

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS  
MESTRADO EM CIÊNCIA DA INFORMAÇÃO

**Antonio Tupinambá Timbira de Oliveira Pinto Filho**

**AS CONTRIBUIÇÕES DA COMUNICAÇÃO E DO  
CONHECIMENTO DA CIÊNCIA DA INFORMAÇÃO PARA A  
ANÁLISE DE REQUISITOS NO DESENVOLVIMENTO DE  
SOFTWARE.**

CAMPINAS  
2005

**Antonio Tupinambá Timbira de Oliveira Pinto Filho**

**AS CONTRIBUIÇÕES DA COMUNICAÇÃO E DO  
CONHECIMENTO DA CIÊNCIA DA INFORMAÇÃO PARA A  
ANÁLISE DE REQUISITOS NO DESENVOLVIMENTO DE  
SOFTWARE.**

Dissertação apresentada ao Programa de Pós Graduação  
em Ciência da Informação, da Pontifícia Universidade  
Católica de Campinas, como parte dos requisitos para  
obtenção do título de Mestre em Ciência da Informação.

Área de concentração: Administração de informação.

Linha de pesquisa: produção e disseminação de  
informação

Orientador:

Prof. Dr. Raimundo Nonato Macedo dos Santos

Orientando:

Antonio Tupinambá Timbira de O Pinto Filho

CAMPINAS  
2005

## **AGRADECIMENTOS**

Ao professor **Raimundo Nonato**, pela sua orientação segura e incentivante, sempre aberto a aprender e ensinar, contribuindo para o bom resultado deste trabalho.

À professora **Nair Kobashi** da PUCC, pelas inúmeras trocas de idéias e experiências, fator fundamental para se atingir o término deste trabalho.

À **Lúcia Guimarães**, grande companheira e incentivadora deste trabalho, pelas horas de convivência roubadas e por sua carinhosa compreensão.

A todas as outras pessoas que direta ou indiretamente contribuíram para que este trabalho alcançasse sua forma final.

## RESUMO

PINTO FILHO, Antonio Tupinambá T. de O, **As Contribuições da Comunicação e do Conhecimento da Ciência da Informação para a Análise de Requisitos no desenvolvimento de software**. Campinas, 2005, 119p, Dissertação (Mestrado em Ciência da Informação) – Programa de pós-graduação em Ciência da Informação, Pontifícia Universidade Católica de Campinas.

*Esta pesquisa busca compreender, na etapa de definição do sistema, os problemas de valorização, prospecção e formalização da informação, focando os aspectos da comunicação, e suas diversas etapas, entre os atores principais de um projeto de desenvolvimento de software na fase de análise de requisitos: analista e usuário. Um modelo de análise de requisitos que considera as questões comunicacionais e do conhecimento da Ciência da Informação, na formalização das especificações do sistema é apresentado. Este modelo ajuda a identificar os aspectos mais importantes, que estão na esfera do domínio de informação do usuário, de modo a serem externalizados e formalizados para que se consiga chegar a uma definição de requisitos mais próxima da realidade.*

Palavras-chave: Análise de Requisitos, Engenharia de Software, Gestão do conhecimento, Paradigmas da Engenharia de Software, Informação, Comunicação, Linguagem, Interação, Ciência da Informação, Representação do conhecimento.

## ABSTRACT

PINTO FILHO, Antonio Tupinambá T. de O, **The contributions of Communication and Knowledge of Information Science for Requirements Analysis at the software development**. Campinas, 2005, 119p, Dissertation (Master Degree in Information Science) – Information Science graduation program, Pontifícia Universidade Católica de Campinas.

*This research intends to understand, in the definition system phase the valuable, prospective and formalization problems of information. The communication aspects are the focus, with their phases, between the main actors of the software development project in the requirement analysis step: analysts and users. A model of requirement analyses which considerer the communication and Information Science knowledge issues in the formalization of the system specifications are presented. This model helps to identify the important aspects, in the information user domain, for externalized and formalized to reach a requirement definition as near as possible of reality.*

Keywords: Requirements Analysis, Software Engineering, Knowledge Management, Software Engineering Paradigms, Information, Communication, Language, Interaction, Information Science, Knowledge Representation..

## SUMÁRIO

SUMÁRIO .....	5
INTRODUÇÃO .....	8
ASPECTOS COMUNICACIONAIS .....	14
ASPECTOS METODOLOGICOS DA PESQUISA.....	19
CAPÍTULO 1 – ENGENHARIA DE SOFTWARE E CIÊNCIA DA INFORMAÇÃO: CONCEITOS FUNDAMENTAIS.....	23
1.1 O PROCESSO DE DESENVOLVIMENTO DE SISTEMAS .....	27
1.2 O PROCESSO DE ANÁLISE E IDENTIFICAÇÃO DE REQUISITOS .....	30
1.2.1 Pontos críticos na análise de requisitos.....	32
1.2.2 Técnicas de comunicação na análise de requisitos.....	33
1.3 GERAÇÃO DE CONHECIMENTO NA ETAPA DE ANÁLISE DE REQUISITOS .....	35
1.4 O PROCESSO DE DOCUMENTAÇÃO NA ETAPA DE ANÁLISE DE REQUISITOS.....	40
1.5 PADRONIZAÇÃO DE TERMINOLOGIAS NA ANÁLISE DE REQUISITOS .....	45
1.6 CONHECIMENTO PRÉVIO PARA ANÁLISE DE REQUISITOS .....	48
1.7 AS FUNÇÕES DA LINGUAGEM .....	50
1.7.1 Conotações .....	51
1.8 COMUNICAÇÃO HUMANA .....	54
1.8.1 A fórmula de Lasswell .....	55
1.8.2 Comunicação humana e personalidade .....	57
1.8.3 Obstáculos à Comunicação humana .....	59
1.8.3.1 Obstáculos devidos à personalidade.....	59
1.8.3.2 Obstáculos devidos à linguagem.....	61

