava explicito era verdade digamos assim. (p04)</p>	<p>Não teria horas o suficiente, então foi delimitado por esse sentido, então a qualidade poderia ter sido melhor, porém foi delimitado um pouco pela quantidade de horas do projeto. (p05)</p>
--	--	---

Entrevistado: Analista G		
Construção da Realidade Inicial	Percepção/Interpretação na Construção da Realidade Inicial	Decisão na Construção da Realidade Inicial
<p>Sim ela partiu do cliente. (p02)</p> <p>Era uma melhoria visando atender um problema que eles tinham de o registro não ser efetuado pelo prestador, e sim pelas secretárias. (p02)</p> <p>Esse projeto ele é bem antigo, ele é ainda do tempo que eu era programador e eu acabei no final das contas assumindo ele como analista. O requisito foi levantado em conjunto com um analista da Totvs na época Datasul, junto com o analista do cliente. (p03)</p> <p>Eu recebi o requisito, enfim ele foi dividido em partes, a parte de documentos a parte de cadastros e a parte de apresentação web. (p03)</p> <p>Não estava adequado. Ele estava bastante trabalhado em um aspecto, por exemplo, a parte da apresentação estava bem trabalhada, a parte de movimentos também de como isso iria impactar lá nos movimentos também estava bem trabalhada, porém a parte de como operacionalizar isso juntou ao prestador, com a secretária, e com o cliente, como iria ser implantado não foi estudado. Então o que aconteceu? nós desenvolvemos todo o sistema de uma forma que funciona plenamente porém da forma como ele foi desenvolvido ficava muito difícil de se implantar e não era exatamente o que o cliente precisava. Ele funciona, mas</p>		

<p>não atendia na época, na primeira versão não atendia completamente o cliente. (p04)</p>	<p>Sim as dúvidas surgiram inicialmente quanto a tecnologia, porque era um aparelho novo, um <i>driver</i> novo, uma forma nova de se trabalhar e foram as mais facilmente solucionadas, apenas com algumas leituras foram solucionadas. Eu creio que a dúvida mais difícil foi entender o real benefício da utilização daquele sistema pelo cliente para então conseguir entender qual seria a melhor forma de se implementar isso no cliente, por que fazer funcionar e deixar redondo sem falhas e rápido não significa que vai ser bom para o cliente. (p04)</p> <p>Foi diretamente com cliente, a única forma de solucionar esse tipo de dúvida foi entrevista direta ao cliente, ele mostrando juntamente com o usuário como funcionaria no dia-a-dia então daí foi que nós começamos a perceber que algumas coisas não seriam tão práticas embora estivesse funcionando, embora estivessem no padrão de qualidade, mas não seriam práticas e nem às vezes tão úteis para o cliente. (p04)</p> <p>Ele sabia o que ele queria, agora a forma... o macro que ele queria ele sabia agora os pormenores que ele foi solicitando para chegar lá nem sempre conduziram até o resultado que ele queria, então nós tivemos as vezes que... (p06)</p>	<p>Na verdade eu já recebi o requisito inicialmente pronto e depois ele teve que ser reformulado, daí eu levantei novamente com o cliente. (p03)</p> <p>Foi bem dinâmico digamos assim, porque conforme iam aparecendo os problemas em conjunto com o cliente mesmo nós íamos já criando as soluções[...].(p05)</p>
--	--	---

Entrevistado: Analista H		
Construção da Realidade Inicial	Percepção/Interpretação na Construção da Realidade Inicial	Decisão na Construção da Realidade Inicial

<p>Não de um problema propriamente dito, na verdade foi uma regulamentação, o órgão que regulamentou um determinado processo dentro do cliente e em função dessa regulamentação alguns processos internos dentro do cliente foram revistos e melhorados. (p02)</p> <p>Sim, eu fiz in loco esse levantamento.... primeiro de tudo foi feito um estudo da legislação, o que a legislação regulamentava enfim né, foi feito um estudo dentro do sistema, o que ele oferecia hoje para atender a regulamentação e foi feita uma avaliação dentro do cliente, in loco, de quais os pontos, não só de sistema mas de processo. [...] então toda a análise foi feita com base em levantamentos não só de uma área né, foi feito o processo como um todo. (p03)</p>	<p>Nós tivemos alguma dificuldade, não na fase final do projeto, mas durante a prototipação que teve um determinado processo que não foi avaliado por completo pela área responsável. (p04)</p>	<p>[...] Qual foi a estratégia do projeto, nós acabamos, eu acabei indo área a área e levantando o que cada área faz no seu dia-a-dia e que envolveria esse projeto. Aí cada área montou uma planilha, montamos em conjunto com a área uma planilha com os itens, então cada área acabou tendo uma descrição de funcionalidades, eu não digo de funcionalidades uma descrição de atividades do seu dia-a-dia e essa ferramenta, ela virou a nossa ferramenta de</p>
---	---	---

	<p>O problema maior que eu tive nesse caso foi que determinada área não mapeou dentro da planilha um determinado fluxo que a gente não sabia que existia e durante a liberação. (p06)</p>	<p>testes na prototipação, então eu trouxe essa planilha para a unidade, para o ambiente de desenvolvimento e nós repassamos o que o setor faz no dia-a-dia depois com o projeto implantado para ver o impacto disso nas áreas[...] (p06)</p>
--	---	---

Entrevistado: Analista I		
Construção da Realidade Inicial	Percepção/Interpretação na Construção da Realidade Inicial	Decisão na Construção da Realidade Inicial
<p>Essa solicitação na realidade ela tinha um problema tá, porque na verdade o que que foi.... esse projeto foi uma reformulação de um projeto que já existia, então na realidade que aconteceu: a primeira vez que foi feito este projeto, foi desenvolvido tudo na qual o cliente pagou e foi homologado ahn.... digamos assim, ficou uma coisa muito complexa para eles apontar de eles não conseguirem padronizar direito ele dentro do sistema, entende? então até um dos motivos por eu escolher ele é isso porque na realidade com essa reformulação dele agente conseguiu adaptar bastante coisa para conseguir deixar ele um pouco mais prático de uma forma funcional para o cliente. (p02)</p> <p>Não era um problema, era uma melhoria. (p02)</p>		

	<p>Não, não estava, tanto que eu tive que exigir algumas reuniões porque realmente o projeto antigo era muito complexo, e só olhando o projeto antigo tu não iria ter uma noção legal e iria acabar desenvolvendo uma coisa que não iria estar conforme a solicitação. (p03)</p> <p>Cara, ele sabia sim uns 90%. Alguma coisa ele acabava pedido para algum usuário que fazia alguma outra parte do sistema, alguma coisa assim porque eu realmente falava com um cara só que era o “cabeça” do projeto. Então através dele eu tirava maior parte das dúvidas. Aquilo que ele não sabia ele acabava perguntando me pedia um tempo e encaminhava por email alguma coisa assim. (p03)</p>	<p>Depois eu vi que existia a viabilidade tirando as dúvidas né... e digamos que ficou um projeto 100% viável. (p06)</p>
--	---	--