CAPÍTULO 2 – PROCESSO DE REQUISITOS: ESTUDO DA ANÁLISE DE REQUISITOS SOB O PONTO DE VISTA DA CIÊNCIA DA INFORMAÇÃO E DA ENGENHARIA DE SOFTWARE.....	64
2.1 ANÁLISE DE REQUISITOS NA ENGENHARIA DE REQUISITOS .....	65
2.1.1 Conhecimento necessário aos analistas de sistemas para análise de requisitos .....	66
2.1.2 Participantes chaves no processo de definição de requisitos.....	67
2.1.3 Problemas mais comuns na identificação de requisitos .....	70
2.1.3.1 Comunicação Pobre.....	70
2.1.3.2 Resistência .....	71
2.1.3.3 Articulação e especialidade para problemas.....	71
2.1.3.4 Perspectivas diferentes para o problema .....	72
2.2 ANÁLISE DE REQUISITOS COM O FOCO NO USUÁRIO .....	73
2.3 TÉCNICAS PARA MELHORAR A ANÁLISE DE REQUISITOS .....	76
2.3.1 Características dos requisitos para garantia de qualidade .....	76
2.3.2 Comunicação Efetiva e coleta de informações.....	77
2.3.3 Modelos de representação gráfica que buscam interação com o usuário.....	79
2.3.4 Técnicas de participação do cliente-usuário.....	80
2.3.5 Etnografia Rápida e Inquisições Contextuais.....	80
2.4 DISCUSSÃO DOS PARADIGMAS DA ENGENHARIA DE SOFTWARE EM RELAÇÃO A INTERAÇÃO COM O USUÁRIO – ASPECTOS COMUNICACIONAIS	82
2.4.1 Modelo Cascata.....	82
2.4.2 Modelo em V .....	84
2.4.3 Prototipação .....	86
2.4.4 Modelo Espiral.....	89
2.4.5 Interação Humano Computador.....	91

CAPÍTULO 3 – PROPOSTA DE UM MODELO DE ANÁLISE DE REQUISITOS BASEADO NAS TEORIAS DA CIÊNCIA DA INFORMAÇÃO .....	94
CAPÍTULO 3 – PROPOSTA DE UM MODELO DE ANÁLISE DE REQUISITOS BASEADO NAS TEORIAS DA CIÊNCIA DA INFORMAÇÃO .....	94
3.1 APRESENTAÇÃO DO MODELO .....	94
3.2 MODELO PROPOSTO .....	96
3.2.1 Fase 1 – Conhecimento prévio .....	96
3.2.2 Fase 2 – Comunicação.....	97
3.2.2.1 Efetividade da Comunicação .....	98
3.2.2.2 Obstáculos devidos à personalidade.....	98
3.2.2.3 Obstáculos devidos à linguagem.....	99
3.2.2.4 Coleta de Informações.....	100
3.2.3 Fase 3 – Especificação .....	101
3.2.3.1 Modelos de representação do problema que busquem interação com o usuário .....	102
3.2.3.2 Prototipação .....	103
3.2.3.3 Documento de definição de requisitos .....	104
3.3 FINALIZAÇÃO DO MODELO .....	108
3.3.1 Visão da Análise de Requisitos .....	112
CONCLUSÃO .....	113
BIBLIOGRAFIA.....	116

## INTRODUÇÃO

O processo de desenvolvimento de software, consubstanciado em um conjunto de atividades e também no seu ordenamento, comporta três fases genéricas, independentemente, do paradigma de Engenharia de Software escolhido. São elas: a definição, o desenvolvimento e a manutenção (Pressman, 1995).

A fase de definição visa determinar o **que** vai ser feito. Nesta fase, deve-se identificar as informações que deverão ser utilizadas, as funções a serem processadas, o nível de desempenho desejado, as interfaces que devem ser oferecidas, as restrições do projeto e os critérios de validação. Essas atividades serão necessariamente desenvolvidas não importando o modelo de desenvolvimento adotado para o software e, independentemente da técnica utilizada para fazê-lo.

Nesta fase de definição ocorre a **identificação e análise de requisitos** do sistema. Segundo Pfleeger (2004), “um requisito é uma característica do sistema ou a descrição de algo que o sistema é capaz de realizar, para atingir os seus objetivos”. A análise de requisitos envolve muito mais do que simplesmente registrar o que o cliente quer. Deve-se identificar requisitos sobre os quais o analista e o usuário possam concordar e com os quais seja possível construir os procedimentos finais de teste.

Os principais obstáculos a serem transpostos na atividade de análise de requisito são:

- Formular questões precisas de requisitos de software;
- Obter informações significativas que ajudem a entender melhor os processos a serem sistematizados;
- Compreender a real necessidade do usuário/cliente para utilização do sistema;
- Reconhecer os elementos problemáticos básicos para a obtenção de informação;
- Reunir as pessoas “certas” que têm envolvimento com os principais processos.

Segundo uma pesquisa realizada pelo Standish Group em mais de 350 empresas americanas, envolvendo 8.000 projetos de software (Pfleeger, 2004: p.112), a maior causa de cancelamentos de projetos ou de não atendimento das expectativas do cliente, está relacionada com falhas no processo de análise de requisitos. A falta de cuidado com o entendimento, a documentação e o gerenciamento dos requisitos podem levar a uma grande quantidade de problemas, tais como: a construção de um sistema que resolve o problema errado, que não funciona como esperado, ou que é difícil para os usuários entenderem e utilizarem.

Na prática corrente, o analista assume que possui referenciais próprios sobre o sistema a ser desenvolvido (Pressmam, 1995). Ele parte do pressuposto que esses referenciais são mais importantes do que as informações em poder do usuário. Como iremos demonstrar nos capítulos seguintes, tal comportamento é, reconhecidamente, o fato gerador de um processo de comunicação “fraca” e “pobre” entre analista e usuário, que compromete o restante do desenvolvimento. Semelhante prática manifesta-se, via de regra, nos seguintes comportamentos do analista:

1. O analista em sua demonstração procura impor seus pontos de vista ao usuário condicionando-o a uma aceitação inadequada do requisito.
2. O analista, por já possuir experiências anteriores para a solução de problemas semelhantes, define requisitos que não levam em consideração a verdadeira amplitude do problema, reduzindo a comunicação com o usuário
3. O analista não obtém todo o conhecimento tácito em poder do usuário, criando requisitos incompletos.