Entrevistado: Analista J		
Construção da Realidade Inicial	Percepção/Interpretação na Construção da Realidade Inicial	Decisão na Construção da Realidade Inicial
<p>Não. A solicitação partiu para que houvesse uma melhoria, não era um problema em si. (p02)</p> <p>A princípio foi a partir de FO, chegou o chamado, a gente analisou e depois disso eu troquei uma idéia com o analista do cliente. Fomos conversando dois ou três dias até chegar em um consenso do que seria o projeto. (p03)</p>	<p>Teve, teve uma situação em que foi pedido um campo no cadastro, só que este campo não ficou muito bem definido então eu entendi uma coisa, mas depois não era bem isso, e o usuário não sabia muito bem o que era[...] (p04)</p>	<p>[...] chegamos a ter de conversar por telefone, ligamos para a área solicitante e aí se definiu o que era. (p06)</p>

		Nesse caso sim sempre verifiquei se dava, para depois confirmar para o cliente. (p06)
--	--	---

Entrevistado: Analista K		
Construção da Realidade Inicial	Percepção/Interpretação na Construção da Realidade Inicial	Decisão na Construção da Realidade Inicial
<p>Na verdade não era um problema e sim uma melhoria, porém a falta dessa melhoria digamos assim, impactava fortemente para o cliente e gerava bastante trabalho para os usuários. (p02)</p> <p>Na definição dos requisitos não sei por que eu não realizei essa parte, mas houveram dúvidas no momento do desenvolvimento e essas foram sanadas com o cliente. (p03)</p>	<p>O requisito até estava bem detalhado se compararmos com outros projetos que eu já fiz, mas como esse era um projeto grande eu entendo que faltou alguns detalhes que depois o cliente cobrou, ou seja, para algumas situações estava bem explicado porém para outras faltou detalhar mais pra ficar bem claro. (p03)</p>	<p>Sim, apesar de estarmos utilizando uma tecnologia e uma ferramenta nova sempre conseguimos encontrar uma alternativa. (p05)</p>

Entrevistado: Analista L		
Construção da Realidade Inicial	Percepção/Interpretação na Construção da Realidade Inicial	Decisão na Construção da Realidade Inicial
<p>Inicial eu diria que partiu de um juiz provavelmente, por que foi por intermédio da justiça que o cliente solicitou o desenvolvimento. (p02)</p> <p>Para o cliente não era um problema.</p> <p>Para o cliente devido a essa exigência da lei né, no caso foi aí que o cliente solicitou o desenvolvimento, mas até então não era um problema do cliente e ele estava trabalhando normal com o <i>software</i>. (p02)</p>	<p>Correto. Isso mesmo porque até a gente questionava como que é essa situação? Daí o cliente disse: não isso não existe. Daí a gente foi ver que existia mesmo e assim tiveram que ir atrás de outros chefes de setores para realmente dizer se aquela informação era verdadeira ou não. (p04)</p>	<p>[...] e aí sim através de reuniões a gente foi lendo item a item e detalhando o processo que eles imaginavam que deveria ser feito juntamente com a nossa análise. Aí a gente chegou a uma conclusão. (p03)</p>

Mapa 2 – Colisão

Entrevistado: Analista A	
Expectativas	Experiências
<p>Teve um caso específico eu estou me lembrando, que era uma chamada específica que o cliente queria ter como específico dele na interface nova, que é em flex e isso como o produto não é todo inventado por nós, nós também responderemos a uma área de tecnologia houve alguma falha de comunicação onde se entendeu que era possível e se descobriu na hora de fazer que não era então teve um desgaste que o cliente ficou por muito tempo tratando como um erro, o fato daquela tela específica ter que utilizar no modo antigo, ele comprou um produto moderno só que teve que ter uma perna que andava no passado, lá no elo perdido e aí ficou desgastado. (p14)</p>	<p>Em alguns momentos nós tivemos que abrir o documento de engenharia mesmo, o documento da análise para discutir com o cliente, o cliente não era apenas o usuário final tinham alguns analistas de sistemas lá dentro e então tornou digamos a conversa equilibrada[...].(p08)</p> <p>[...]e aí se buscou caminhos alternativos que alguns contentaram o cliente e outros foi meio que empurrado, ó é assim ou não tem outro jeito e aí começou um pouco de desgaste em alguns momentos. (p12)</p> <p>[...] teve caso de requisito ter sido tirado do sistema porque o cliente idealizou uma coisa, o que foi possível fazer ficou muito complexo no ponto de vista do usuário final e como o cliente, o usuário final seria um usuário limitado, não seria um funcionário do cliente e sim um cliente do cliente e que.... e aí é um universo enorme de possibilidades de pessoas, se optou por tirar fora, tirar essa validade, por que o usuário final não conseguiria usar e aí se criou caminhos alternativos dentro do produto. (p08)</p> <p>Principalmente no tocante à regra de negócio, principalmente por já conhecer o produto em si, o produto que estava sendo descontinuado para se criar o novo, se eu não tivesse participado de projetos anteriores eu estaria naquela situação que eu te falei antes que se for um analista mais novato ele não vai saber, ele não vai ter firmeza para orientar melhor o cliente e nem para decidir de que forma atuar né para atender um requisito. (p20)</p>

Entrevistado: Analista B	
Expectativas	Experiências
Não foi identificado	Não foi identificado

Entrevistado: Analista C	
Expectativas	Experiências
O cliente, ele entendia que era um projeto complexo então ele solicitou alguém que tivesse experiência no sistema para estar acompanhando o próprio levantamento dos requisitos. (p08)	<p>por diversas vezes o requisito, a solicitação do cliente implicava em outras áreas do sistema e precisaram ser modificadas ou adequadas para se enquadrar no processo já existente do sistema, para não gerar inconsistência de dados, gerar informações errôneas, performance. (p18)</p> <p>[...] o cliente ele solicitava da forma como ele desejava, como disse anteriormente, nem sempre se enquadrava dentro do contexto do sistema, mas o cliente possuía um bom conhecimento do funcionamento do sistema e muitas coisas ele mesmo foi no levantamento dele ele já foi definido como poderia ser feito. (p08)</p> <p>[...] nós definíamos o que ia ser feito e como poderia ser feito todo o fluxo de informações dentro do sistema se levantava dúvidas e dizia para o cliente oh! isso eu preciso que tu defina como que tu deseja porque se eu fizer da minha maneira talvez não vai atender a tua necessidade. (p08)</p> <p>Não, dentro do conhecimento do sistema sim.[...] (p10)</p> <p>[...] por diversas vezes foi informado ao cliente que a solicitação dele poderia trazer alguns prejuízos em relação às informações que ele ia ter dentro do sistema, então algumas informações foram levantadas essas situações onde que se essas situações</p>