Assim, é necessário adotar uma forma de comunicação mais ampla e significativa com o usuário, métodos de demonstração do problema e dos requisitos do sistema, de forma a tentar

garantir que todos os aspectos importantes, que estão na esfera do domínio de informação do usuário, sejam plenamente externalizados e formalizados para que se consiga chegar a uma definição de requisitos mais próxima da realidade.

Segundo Hayakawa (1972), existem certos fatores na comunicação humana que dificultam o entendimento dos interlocutores como, por exemplo, o “**enviés**” - processo em que se selecionam pormenores favoráveis ou desfavoráveis do assunto que está sendo descrito, exatamente pela bagagem cultural e experiências vividas dos atores envolvidos. O enviés torna deliberadamente inevitáveis certas interpretações nos comunicados.

Por outro lado, de acordo com Nonaka & Takeuchi, informações relevantes em poder do usuário não estão, naturalmente, externalizadas (Nonaka & Takeuchi, 1997). Assim, a externalização consiste num processo sistemático de conversão do conhecimento tácito em conhecimento explícito, através da articulação do conhecimento tácito.

Dentro deste conceito de externalização, proposto por Nonaka & Takeuchi, surge uma outra dificuldade na etapa de análise de requisitos que consiste em documentar o “arsenal” de informações e conhecimento tácito em poder do usuário.

A documentação é um conceito muito amplo, mas neste trabalho utilizou-se o relacionamento com o processo de armazenar e recuperar informações com a conseqüente utilização de linguagens documentárias. Essas linguagens são construídas para indexação, armazenamento e recuperação da informação e correspondem a sistemas de símbolos destinados a traduzir os conteúdos dos documentos (Cintra, 2002).

Segundo Kobashi (1994),

*“um sistema documentário propõe-se a responder de maneira duradoura a um conjunto específico de usuários. Desse modo, cada etapa da cadeia documentária é refratária à improvisação. A coleta de documentos deve ser sistemática, baseada no conhecimento dos organismos que produzem documentos que possam*

*interessar aos usuários do sistema. O tratamento, por sua vez, deve ser regido por parâmetros claros, sistematizados em uma política explícita e em procedimentos metodológicos claros. A difusão, finalmente, deve pautar-se por um princípio básico: os produtos e serviços devem ser planejados e executados de modo a responder às especificidades da demanda”.*

O processo de documentação reúne uma grande quantidade de características e detalhes e um grau de complexidade que, se não levados devidamente em consideração, podem gerar distorções importantes na tarefa de formalização do conhecimento tácito do usuário, acarretando o desenvolvimento de sistemas que não atendem de forma adequada às necessidades deste mesmo usuário.

Utilizando-se dos conceitos e métodos da Ciência da Informação relativos à comunicação com o usuário e a documentação, juntamente com a Engenharia de Software, que tem como foco principal a metodologia para o desenvolvimento do software, esta pesquisa busca compreender, na etapa de definição do sistema, os problemas de valorização, prospecção e formalização da informação.

O foco deste trabalho está nos aspectos da comunicação entre os atores principais de um projeto de desenvolvimento de software na etapa de análise de requisitos: analista e usuário. Existem diversos modelos na Engenharia de software que tratam da formalização dessa etapa. Existem modelos e padrões de qualidade de software, tais como CMM (Capability Maturity Model) e SPICE (Software Process Improvement and Capability dEtermination, projeto da norma ISO/IEC 15504), que também procuram estabelecer algumas diretrizes para que a análise de requisitos seja feita de acordo com os objetivos traçados pelo cliente-usuário. Entretanto, o foco desses modelos está em criar um formalismo que comprometa os integrantes do processo com aquilo que foi definido, escrito e assinado.

Este trabalho pesquisou os aspectos subjacentes desses modelos, tais como: a comunicação e os aspectos cognitivos entre os principais atores do processo de desenvolvimento de software, o modelo semiótico no processo comunicativo e suas principais características.

Para tanto se estabeleceu como objetivo geral, estudar e compreender as diversas etapas dos processos comunicacionais entre os atores envolvidos na atividade de análise de requisitos para desenvolvimento de softwares, com vistas à formulação de um modelo de análise de requisitos que considere as questões comunicacionais e do conhecimento da Ciência da Informação, na formalização das especificações do sistema.

Da mesma forma, os objetivos específicos foram discutir as abordagens da Engenharia de Software que tratam da análise de requisitos no processo de desenvolvimento de sistemas e compreender os principais problemas de comunicação entre os elementos envolvidos nesta etapa do processo; proceder a uma pesquisa exploratória acerca de teorias e processos vinculados à Ciência da Informação que discutem os aspectos subjacentes aos modelos comunicacionais e do conhecimento, como forma de compreender, na etapa de análise de requisitos do sistema, aspectos de valorização, prospecção e formalização da informação; estudar questões relevantes relativas ao processo de identificação, formalização e de organização das informações levantadas junto ao usuário, que possam ajudar analistas de sistemas e engenheiros de software no processo alvo do software a ser desenvolvido.

A experiência profissional do autor deste trabalho com o desenvolvimento de sistemas contemplou por diversas oportunidades a construção de sistemas que atendiam parcialmente ao usuário e algumas vezes sistemas que divergiam de requisitos importantes.

Nessa experiência, por mais formalizações que se fizessem no processo, ainda assim, conseguia-se chegar, na melhor das hipóteses, a sistemas não totalmente adequados ao usuário, o que exige, após o sistema já estar concluído, um maior trabalho de manutenção.

Com a experiência acumulada, muitas vezes de forma subjetiva, conseguia-se bons resultados, por uma comunicação mais eficiente com o usuário. Esse fato, que era uma conquista empírica até antes deste contato com a Ciência da Informação, constituiu a motivação para estudar melhor o processo de desenvolvimento de software, como forma de buscar explicitar mais claramente os aspectos subjetivos da prática, com vistas a geração de um ordenamento conceitual mais claro e rigoroso.