	<p>ocorressem ele ia ter informações e não muito confiáveis dentro do sistema. (p10)</p> <p>[...] sim todos os casos o cliente aceitou porque com exemplos eu consegui mostrar para ele onde é que estavam os furos, digamos assim os problemas que poderiam causar. (p14)</p> <p>Nesse caso, nesse projeto foi com base nas minhas experiências[...].(p20)</p> <p>[...] todas as situações que poderiam gerar algum... alguma forma de inconformidade, foram levantadas antes mesmo de se desenvolver o projeto então quando elas vieram a ocorrer não poderiam mais ser consideradas inconformidade, o cliente estava ciente que deveria ser daquela forma. (p14)</p> <p>Primeiro na experiência da regra de negócio, no conhecimento de algumas regras de negócio do cliente, ahn... regras que não são padrões do produto que foram somente com o conhecimento desses projetos anteriores eu consegui ter uma visão do que essas regras impactariam nesse projeto ou esse projeto impactaria nessas... (p20)</p> <p>Se eu fosse uma pessoa que não estivesse acostumada a lidar com enfim, com as situações do cliente com a regra de negócio do que cliente, com regra de negócio do sistema, teria sido mais impactante não ter o conhecimento das regras existentes. (p20)</p>
--	---

Entrevistado: Analista D	
Expectativas	Experiências
<p>Sim, estava claro o que ele queria ao menos, agora se eu conversasse com ele eu não sei se ia ser a mesma percepção de quem fez o orçamento. (p09)</p> <p>na minha opinião esse levantamento pela forma como foi proposto a funcionalidade era uma coisa que não iria influenciar no desempenho. (p08)</p>	<p>[...] mas como nesse caso a funcionalidade não tinha nenhum impacto maior no <i>software</i>, ou seja, poderia aplicar ela sem problema algum, sem mexer na regra de negócio, aí não foi questionado o cliente sobre esse desenvolvimento em momento algum. (p08)</p> <p>Sim, sempre contatando o cliente até porque pra ver se isso não impacta em um outro processo dele né. (p13)</p>

Entrevistado: Analista E	
Expectativas	Experiências
<p>Sim, a definição dos requisitos estava clara e adequada à solicitação do cliente, porque o cliente participou ativamente no repasse de informações para que agente pudesse definir e especificar de forma clara e detalhada. Todas as dúvidas eram sanadas de imediato. (p09)</p>	<p>Sim, o sistema está preparado para receber estes novos desenvolvimentos, e isso tornou viável a implementação da demanda solicitada. (p12)</p> <p>Praticamente tudo o que foi solicitado pelo cliente, o sistema poderia atendê-lo de forma específica. Se fosse colocado essa implementação no produto padrão do sistema, quem sabe poderia ocorrer colisões em virtude de que a regra é muito específica do cliente e não atenderia outros clientes, para estar no produto. (p20)</p> <p>Neste caso a negociação foi tranquila. Sempre me preocupo em deixar claro para o cliente que, se o que o ele quer é viável para o sistema, e aceito exatamente como ele coloca. Caso não seja viável, é exposto para o cliente e oferecido outra forma pra atendê-lo. Tudo é negociável para que fique bom para ambos. (p15)</p> <p>Sim, todos os projetos que participo seja pequeno ou grande, procuro agir da mesma forma. O principal para qualquer projeto é o contato inicial com o cliente, deixando esclarecido o que o cliente quer, como sistema atualmente atua e como o sistema vai atuar após a implantação do projeto né. Tanto o cliente quanto o analista tem que estar com o mesmo entendimento. (p20)</p> <p>Sim, com certeza. Já tínhamos conhecimento do perfil do cliente que iria atuar neste projeto e agente sabia da necessidade de descrevermos tudo o que fosse solicitado ou negociado. Isto facilitou muito o andamento do projeto. Projetos passados com pouco detalhe geraram conflitos no decorrer no projeto, pela falta de documentação. (p20)</p>

Entrevistado: Analista F	
Expectativas	Experiências
<p>[...]então muitas vezes eu também não, não no momento assim eu não consegui sanar todas as dúvidas era um processo que o cliente também tinha duvida [...](p08)</p> <p>[...] até porque esse projeto tem uma dependência direta com outro <i>software</i> e esse outro <i>software</i> eles tinham um histórico péssimo, posso dizer por passagem [...](p08)</p> <p>já o levantamento, o detalhamento teve a participação do usuário então o usuário já vem com projeto totalmente completo, detalhado né de ponta a ponta eu preciso disso e disso, ta atendendo ele de uma forma ahh..completa. (p09)</p> <p>Agora questões mais relacionadas a tecnologia ahh..tive um pouco mais de..como posso dizer...dificuldades, para tirar essas duvidas, porque esse conhecimento ele tava ahh.. apenas uma pessoa tinha esse conhecimento, mas nada que pudesse impacta no andamento do projeto. (p09)</p>	<p>[...] se o analista de TI juntamente com o usuário já tivessem alinhado o que eles queriam e detalhado o que eles queriam melhor, então no levantamento ocorreram muitos conflitos ou seja não há inicialmente levantado isso o analista de TI argumentando né com o usuário final,que definido isso e o usuário final tinha entendido. (p08)</p> <p>[...] o usuário final então no levantamento ele não enxergava como um todo então cada usuário enxergava o seu processo, não tinha a visão do todo, então eu acho que nesse sentido o próprio cliente ele tinha muitas duvidas nesse sentido, ele solicitava algo que não tinha noção do impacto que ele poderia provocar, era mais assim oh preciso atender isso pra otimizar o meu processo. Era mais nesse sentido. (p08)</p> <p>Eles não tinham muita noção tá. A não ser quando os usuários das áreas se reuniam, aí muitas vezes acabava ocorrendo que mais dúvidas daquilo e um não entendendo o porquê daquilo. Um não entendia o porquê aquela área fez daquela forma ou tinha que ser assim ou um dizia que tinha que ser assim outro dizendo que não que ia prejudicar e tal..tal..tal..(p08)</p> <p>Muitas coisas que eles solicitaram que era pra atender o processo deles de uma forma plena assim de uma forma completa não pode ser atendido por questões mais técnicas, questões técnicas com relação a ferramenta. (p08)</p>

<p>uma questão, fator complicador do projeto, é que muitas vezes o cliente não via aquilo como fora do escopo né, até porque assim o documento formal, o documento final do levantamento né, onde constavam destacado a questão do que era fora de escopo foi solicitado pelo cliente, pelo analista de TI, que fosse retirado né. Então esse documento quando foi apresentado para o pessoal, para os usuários, então me transpareceu o seguinte que a questão de fora de escopo, não deveria ser tipo levada a parte de coordenação, gerencia e demais usuários, então acho que essa questão foi um fator complicador mesmo né, e até durante o projeto né, muitos dos requisitos foram alterados, novos requisitos foram solicitados e sempre ocorreu essa resistência por parte do cliente, de entender isso. (p11)</p>	<p>É com certeza nós temos aqui na empresa aqui então gerentes de produto tá que eles avaliam a demanda do cliente a necessidade do cliente se ela pode ou não ser ahn.. integrada ao sistema se ela pode ser contemplada ao sistema então nesse sentido. (p15)</p> <p>E outras questões mais técnicas, que o cliente solicitou por questão de usabilidade, digamos assim, ou funcionalidade ou praticidade, então eu pude, digamos limita o que o cliente queria, ta então onde eu tinha um digamos aval maior em pude limitar o cliente. (p15)</p> <p>Até pelo conhecimento meu das regras do próprio sistema, do nosso sistema. Então eu tive que moldar o levantamento conforme o sistema atende né, e também com relação a tecnologia, questões técnicas e claro, considerando esse embasamento, essa questão toda foi levando em consideração a experiência do sistema pela minha parte. (p13)</p> <p>Então assim, muitas coisa que o cliente solicitou mais relacionados a funcionalidade, digamos assim né, questão de telinhas para facilitar isso, eu acabei cortando né, cortando mais o que? Relacionado a como eu já falei a limitações e também a considerar funcionalidades que não vinham agregar nenhum valor ao processo né, no caso iam aumentar o numero de horas do projeto porém não iam agregar funcionalidades né, não iam ter nenhum valor a mais. (p13)</p> <p>Ahn, também questões de lentidão de processos né, então baseado no conhecimento já de outros processos eu pude identificar a questão de lentidão e já tomar algumas ações em cima disso né, mudando muitas vezes o requisito. (p13)</p> <p>O sistema, essa solução ou esse projeto, não tinha como pegar essas informações e inputar elas pra dentro desse outro <i>software</i> certo? Então todo o conhecimento né, eu sabia que coisas novas não existiam nesse sistema e eu não teria como trazer essas informações pra esse sistema. (p20)</p> <p>Assim, a questão de baseado é baseado na experiência em si né, é baseado no teu dia-a-dia</p>
---	---

<p>[...]se a pessoa não tem um total domínio daquilo né, é mais complicado, mas tudo o que eu já fiz nesse levantamento já poderia indicar para o cliente que talvez isso não seria possível por causa disso, disso e daquilo. (p17)</p> <p>O levantamento inicial ele foi feito de uma forma bem completa, algumas coisas tiveram que ter sido renegociadas com o cliente tá, porém durante o projeto varias homologações foram feitas tá, pra ver se aquilo que estava sendo desenvolvido estava dentro do esperado, estava dentro do escopo. (p13)</p> <p>Ocorreram vários contratemplos no sentido mais relacionado ao que? Como o projeto envolvia três equipes né, inclusive uma fora de Caxias, nós tivemos muitos problemas com relação a integração dessas aplicações né, ahn... problemas maiores relacionados a ambientes, ambientes distintos de desenvolvimento, e na hora unificar de sincronizar todos esses ambientes muitos problemas foram encontrados(p13)</p>	<p>de outros projetos que tu já toco né, então conhecimento teu de anos de empresa né.. (p17)</p> <p>[...]tu já tem que ter a visão do todo pra tu ver naquilo que impacta[...].(p20)</p> <p>A questão de escopo, de detalhamento é uma coisa que em outros projetos eu “apanhei” bastante, então detalhar o máximo possível né, identificar a viabilidade de fazer ou não fazer, muitas vezes falava em projetos antigos, dá pra fazer! Aí chegava lá e o da pra fazer dava pra fazer, mas ia levar o dobro de horas estimadas inicialmente, então os projetos antigos eles me auxiliaram nesse sentido né. (p20)</p> <p>Com relação a regras já do nosso sistema aqui onde eu atuo já faz alguns anos, então esse conhecimento veio do meu trabalho diário de outros projetos, parte de documentações, de conversas com outros colegas, discussões. (p26)</p> <p>Sem esse conhecimento o projeto por tamanha complexidade né, se eu tivesse menos conhecimento o projeto poderia ser comprometido né. (p26)</p>
---	---

Entrevistado: Analista G	
Expectativas	Experiências
	<p>Não ficou claro como seria feito o cadastro da biometria do prestador. Foi desenvolvido de forma que o prestador conseguisse fazer seu cadastro a partir do consultório. Depois quando fomos entender o processo como um todo como funcionaria foi visto que isso não seria possível porque? abrindo esta possibilidade ele poderia simplesmente cadastrar um</p>

	<p>dedo dele, um dedo da secretária, um dedo da filha, um dedo da esposa e aí qualquer um poderia registrar no lugar dele. (p08)</p> <p>[...] então nós simplesmente dissemos que isso aqui vai ter que ser dessa forma as questões tecnológicas que não permitem... (p13)</p> <p>Sim, com certeza porque houve bastante contato com o cliente e creio que se fosse... se não fosse bem conduzido ali os testes inclusive quando houve discordância dos requisitos o projeto poderia ter simplesmente afundado. (p18)</p> <p>Ahn... sim, na verdade era um dos primeiros projetos como analista, mas por isso que foi bastante novo para mim, mas a minha experiência como programador ajudou bastante no ponto de chegar e dizer para o cliente: Não, tecnologicamente isso aqui não dá para fazer, isso aqui da. (p20)</p>
--	--

Entrevistado: Analista H	
Expectativas	Experiências
	<p>[...] como o sistema ele envolveu mais de uma área simultânea dentro do cliente, cada área pensou na sua realidade então tiveram soluções que partiram do próprio cliente de processo que atendeu a área que aquele usuário trabalha, mais que em determinado momento essa solução ela impactou em o segundo setor, então a gente teve que interagir e reunificar os dois setores e montar uma reunião para alinhar esse entendimento, porque mais que via processo tenha atendido uma área a outra ficou prejudicada. (p09)</p> <p>Foi, basicamente foi porque quando foi levantado esse problema o setor já levantou o problema e já estava com a solução definida né, como não envolvia sistema era só processo puro, mas aí era uma situação que eu já tinha trabalhado em outro projeto e aí a gente conseguiu juntar os dois setores para discutir a melhor solução. (p20)</p> <p>Eu acho que se eu não tivesse vivenciado algumas outras situações talvez eu teria demorado um pouco mais para identificar algumas soluções entende. (p20)</p>