Softwares que não atendem as necessidades do usuário é um problema clássico da Engenharia de Software. Pesquisas sobre estudo de necessidades de usuário, visando a criação de softwares melhores e mais adequados, são uma constante. Porém os modelos adotados na Engenharia de Software não consideram os diferentes aspectos da comunicação humana, aspectos esses fundamentais para tratar o problema. Dessa forma, neste trabalho, a apropriação de teorias da Ciência da Informação que sustentam os modelos comunicacionais voltados para o usuário, mostrou-se de maior importância.

## ASPECTOS COMUNICACIONAIS

A Comunicação humana apresenta nuances que dificultam o completo entendimento do que se quer transmitir e/ou receber. Segundo Cherry (Cherry, 1968), *comunicação significa compartilhar elementos de comportamento, ou modos de vida, pela existência de conjuntos de regras*. A comunicação não é a resposta em si, mas essencialmente, a relação estabelecida pela transmissão de estímulos e pelo suscitamento de respostas.

Nos estudos de comunicação distinguem-se duas grandes correntes de investigação: uma que entende a comunicação, sobretudo como um fluxo de informação e, outra que entende a comunicação como uma “produção e troca de sentido”. A primeira corrente é a escola processual da comunicação e a segunda é a escola semiótica (Fidalgo, 1999).

Na linha processual da comunicação temos a obra clássica de Shannon e Weaver, *The Mathematical Theory of Information*, que estabelece o processo comunicativo em 3 níveis importantes:

1. Nível Técnico – relativo ao rigor da transmissão dos sinais;
2. Nível Semântico – relativo à precisão com que os sinais transmitidos convêm ao significado desejado;
3. Nível de Eficácia – relativo à eficácia com que o significado da mensagem afeta da maneira desejada a conduta do destinatário

Nesse modelo, a tarefa de comunicação tem seu foco na transmissão da mensagem entre A e B, partindo-se do pressuposto de que as mensagens já estão determinadas no seu significado, não se levando em consideração aspectos de formação da mensagem: sua estrutura interna, sua adequação ao que significa e sua relevância.

Já o modelo semiótico é aquele em que a ênfase é colocada na criação dos significados e na formação das mensagens a transmitir. Para que haja comunicação é preciso criar uma

mensagem a partir de signos (sinais). Esse modelo considera inseparáveis o conteúdo e o processo de comunicação.

Conteúdo e processo condicionam-se reciprocamente, pelo que o estudo de comunicação passa pelo estudo das relações sógnicas, dos signos utilizados, dos códigos em vigor das culturas em que os signos criam, vivem e atuam.

O modelo semiótico de comunicação não é linear, não se centra nos passos que a mensagem percorre desde a fonte até o destinatário. A comunicação não é tomada como um fluxo, mas como um sistema estruturado de signos e códigos. Esse modelo possui uma abordagem mais ampla para o estudo da comunicação entre “analista e usuário”, e será, portanto, o modelo a ser adotado neste estudo do processo de comunicação.

Na comunicação lingüística ainda existe a possibilidade de mais de uma interpretação, ocorrendo o que se chama de ambigüidade. Essa se dá tanto em função da homonímia dos vocábulos como pela polissemia (Cintra et al, 2002).

A polissemia ocorre quando uma palavra pode comportar mais de um significado como em “Hoje trabalhei muito com ar-condicionado”, em que se pode estar dizendo que se trabalhou tanto, em aparelhos de ar-condicionado ou, quanto, em ambiente refrigerado por ar-condicionado.

A homonímia consiste em uma mesma forma significante remeter a duas realidades vocabulares diversas, seja com relação à identidade fonética como na identidade gráfica. Por exemplo, numa frase como “O mestre entregou a cadeira ao colega”, o significante “cadeira” tanto pode remeter à palavra cadeira como objeto para sentar como cadeira igual à cátedra de um docente.

Essas características do processo comunicativo já demonstram a idéia da complexidade envolvida na relação entre “analista e usuário” que, se não levada em consideração, pode causar distorções.

*A Ciência da Informação tem como objetivo principal o estudo das propriedades gerais da informação e a análise dos processos de sua construção, sua comunicação e de seu uso* (cf. Le Coadic, 1997, p.516). Entretanto observa-se que a Ciência da Informação também está muito relacionada com a Tecnologia da Informação e, dentro dessa visão, seria a aplicação dos conhecimentos científicos para a concepção de técnicas e produtos de Sistemas de Informação que vão permitir a comunicação, o armazenamento e o uso da informação. O conceito de Sistemas de informação que se utilizou neste trabalho é o definido por Laudon (2004) de que “sistema de informação é um conjunto de componentes inter-relacionados que coleta, processa, armazena e distribui informações destinadas a apoiar a tomada de decisões, a coordenação e o controle de uma organização”. Além de dar suporte à tomada de decisões, à coordenação e ao controle, esses sistemas também auxiliam os gerentes e trabalhadores a analisar problemas, visualizar assuntos complexos e criar novos produtos.

Na Engenharia de Software existe uma preocupação em documentar os requisitos do cliente e estabelecer-se uma “baseline” para o gerenciamento dos mesmos. Uma *baseline* é um conjunto de definições aceitas e controladas, e que serão utilizadas em atividades posteriores à sua aceitação. A evolução da baseline se dá através da evolução das definições existentes, exclusão de definições ou adição de novos artefatos e documentos, e é um indicador do progresso realizado no desenvolvimento dos trabalhos. Uma vez estabelecida uma *baseline*, a alteração da mesma (dos artefatos aceitos e aprovados) só pode ser realizada via procedimento formal. Assim, a partir do momento que alguns requisitos forem definidos pode-se estabelecer

uma baseline de requisitos e desta forma, tem-se um controle maior sobre as alterações e as versões dos requisitos. A ênfase está na documentação dos requisitos, como se o fato de se colocar no papel o que se acha que se entendeu do levantamento de informações, fosse resolver todos os problemas para se iniciar um processo de desenvolvimento de sistemas.