Entrevistado: Analista I	
Expectativas	Experiências
<p>[...]eles pediram para lá no botão da consulta ter uma nova função que quando eu clicasse lá já fosse direto para aquela informação, essa foi uma das questões que houve. (p15)</p>	<p>eu acho que measurei muito bem as horas assim, ficou bem... se estourou, estourou muito pouco. (p19)</p> <p>Primeiramente eu sempre parto de tentar achar a solução por mim mesmo, então na realidade através do meu conhecimento, das buscas que eu tenho dentro do sistema, de documentos ou daquilo que está especificado pelo cliente, às vezes naquilo que está especificado já tem uma parte da resolução né. Eu sempre parto a partir deste ponto. (p20)</p>

Entrevistado: Analista J	
Expectativas	Experiências
<p>Teve, teve uma situação em que foi pedido um campo no cadastro, só que este campo não ficou muito bem definido então eu entendi uma coisa, mas depois não era bem isso, e o usuário não sabia muito bem o que era, chegamos a ter de conversar por telefone.</p> <p>para a área solicitante e aí se definiu o que era, mas houve sim. (p09)</p>	<p>Teve, teve uma atitude tomada para melhorar a usabilidade, porque o cliente gostaria de fazer um controle por prestadores e isso ele imaginava de um jeito. Ele imaginava fazer lá em um arquivo de inicialização e eu mudei o escopo do projeto, eu criei um cadastro específico para controlar tudo que ele queria a partir desse cadastro. (p18)</p> <p>Sim. Nesse caso sim sempre verifiquei se dava, para depois confirmar para o cliente. (p12)</p> <p>É na experiência mesmo, desse jeito não vai funcionar como o cliente quer de fato. Foi então que eu sugeri, vamos fazer assim? E o cliente aceitou muito bem a idéia.(p12)</p> <p>[...] ele imaginava de uma forma e essa forma era inviável da gente fazer. (p13)</p> <p>[...] era outro projeto também Web em que houve muita dificuldade no desenvolvimento.</p>

	<p>Então a gente tomou como lição o primeiro para não errar nos próximos. (p20)</p> <p>Ajudou sim, alguma dúvida que eu tinha. O cliente me pediu isso, cliente o que é esse campo? porque tem muitos clientes que tu pede e ele não sabe o que é. E esse cliente fez todo o levantamento.... então eu pedia o que é esse campo cliente? aí o analista me falava isso é para fazer isso e isso e isso, definia o que era. Existem alguns casos que tu liga e pergunta cliente o que tu quer? Ah mas eu não sei muito bem, foi a área que pediu isso...(p22)</p>
--	---

Entrevistado: Analista K	
Expectativas	Experiências
<p>Olha, o requisito até estava bem detalhado se compararmos com outros projetos que eu já fiz, mas como esse era um projeto grande eu entendo que faltou alguns detalhes que depois o cliente cobrou, ou seja, para algumas situações estava bem explicado porém para outras faltou detalhar mais pra ficar bem claro. Só que isso eu só fui descobrir depois, nas homologações com o cliente. (p09)</p> <p>Na definição dos requisitos eu não sei porque eu não realizei essa parte, mas houveram dúvidas no momento do desenvolvimento e essas foram sanadas com o cliente. (p08)</p> <p>Bom, eu acredito que sim, ele sabia exatamente o que ele queria, já penso que ele não soube explicar tudo[...] (p11)</p>	<p>Eu acredito que no geral os clientes entendem e preferem que exista esse contato, claro sem fazer perguntas básicas porque senão ele vai pensar que você não conhece o sistema e pode ser pior. (p13)</p> <p>Ambíguo não, mas incompleto sim. Como eu falei existia uma parte do processo que poderia ter sido mais bem detalhada. (p09)</p> <p>Teve um caso, estou me lembrando agora que a ferramenta não nos permitia construir a tela exatamente como o cliente queria[...] (p12)</p> <p>Foi. Tinha uma parte do requisito que era relacionada a uma alteração em outra parte do produto e isso precisou ser retirado da proposta porque não cabia estar sendo desenvolvido por nós naquele momento[...] (p13)</p>

	<p>ocorreu um caso que ele queria que inseríssemos alguns status que o sistema não contemplava. Nesse caso eu sabia que se eu simplesmente utilizasse os novos status que o cliente pediu eu estaria gerando problemas para os demais clientes porque eles não conheciam esses status, eu estaria alterando o padrão. E nesse caso contatei o cliente e coloquei a situação pra ele. (p20)</p> <p>Com certeza. Teve influencia direta no projeto. Com essas experiências passadas eu pude compreender pra que o cliente utilizaria aquilo que ele estava solicitando. Me ajudou muito, porque se eu não tivesse esse conhecimento eu iria encontrar muito mais dificuldades. (p20)</p>
--	--

Entrevistado: Analista L	
Expectativas	Experiências
<p>[...] na hora que eles colocaram para nós o projeto eles não lembravam que tinham aqueles três ou quatro casos lá que era uma outra situação. Tinha um caso também que foi encontrado que tinham uns vinte usuários que em um determinado período. (p08)</p>	<p>[...] Só que agora que nós precisamos do histórico aí identificamos isso[...].(p08)</p> <p>Foi um projeto que só tinha impacto nele mesmo, que era o setor de faturamento. Então tudo que a gente incrementou não existia, então o impacto foi dele mesmo. O resultado foi que outros setores estavam interferindo nesse projeto, no nosso resultado. (p08)</p> <p>Sim, foram identificados alguns casos que a gente teve que adaptar e no início a gente não consegui identificar. Mas assim foram coisas que não teve interferência no projeto, foram detalhes, mas que existiram, existiram. (p13)</p> <p>[...]Eu acho que com as reuniões eles entenderam, que a gente estava detalhando o máximo, se não consegui detalhar foi uma situação isolada. (p13)</p> <p>Sim, até porque como a gente já trabalhava com o cliente então nós temos vários projetos com o cliente e eu acho que isso facilitou né. Com o módulo também eu tinha trabalhado em outros projetos com esse módulo, e eu acho que isso facilitou bastante em termos de parametrização eu já tinha conhecimento então eu acho que tudo isso agregou. (p20)</p>

Mapa 3 – Negociação

Entrevistado: Analista A			
Construção da Realidade Secundária	Percepção/Interpretação na Construção da Realidade Secundária	Adesão na Construção da Realidade Secundária	Abandono na Construção da Realidade Secundária
<p>Olha, a questão de tecnologia nós procuramos envolver o cliente o mínimo possível até porque ele queria ver o produto final com funções maravilhosas que desejou que ele sonhou. Então a questão de tecnologia nós tivemos muitas discussões internas, como eu te falei como nós temos uma área de</p>	<p>[...] só que quando tu faz uma pesquisa inteligente, tem que ter um banco de dados muito rápido, tem que ter pesquisas refinadas né, enfim, e a tecnologia usada não previa isso, não previa paginação por exemplo, em telas com muita informação e isso quebrou o processo na hora de usar. (p18)</p> <p>[...] que deu discussão de os dois lerem o mesmo texto, é como interpretar uma lei né, às vezes tem só um parágrafo ali eu entendo que é uma coisa e tu entende que é outra diferente e aí cai em discussão. (p09)</p> <p>[...]e aí se buscou caminhos alternativos que alguns contentaram o cliente e outros foi meio que empurrado, ó é assim ou não tem outro jeito e aí começou um pouco de desgaste em alguns momentos. (p12)</p>	<p>Se tentou, se dialogou no início, mas nós acabamos meio que impondo uma solução porque foi o que tecnicamente era viável e ficou dentro... não era o que o cliente queria no começo mas atendeu o que ele precisava. (p16)</p>	

tecnologia que não fica aqui na cidade, teve algum conflito até de discussão, conflito de idéias, por que era tudo novo né, era um framework novo, e o primeiro cliente grande a receber esse produto era o nosso, as outras unidades que atuavam, digamos estavam gatinhando no projeto e nós tínhamos cliente com data prometida para entrar, então deu muita discussão. (p16)			
--	--	--	--

Entrevista Analista B			
Construção da Realidade Secundária	Percepção/Interpretação na Construção da Realidade Secundária	Adesão na Construção da Realidade Secundária	Abandono na Construção da Realidade Secundária
	Quando identifiquei assim eu recorri a coordenação que...a gente tentou achar uma forma de achar alguém que conhecia então quem foi? Foi a pessoa, foi pessoa não foi documento, foi a pessoa que fez a parte um. Então ela que sentou do meu lado e aí agente conseguiu fechar o que faltava. (p16)		

Entrevistado: Analista C			
Construção da Realidade Secundária	Percepção/Interpretação na Construção da Realidade Secundária	Adesão na Construção da Realidade Secundária	Abandono na Construção da Realidade Secundária
Não, foram após o desenvolvimento inicial ahn... o cliente solicitou melhorias, mas		Sim ele assumiu o risco dessa, dessas situações acontecerem. (p17)	

<p>não que o projeto tivesse alterações de escopo. Com a utilização foram solicitadas melhorias, avanços no conceito, situações novas passaram a ser tratadas. (p13)</p> <p>Aconteceu por algumas vezes, e em algumas situações que nós não tínhamos alternativa o cliente assumiu o risco de realizar da maneira como o processo dele estava sendo feito. (p17)</p>			
--	--	--	--

Entrevistado: Analista D			
Construção da Realidade Secundária	Percepção/Interpretação na Construção da Realidade Secundária	Adesão na Construção da Realidade Secundária	Abandono na Construção da Realidade Secundária
		<p>Mas como nesse caso a funcionalidade não tinha nenhum impacto maior no <i>software</i>, ou seja, poderia aplicar ela sem problema algum, sem mexer na regra de negócio, aí não foi questionado o cliente sobre esse desenvolvimento em momento algum. (p13)</p>	

Entrevistado: Analista E			
Construção da Realidade Secundária	Percepção/Interpretação na Construção da Realidade Secundária	Adesão na Construção da Realidade Secundária	Abandono na Construção da Realidade Secundária
Sempre foi realizado o esclarecimento com os usuários que tinham domínio do assunto que gerou dúvida ou a confusão. (p17)	Neste caso a negociação foi tranquila. Sempre me preocupo em deixar claro para o cliente que se o que o ele quer é viável para o sistema, é aceito exatamente como ele coloca. Caso não seja viável, é exposto para o cliente e oferecido outra forma pra atendê-lo. Tudo é negociável para que fique bom para ambos. (p15)		

Entrevistado: Analista F			
Construção da Realidade Secundária	Percepção/Interpretação na Construção da Realidade Secundária	Adesão na Construção da Realidade Secundária	Abandono na Construção da Realidade Secundária
[...] pegar três opiniões né, do analista de TI no caso a minha opinião e juntamente com o usuário, porém assim o que valeu digamos assim, o que...a opinião que mais valia era do usuário final. (p17) Então muitas vezes eu participei, tu tem que participar dessas discussões e até contribuir assim pra que fosse finalizado esse processo. (p17)			

<p>É questões de regra então, o que podia argumentar o que, baseado</p>	<p>foi detalhado o requisito, porém poderia ter sido melhor detalhado caso tivesse ocorrido esse alinhamento e se eu tivesse conhecimento um pouco maior desse processo. (p18)</p>	<p>Bom, então muitas coisas então, digamos assim, não posso dizer impostas, mas tudo o que era aquilo relacionada ao sistema atual, teve que ser de uma forma porque é assim atualmente, então teve que ser definido aquele processo. Outras questões foram levadas a própria coordenação a própria gerência do cliente que teve que definir, não vai ser assim porque é assim e ponto. (p16)</p> <p>Muitas coisas que eles solicitaram que era pra atender o processo deles de uma forma plena assim de uma forma completa onde o processo bem otimizado, não pode ser atendido por questões mais técnicas, questões técnicas com relação a ferramenta. (p17)</p> <p>[...] a participação do usuário, então ele passou todo o processo dele, então muita coisa eu não pude argumentar, não pude contestar aquilo porque é o processo dele do dia a dia, e como ele ia enxergar daquela forma, tinha que ser tratado daquela forma. (p17)</p> <p>Outras questões mais técnicas, que o cliente solicitou por questão de usabilidade, digamos assim, ou funcionalidade ou praticidade, então eu tive argumentos então né, pra digamos corta né, digamos limita ao que o cliente queria, então onde eu tinha um digamos aval maior em pude limitar o cliente. (p17)</p>	
---	--	--	--