Estabelecer os requisitos do cliente é um processo de interpretação do analista em relação ao entendimento do problema do usuário. Muitos desses requisitos não estão formalizados, encontrando-se de forma tácita em poder do usuário. Dessa forma, existe um trabalho delicado do analista em formalizar esse conhecimento tácito, de forma que se possa iniciar o processo de definição de análise de requisitos.

Segundo Pressman, (1995) existem diversos métodos de análise de requisitos, todos eles baseados em um conjunto de princípios fundamentais, tais como:

- O domínio de informação de um problema deve ser representado e compreendido
- Modelos que descrevam a informação, função e comportamento do sistema devem ser desenvolvidos
- Os modelos e o problema devem ser divididos em partições, de maneira que revelem os detalhes em forma de camadas
- O processo de análise deve mover-se da informação essencial para os detalhes de implementação.

Tanto o analista como o usuário têm posições ativas na etapa de análise de requisitos. O usuário tenta transpor seus conceitos sobre o problema e suas necessidades para funções e processos do software que ainda estão nebulosos. O analista age como indagador, consultor e solucionador de problemas.

O nível de ambigüidade nessa relação pode ser entendida por uma declaração clássica da Engenharia de Software: “Sei que você acredita que entendeu o que acha que eu disse, mas não estou certo de que percebe que aquilo que ouviu não é o que eu pretendia dizer...” (Pressman, 1995).

## **ASPECTOS METODOLOGICOS DA PESQUISA**

Este trabalho explorou o que está por traz da simples formalização do modelo de comunicação e os aspectos cognitivos entre os principais atores do processo de desenvolvimento de software, e do modelo semiótico no processo comunicativo e suas principais características. Para tanto, estaremos procedendo a uma pesquisa exploratória entre autores da Ciência da Informação que têm como foco de seus trabalhos, os diversos aspectos da comunicação humana e do conhecimento.

Este trabalho utiliza em primeiro lugar, o método de pesquisa documental para uma investigação formal e fundamentação sobre os conceitos mais importantes da Ciência da Informação e da Engenharia de Software ligados ao tema, a partir da discussão e do entendimento dos conceitos de análise documentária (Kobahsi, 1994), externalização e socialização (Nonaka & Takeuchi, 1997), conhecimento explícito (Lê Coadic, 1996), estruturação de problemas (Vakkari, 1998), linguagem (Hayakawa, 1972), comunicação humana (Cherry, 1968 e Penteadó, 1986) e da definição de requisitos centrada no usuário (Sato, 1995 e Saiedian, 2000).

Por meio desses procedimentos, espera-se poder selecionar informações relevantes (formais e informais); quem são os atores certos a serem envolvidos neste processo de levantamento de informações; quais são os aspectos críticos a serem considerados, garantindo maior confiabilidade e acerto no levantamento de informações na fase de definição de sistema.

Nesta pesquisa documental foi fundamental a realização de uma busca eletrônica no tesauro da base INSPEC para identificar os termos que indexam o assunto. Identificados os termos e a sua estrutura dentro do tesauro, procedeu-se a uma busca em linha, exaustiva, dentro da base INSPEC, utilizando-se do provedor DIALOG.

A base INSPEC corresponde a uma base de dados para as áreas de Física, Eletrônica e Computação, com resumos de artigos em inglês com sua respectiva indexação.

Este levantamento permitiu obter 65 referências, que após análise minuciosa dos assuntos tratados, considerou-se como relevantes 15 delas. Dessas referências solicitamos os artigos na íntegra, que nos permitiram concluir sobre os estudos e tendências que a temática "análise de requisito" suscita em nível internacional, o que trouxe contribuições muito importantes discutidas no capítulo 2 deste trabalho.

Utiliza-se ainda, o método da observação sistemática para coletar dados de um processo de análise de requisitos em um projeto real de desenvolvimento de sistemas.

Segundo Selltiz (Selltiz, 1971), a observação não é apenas uma das atividades mais difusas da vida diária; é também um instrumento básico da pesquisa científica. A observação se torna uma técnica científica na medida em que:

- a. serve a um objetivo formulado de pesquisa;
- b. é sistematicamente planejada;
- c. é sistematicamente registrada e ligada a proposições mais gerais, em vez de ser apresentada como conjunto de curiosidades interessantes;
- d. é submetida a verificações e controles de validade e precisão.

A grande vantagem das técnicas de observação é o fato de permitirem o registro do comportamento, tal como esse ocorre. Tais descrições são feitas, geralmente, de maneira objetiva, em que a pessoa que descreve está um pouco distante das tensões e preocupações que influem no que faz ou diz na vida diária, embora possa estar influenciada por outras tensões peculiares à situação da pesquisa.

A observação pode servir a diferentes objetivos da pesquisa. Pode ser usada de maneira exploratória, a fim de conseguir intuições que mais tarde serão verificadas por outras técnicas; seu objetivo pode ser a obtenção de dados suplementares significativos ou que possam

auxiliar na interpretação de resultados obtidos por outras técnicas; pode ser usada como método básico de coleta de dados nos estudos destinados à obtenção de descrições exatas de situações ou à verificação de hipóteses causais.

A observação direta ou participante é obtida por meio do contato direto do pesquisador com o fenômeno observado, para recolher as ações dos atores em seu contexto natural, a partir de sua perspectiva e seus pontos de vista.

Devido à área de atuação profissional do autor, análise e projetos de sistemas de informações, permitir um contato constante com projetos de manutenção e desenvolvimento de sistemas, utilizou-se deste método visando validar os pontos críticos apontados no capítulo 2 e também para verificar os resultados obtidos na implantação de um modelo para definição de requisitos, descrito no capítulo 3.

Este modelo foi validado em um projeto de implantação de um sistema de controle de reparos em um Centro Avançado de Reparos de aparelhos celulares, durante o ano de 2004 e que serviu como base para os ajustes e definição deste modelo.

A estrutura deste trabalho está dividida da seguinte forma:

- Do capítulo 1, consistindo da definição do marco teórico, fundamentado em 2 áreas específicas: a Engenharia de Software e a Ciência da Informação.
- No capítulo 2, discutem-se as principais dificuldades encontradas nos modelos adotados pela Engenharia de Software para a definição de requisitos e, identificam-se, sob o ponto de vista das teorias da Ciência da Informação, pontos de controle para sua organização na etapa de análise de requisito.

- No terceiro capítulo, apresenta-se um modelo para auxiliar a análise de requisitos, com base no foi discutido nos capítulos 1 e 2 e validado durante o projeto de desenvolvimento de sistemas descrito anteriormente.
- Conclui-se este estudo, fazendo-se uma avaliação e apresentando-se as considerações finais sobre possibilidades de soluções que podem ser adotadas e recomendações necessárias.

## **CAPÍTULO 1 – ENGENHARIA DE SOFTWARE E CIÊNCIA DA INFORMAÇÃO: CONCEITOS FUNDAMENTAIS**

Neste capítulo, apresenta-se o marco teórico, fundamentado em 2 áreas específicas: a Engenharia de Software e a Ciência da Informação. Na Engenharia de Software, realizou-se uma pesquisa exploratória dos principais conceitos ligados a análise de requisitos e o processo de desenvolvimento de sistemas. Na Ciência da Informação, trabalhou-se com os conceitos de externalização, semiótica, comunicação e documentação, procurando demonstrar como esses conceitos estão profundamente relacionados com a atividade de definição de requisitos e de que forma podem ajudar a criar uma nova abordagem.

Destaca-se a importância do processo de comunicação e dos aspectos cognitivos na relação analista-usuário, o modelo semiótico; as propriedades sintáticas, semânticas e pragmáticas dos signos e as estruturas da linguagem na comunicação são referenciais teóricos importantes deste trabalho para ajudar na proposição de um novo modelo a ser considerado para a etapa de análise de requisitos.

A atividade mais importante na etapa de análise de requisitos do processo de desenvolvimento de software é o *levantamento de informações*. Porém o conceito de informação na Engenharia de Software tem diferenças significativas face aos conceitos estabelecidos na Ciência da Informação.

Quando se discute hoje aplicações de software, as palavras “dados” e “informações” ocorrem repetidamente, muitas vezes como termos sinônimos. Nas décadas de 70 e 80, o termo “processamento de dados” era uma expressão operativa para descrever o uso de computadores num contexto comercial. Hoje, utiliza-se “tecnologia de informação” para representar a

mesma coisa, mas com uma sutil diferença de foco. A ênfase não está meramente no processamento de grandes quantidades de dados, mas ao contrário, em se extrair informações significativas desses dados.

Na concepção mais convencional da Engenharia de Software, segundo Pressman, (1995) *dados* são coleções de fatos que devem ser processados para se tornarem significativos – “são informações brutas”. A *informação* é obtida pela associação de fatos dentro de determinado contexto, ou seja, a partir da associação e organização dos dados se obtém uma significância, um sentido. Informações coletadas sobre uma variedade de tópicos relacionados e não relacionados são interligadas para formar um corpo de fato, ao qual denominamos *conhecimento*.

Na Ciência da Informação, o conceito de informação está associado à semântica, a harmonização do indivíduo com os fenômenos que lhe solicitam, como se pode constatar nas seguintes definições:

*[...] enquanto objeto da Ciência da Informação, a informação aparece como produto de um processo intencional, como algo construído, portanto, cujo propósito é o de promover a adequação significativa dos conteúdos.*(Tálamo, em Cintra, 2002, p.11)

*“A informação comporta um elemento de sentido. É um significado transmitido a um ser consciente por meio de uma mensagem inscrita em um suporte espaço-temporal: impresso, sinal elétrico, onda sonora, etc.”*(Lê Coadic, 1996, p.5).

Shannon e Weaver (em Molina, 1994, p.322), têm uma definição de informação mais adequada à construção de sistemas informáticos nos quais dados podem ser totalmente descritos através de representações formais, podendo ser quantificados, armazenados em um computador e processados por ele. Segundo Shannon, “Recebemos informação quando o que

conhecemos se modifica. Informação é aquilo que logicamente justifica alteração ou reforço de uma representação ou estado de coisas. As representações podem ser explicitadas como num mapa ou proposição, ou implícitas como no estado de atividade orientada para um objetivo do receptor”.

Esta definição de informação é a base da Teoria da Informação e é a que está mais próxima dos conceitos adotados pela Engenharia de Software e ainda desempenha um papel importante no estudo da informação em diversos contextos. Porém, o modelo da teoria da informação torna a relação dos atores envolvidos no processo informacional, uma relação linear, onde se considera análogos o conceito de informação da teoria matemática da transmissão de sinais elétricos e o conceito de informação do processo de comunicação humana (Le Coadic, 1996).

Segundo Machlup (em Barreto, 1994, p. 7), “a informação proporciona um novo ponto de vista para a interpretação de eventos ou objetos, o que torna visíveis significados antes invisíveis ou lança luz sobre conexões inesperadas. Por isso, a informação é um meio ou material necessário para extrair e construir conhecimento. Afeta o conhecimento acrescentando-lhe algo ou o reestruturando”.

O foco em problemas relacionados com o tratamento da informação pode ser bem compreendido na definição de Shera e Cleveland (em Batista, 2003):

*“Ciência da Informação é a ciência que investiga as propriedades e comportamento da informação, as forças que regem o fluxo da informação e os meios de processamento da informação para uma acessibilidade e usabilidade ótimas. Os processos incluem a origem, disseminação, coleta, organização, recuperação, interpretação e uso da informação. O campo deriva de ou relaciona-se com a matemática, a lógica, a lingüística, a psicologia, a tecnologia*

*da computação, a pesquisa operacional, as artes gráficas, as comunicações, a biblioteconomia, a administração e alguns outros campos”.*

Nonaka e Takeuchi (1997, p.64) destacam que “o aspecto semântico da informação é mais importante para a criação do conhecimento, pois se concentra no significado transmitido. Se limitarmos o escopo da consideração apenas ao aspecto sintático, não poderemos captar a verdadeira importância da informação no processo de criação do conhecimento”. Ainda em relação a complexidade do processo informacional, Nonaka e Takeuchi afirmam que “... tanto a informação quanto o conhecimento são específicos ao contexto e relacionais, na medida em que dependem da situação e são criados de forma dinâmica na interação social entre as pessoas”.