<p>nos requisitos macros né, aquilo que estava muito fora digamos assim, aquilo que não tinha como estar contemplado nos requisitos, foram considerados como fora de escopo, isso eu pude já expor no momento do levantamento. (p13)</p> <p>[...]Ahn, também questões de lentidão de processos né, então baseado no conhecimento já de outros processos eu pude identificar a questão de lentidão e já tomar algumas ações em cima disso né, mudando muitas vezes o requisito. (p18)</p> <p>Assim, a questão de baseado é baseado na experiência em si né, é baseado no teu dia-a-dia de outros projetos que tu já fez né, então conhecimento teu de anos de empresa né.. Então aquilo que o cara já tem uma convicção maior, o cara já toma. Se a pessoa não tem um total domínio daquilo né, tu já já...tudo o que eu já fiz nesse levantamento já indicar para o cliente que talvez isso não seria possível por causa disso, disso e daquilo. (p17)</p>	<p>Outras questões eu tive que trazer pra empresa tá, e discutir juntamente com coordenação, gerente de produto, pra avaliar se aquilo era viável ou se aquilo poderia ser considerado como fora do escopo. Aham.. outros, outros requisitos também foram levantados como específicos do cliente[...].(p13)</p>	<p>Então assim, muitas coisa que o cliente solicitou mais relacionados a funcionalidade, digamos assim né, questão de telinhas para facilitar isso e aquilo, eu acabei cortando né[...].(p16)</p>	
--	---	---	--

Entrevistado: Analista G			
Construção da Realidade Secundária	Percepção/Interpretação na Construção da Realidade Secundária	Adesão na Construção da Realidade Secundária	Abandono na Construção da Realidade Secundária
<p>[...] depois quando fomos entender o processo como um todo como funcionaria foi visto que isso não seria possível porque abrindo esta possibilidade ele poderia simplesmente cadastrar um dedo dele, um dedo da secretária, um dedo da filha, um dedo da esposa e aí qualquer um poderia registrar no lugar dele. (p12)</p>	<p>Nesse caso reagiu bem, não é o que acontece sempre. Nesse caso reagiu bem e nós simplesmente notamos juntos, foi a percepção em conjunto tanto do cliente quanto nossa que isso não,.. não funcionaria e foi partido para outra situação. (p14)</p>	<p>Houve uma negociação, nós acabamos aderindo a necessidade do cliente e parte do custo foi arcado pela empresa e outra parte do custo pelo cliente. (p15)</p> <p>[...] houve também uma situação que o cliente solicitou para fazer de uma forma e que não foi possível. Então nós simplesmente dissemos que isso aqui vai ter que ser dessa forma as questões tecnológicas que não permitem... (p15)</p> <p>Conseguiu porque o cliente era bastante esclarecido esse analista no caso[...].(p17)</p>	

Entrevistado: Analista H			
Construção da Realidade Secundária	Percepção/Interpretação na Construção da Realidade Secundária	Adesão na Construção da Realidade Secundária	Abandono na Construção da Realidade Secundária
<p>[...] mas aí era uma situação que eu já tinha trabalhado em outro projeto e aí a gente conseguiu juntar os dois setores para discutir a melhor solução. (p20)</p> <p>Aconteceram, aconteceram... nós tínhamos na verdade em determinado momento a definição inicial era seguir uma determinada codificação que era o que essa normativa regulamentava né, e na verdade nós chegamos a prototipar com essa codificação e no teste agente viu que não poderia seguir ela em 100%. (p13)</p>		<p>A própria legislação deixou falhas de interpretação e ela mesmo permitia ter uma outra solução para esse caso. Então nós tivemos que mudar. (p16)</p> <p>Entendeu. Assim, não tivemos nenhum problema de definição de projeto por causa do cliente, mas essa definição ela foi entendida pelo cliente por que boa parte dessa definição era por parte deles, seriam espécies de negociações, coisas que eu não consigo colocar dentro do sistema, ele tinha particularidades em que envolviam processos que eram desse cliente e ele sabe que isso é particularidade dele. (p14)</p>	

<p>[...] então teve determinado ponto que eu precisava de uma funcionalidade no sistema, ou de desenvolvimento específico, uma listagem ali então eu abri essa demanda junto à fábrica aqui e um outro analista tocava aquela necessidade. (p17)</p>			
--	--	--	--

Entrevistado: Analista I			
Construção da Realidade Secundária	Percepção/Interpretação na Construção da Realidade Secundária	Adesão na Construção da Realidade Secundária	Abandono na Construção da Realidade Secundária
<p>Existiu na realidade assim, como o nosso sistema é bem assim, a linguagem é em telas... é um processo bem procedural, não é aquela coisa bem visível, não é uma usabilidade tão boa né, então teve questões que ele solicitou que digamos para tu alterar o registro no sistema, tu tinha que dar 40 "enters" para chegar até na informação que tu queria alterar[...].(p13)</p>			<p>na realidade eu ia manter dessa forma, mas eles pediram para lá no botão da consulta ter uma nova função que quando eu clicasse lá já fosse direto para aquela informação, essa</p>

			<p>foi uma das questões que houve. (p15)</p> <p>Mas claro, às vezes gente pensa de uma forma muito técnica né, digamos a gente fez a nossa forma, mas para o usuário foi muito melhor a idéia dele do que ter mantido do mesmo jeito que estava. (p17)</p>
--	--	--	--

Entrevistado: Analista J			
Construção da Realidade Secundária	Percepção/Interpretação na Construção da Realidade Secundária	Adesão na Construção da Realidade Secundária	Abandono na Construção da Realidade Secundária
<p>É na experiência mesmo, desse jeito não vai funcionar como o cliente quer de fato. Foi então que eu sugeri, vamos fazer assim? E o cliente aceitou muito bem a idéia. (p20)</p>		<p>Nesse caso entendeu. É claro ele fez alguns questionamentos, e aí eu tive que explicar, e justifiquei por que era melhor fazer daquele jeito e aí aceitou sem problemas. (p15)</p>	

Entrevistado: Analista K			
Construção da Realidade Secundária	Percepção/Interpretação na Construção da Realidade Secundária	Adesão na Construção da Realidade Secundária	Abandono na Construção da Realidade Secundária
<p>[...] mas para não inviabilizar o projeto agente construiu uma outra um pouco fora do padrão para atender aquela necessidade do cliente[...].(p17)</p> <p>[...] e nesse caso contatei o cliente e coloquei a situação pra ele. (p17)</p>		<p>Entendeu, eu justifiquei e aí ele entendeu. (p17)</p>	

Entrevistado: Analista L			
Construção da Realidade Secundária	Percepção/Interpretação na Construção da Realidade Secundária	Adesão na Construção da Realidade Secundária	Abandono na Construção da Realidade Secundária
<p>Sim, foram identificados alguns casos que a gente teve que adaptar e no início a gente não conseguiu identificar. Mas assim foram coisas que não teve interferência no projeto, foram detalhes, mas que existiram, existiram. (p20)</p>		<p>Ele entendeu. Só que claro, o cliente quer que a gente pense em um todo, mas ele entendeu a situação. E entendeu que era um caso avulso, acho que foi tranquilo. (p14)</p>	

APÊNDICE C – Instrumento de Pesquisa

Chrome

ey=dF9QREFPRWtzV2J4bU9DShVNNWl0emc6MQ

Pesquisa Qualitativa - Qualidade de Software

As perguntas a seguir buscam identificar se o projeto entregue apresenta características da qualidade do software baseado na ISO/IEC 9126-1 - Engenharia de software - Qualidade de produto

Lembro que as respostas aqui concedidas não serão identificadas no trabalho final!

***Obrigatório**

Respondente: *

Projeto: *

Funcionalidade

Quanto a Funcionalidade responde: *
O conjunto de funções satisfazem as necessidades explícitas e implícitas para a finalidade a que se destina o produto?

Sim
 Não
 Não se Aplica

Ainda sobre a funcionalidade responda as perguntas abaixo quanto as suas subcaracterísticas:

Quanto a Adequação: O sistema propõe-se a fazer o que é apropriado? *

Sim
 Não
 Não se Aplica

Quanto a Acurácia: Gera resultados corretos ou conforme acordados? *

- Sim
- Não
- Não se Aplica

Quanto a Interoperabilidade: É capaz de interagir com os sistemas especificados? *

- Sim
- Não
- Não se Aplica

Quanto a Segurança de Acesso: Evita acesso não autorizado, acidental ou deliberado a programas e dados? *

- Sim
- Não
- Não se Aplica

Quanto a Conformidade: Está de acordo com normas e convenções previstas em leis e descrições similares? *

- Sim
- Não
- Não se Aplica

[Continuar »](#)

Tecnologia [Google Docs](#)

[Denunciar abuso](#) - [Termos de Serviço](#) - [Termos Adicionais](#)

Pesquisa Qualitativa - Qualidade de Software

*Obrigatório

CONFIABILIDADE

Quanto a Confiabilidade responde: *

O desempenho se mantém ao longo do tempo e em condições estabelecidas?

- Sim
- Não
- Não se Aplica

Ainda sobre a confiabilidade responda as perguntas abaixo quanto as suas subcaracterísticas:

Quanto a Maturidade: Com que frequência apresenta falhas? *

- Alta
- Baixa
- Não se Aplica

Quanto a Tolerância a Falhas: Ocorrendo falhas, como ele reage? *

- Bem
- Mal
- Não se Aplica

Quanto a Recuperabilidade: É capaz de recuperar dados após uma falha? *

- Sim
- Não
- Não se Aplica

Tecnologia [Google Docs](#)

[Denunciar abuso](#) - [Termos de Serviço](#) - [Termos Adicionais](#)

Pesquisa Qualitativa - Qualidade de Software

*Obrigatório

USABILIDADE

Quanto a Usabilidade responda: *

É fácil de utilizar o software?

- Sim
 Não
 Não se Aplica

Ainda sobre a usabilidade responda as perguntas abaixo quanto as suas subcaracterísticas:

Quanto a Inteligibilidade: É fácil entender os conceitos utilizados? *

- Sim
 Não
 Não se Aplica

Quanto a Apreensibilidade: É fácil aprender a usar? *

- Sim
 Não
 Não se Aplica

Quanto a Operacionalidade: É fácil de operar e controlar a geração? *

- Sim
 Não
 Não se Aplica

Quanto a Eficiência responda: *

Os recursos e os tempos utilizados são compatíveis com o nível de desempenho requerido para o produto?

- Sim
 Não
 Não se Aplica

Quanto ao Comportamento em Relação ao Tempo: Qual é o tempo de resposta e de processamento? *

- Alto
- Baixo
- Não se Aplica

Quanto ao Comportamento em Relação aos Recursos: Quanto recurso utiliza? *

- Muito
- Pouco
- Não se Aplica

Tecnologia [Google Docs](#)

[Denunciar abuso](#) - [Termos de Serviço](#) - [Termos Adicionais](#)

Pesquisa Qualitativa - Qualidade de Software

*Obrigatório

MANUTENIBILIDADE

Quanto a Manutenibilidade responde: *

Há facilidade para correções atualizações e alterações?

- Sim
 Não
 Não se Aplica

Ainda sobre a manutenibilidade responda as perguntas abaixo quanto as suas subcaracterísticas:

Quanto a Analisabilidade: É fácil encontrar uma falha quando ocorre? *

- Sim
 Não
 Não se Aplica

Quanto a Modificabilidade: É fácil modificar e remover defeitos? *

- Sim
 Não
 Não se Aplica

Quanto a Estabilidade: Há grandes riscos de bugs quando se faz alterações? *

- Sim
 Não
 Não se Aplica

Quanto a Testabilidade: É fácil testar quando se faz alterações? *

- Sim
 Não
 Não se Aplica

[« Voltar](#) [Continuar »](#)

Tecnologia [Google Docs](#)

[Denunciar abuso](#) - [Termos de Serviço](#) - [Termos Adicionais](#)

Pesquisa Qualitativa - Qualidade de Software

*Obrigatório

PORTABILIDADE

Quanto a Portabilidade responde: *

É possível utilizar o produto em diversas plataformas com pequeno esforço de adaptação?

- Sim
 Não
 Não se Aplica

Ainda sobre a portabilidade responda as perguntas abaixo quanto as suas subcaracteristicas:

Quanto a Adaptabilidade: É fácil adaptar a outros ambientes sem aplicar outras ações ou meios além dos fornecidos para esta finalidade no software considerado? *

- Sim
 Não
 Não se Aplica

Quanto a Capacidade para ser Instalado: É fácil instalar em outros ambientes? *

- Sim
 Não
 Não se Aplica

Quanto a Capacidade para Substituir: É fácil substituir por outro software? *

- Sim
 Não
 Não se Aplica

Quanto a Conformidade: Está de acordo com padrões ou convenções de portabilidade? *

- Sim
 Não
 Não se Aplica

[« Voltar](#) [Continuar »](#)

Chrome

ormkey=dF9QREFPRWtzV2J4bU9DSHVNNWI0emc6MQ&theme=0AX42CRMsmRFbUy03NTAzM2Q4My03ODU1LTQ2NzIt

Pesquisa Qualitativa - Qualidade de Software

*Obrigatório

Para Finalizar...

Responda com relação ao projeto

Na definição do requisito, você teve a sensação de estar sendo entendido? *

- Sim
 Não
 Não se Aplica

O projeto entregue ficou como você havia solicitado, ou seja, atendeu os requisitos? *

- Sim
 Não
 Não se Aplica

Deixe seu comentário sobre o que achar importante:

[⏪ Voltar](#) [Enviar](#)

Tecnologia [Google Docs](#)

[Denunciar abuso](#) - [Termos de Serviço](#) - [Termos Adicionais](#)