Desta forma, pode-se concluir que a abordagem sobre informação e conhecimento na Engenharia de Software estabelece uma visão linear de um processo complexo que, se não levadas em consideração tem-se uma comunicação analista-usuário pobre e deficiente.

## 1.1 O PROCESSO DE DESENVOLVIMENTO DE SISTEMAS

De um modo geral, pode-se organizar o processo de desenvolvimento de um software a partir de três grandes fases: a **fase de definição**, a **fase de desenvolvimento** e a **fase de manutenção**.

⇒ Fase de Definição

A fase de definição está associada à determinação do **que** vai ser feito. Nesta fase, o profissional encarregado do desenvolvimento do software deve identificar as informações que deverão ser manipuladas, as funções a serem processadas, qual o nível de desempenho desejado, que interfaces devem ser oferecidas, as restrições do projeto e os critérios de validação. Isto terá de ser feito não importando o modelo de desenvolvimento adotado para o software e independente da técnica utilizada para fazê-lo.

Esta fase é caracterizada pela realização de **três etapas específicas**:

**Análise (ou Definição) do Sistema**, a qual vai permitir determinar o papel de cada elemento (hardware, software, equipamentos, pessoas) no sistema, cujo objetivo é definir, como resultado principal, as funções atribuídas ao software;

**Planejamento do Projeto de Software**, no qual, a partir da definição do escopo do software, será feita uma análise de riscos e a definição dos recursos, custos e a programação do processo de desenvolvimento;

**Análise de Requisitos**, que vai permitir determinar o conjunto das funções a serem realizadas assim como as principais estruturas de informação a serem processadas.

⇒ Fase de Desenvolvimento

Nesta fase, será determinado **como** realizar as funções do software. Aspectos como a arquitetura do software, as estruturas de dados, os procedimentos a serem implementados, a forma como o projeto será transformado em linguagem de programação, a geração de código e os procedimentos de teste devem ser encaminhados nesta fase.

Normalmente, esta fase é também organizada em **três principais etapas**:

**Projeto de Software**, o qual traduz, num conjunto de representações gráficas, tabulares ou textuais, os requisitos do software definidos na fase anterior; estas representações (diversas técnicas de representação podem ser adotadas num mesmo projeto) permitirão definir, com um alto grau de abstração, aspectos do software como a arquitetura, os dados, lógicas de comportamento (algoritmos) e características da interface;

**Codificação**, onde as representações realizadas na etapa de projeto serão mapeadas numa ou em várias linguagens de programação, a qual será caracterizada por um conjunto de instruções executáveis no computador; nesta etapa, considera-se também a geração de código de implementação, aquele obtido a partir do uso de ferramentas (compiladores, linkers, etc...) e que será executado pelo hardware do sistema;

**Testes de Software**, onde o programa obtido será submetido a uma bateria de testes para verificar (e corrigir) defeitos relativos às funções, lógica de execução, interfaces, etc...

⇒ Fase de Manutenção

A fase de manutenção, que se inicia a partir da entrega do software, é caracterizada pela realização de alterações de naturezas as mais diversas, seja para corrigir erros residuais da fase anterior, para incluir novas funções exigidas pelo cliente, ou para adaptar o software a novas configurações de hardware.

Sendo assim, pode-se caracterizar esta fase pelas seguintes **atividades**:

- **Correção** ou **Manutenção Corretiva**, a qual consiste da atividade de correção de erros observados durante a operação do sistema;
- **Adaptação** ou **Manutenção Adaptativa**, a qual realiza alterações no software para que ele possa ser executado sobre um novo ambiente (CPU, arquitetura, novos dispositivos de hardware, novo sistema operacional, etc...);
- **Melhoramento Funcional** ou **Manutenção Perfectiva**, onde são realizadas alterações para melhorar alguns aspectos do software, como por exemplo, o seu desempenho, a sua interface, a introdução de novas funções, etc...

A manutenção do software envolve, normalmente, etapas de análise do sistema existente (entendimento do código e dos documentos associados), teste das mudanças, teste das partes já existentes, o que a torna uma etapa complexa e de alto custo.

Além disso, considerando a atual situação industrial, foi criado, mais recentemente, o conceito de **Reengenharia**, em que, através do uso das técnicas e ferramentas da Engenharia de Software, o software existente sofre uma "reforma geral", cujo objetivo é aumentar a sua qualidade e atualizá-lo com respeito às novas tecnologias de interface e de hardware.

## 1.2 O PROCESSO DE ANÁLISE E IDENTIFICAÇÃO DE REQUISITOS

A análise de requisitos possibilita que o analista de sistemas especifique a função e o desempenho do software, indique a interface do software com outros elementos do sistema e estabeleça quais são as restrições de projeto que o software deve enfrentar. A análise de requisitos proporciona ao projetista de software uma representação da informação e da função que pode ser traduzida em projeto procedimental, arquitetônico e de dados (Pressman, 1995).

Segundo Pfleeger (2004), “um requisito é uma característica do sistema ou a descrição de algo que o sistema é capaz de realizar, para atingir os seus objetivos”.

A identificação dos requisitos é uma parte especialmente importante do processo de desenvolvimento de software. Deve-se determinar o que os usuários e os clientes realmente querem, com o objetivo de entender o problema quando ainda não encontramos uma solução.

A meta do analista é o reconhecimento dos elementos problemáticos básicos, conforme percebidos pelo usuário/cliente. Logo que o problema é identificado, o analista determina quais informações devem ser produzidas pelo novo sistema e quais dados serão oferecidos ao sistema.

Depois de avaliar os problemas atuais e as informações desejadas (entrada e saída), o analista começa a sintetizar uma ou mais soluções. O processo de avaliação e síntese prosseguirá até que o analista bem como o cliente tenham confiança de que o software está adequadamente especificado para as etapas de desenvolvimento subsequentes.

No decorrer da síntese de avaliação e solução, o principal foco do analista recai sobre “o que” e não sobre “como”. Quais dados o sistema produz e consome, quais funções o sistema deve executar, quais interfaces são definidas e quais restrições se aplicam.

Durante a atividade de síntese de avaliação e solução, o analista cria modelos do sistema num esforço para compreender melhor o fluxo de dados e de controle, o processamento funcional e a operação comportamental, além do conteúdo de informação. Estes modelos servem como um fundamento para o projeto de software e como base para criação de sua definição.

Logo que informações básicas, funções, desempenho, comportamento e interfaces forem descritos, critérios de validação serão especificados para demonstrar o entendimento e viabilizar uma implementação de software bem sucedida. Esses critérios servem de base para as atividades de teste que ocorrerão posteriormente no processo de engenharia de software.

Uma especificação de exigências formal é escrita para definir as características e os atributos do software. Nesta fase é importante que o analista assuma o ponto de vista de usuário do software, visando estimular a produção de documentos que demonstrem como o software irá funcionar, a sua abrangência em termos de interações com outros sistemas já existentes e principais opções a serem oferecidas, na forma de um manual do usuário preliminar, visto que o software ainda não existe.

Os documentos da análise de requisitos (especificação e manual do usuário) servem de base para uma revisão levada a efeito pelo cliente e pelo desenvolvedor. Esta revisão quase sempre resulta em modificações na função, desempenho, representações da informação, restrições ou critérios de validação.

Pfleeger (2004) separa o processo de requisitos em 2 partes distintas: **análise de requisitos**, onde se procura trabalhar com o usuário para a obtenção dos requisitos, fazendo-lhes perguntas, demonstrando sistemas similares e **a definição e especificação de requisitos**, onde se procura registrar os requisitos em um documento. A figura 1.1 ilustra melhor esta visão.

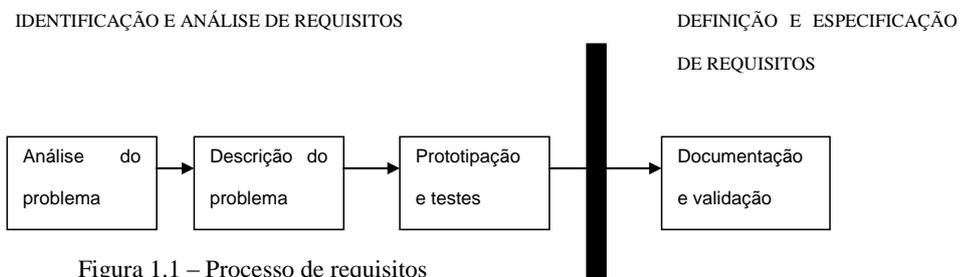


Figura 1.1 – Processo de requisitos

### 1.2.1 Pontos críticos na análise de requisitos

A análise de requisitos é uma atividade com comunicação intensiva. Quando ocorre a comunicação, a interpretação errônea ou omissão de informações (ruído) podem acarretar dificuldades tanto para o analista como para o cliente.

O início da tarefa de análise (reconhecimento de problemas e síntese de avaliação e solução) se baseia na aquisição, bem sucedida, das informações. Frequentemente as informações fornecidas pelo cliente entram em conflito com outras exigências declaradas anteriormente por outras pessoas; a função e o desempenho entram em conflito com as restrições impostas por outros elementos do sistema, ou a percepção das metas do sistema modifica-se com o tempo. Desta forma, algumas questões fundamentais surgem nesta fase: quais informações devem ser compiladas e como devem ser representadas? Quem fornece as várias peças de informação? Quais informações estão ligadas ao escopo do sistema?

Algumas causas são atribuíveis aos problemas subjacentes à análise de requisitos:

- Comunicação que dificulta a aquisição de informações;
- Técnicas e ferramentas inadequadas que resultam em definição inadequada ou imprecisa;
- Falta de se levar em consideração alternativas, antes que o software seja especificado.

### **1.2.2 Técnicas de comunicação na análise de requisitos**

A análise de requisitos de software sempre começa com a comunicação entre duas ou mais partes: o usuário-cliente e o analista-desenvolvedor.

A técnica de análise mais comumente usada para melhorar a falta de comunicação entre o cliente e o desenvolvedor e para fazer com que o processo de comunicação seja iniciado é realizar um encontro ou entrevista preliminar. O primeiro encontro entre um analista e o cliente pode parecer com o desconcerto de um encontro marcado entre dois adolescentes. Nenhum deles sabe o que dizer ou perguntar; ambos estão preocupados com a possibilidade de que sejam mal interpretados; ambos estão pensando sobre onde isso tudo poderia levar; ambos querem que a coisa se encerre logo, mas ambos querem que ela seja um sucesso.

Pressman (1995) sugere que o analista comece fazendo perguntas de livre contexto, ou seja, um conjunto de perguntas que leve a uma compreensão básica do problema; à natureza da solução que é desejada e à efetividade do primeiro encontro em si. O primeiro conjunto de perguntas de livre contexto concentra-se no cliente, nas metas globais e nos benefícios, como exemplo:

- Quem está por trás do pedido desse trabalho?
- Quem usará a solução?
- Qual benefício econômico desta solução se for bem sucedida?
- Há outra fonte para a solução exigida?

O conjunto de perguntas seguinte capacita o analista a obter uma melhor compreensão do problema e, o cliente a verbalizar suas percepções sobre uma solução:

- Como você caracterizaria um bom resultado que seria gerado por uma solução bem sucedida?
- Qual problema essa solução resolverá?
- Você poderia mostrar o ambiente em que a solução será usada?
- Existem questões de desempenho ou restrições especiais que afetarão a maneira pela qual a solução é abordada?

O próximo conjunto de perguntas concentra-se na efetividade do encontro (metaperguntas):

- Você é a pessoa certa para responder a essas perguntas? Suas respostas são oficiais?
- Minhas perguntas são pertinentes ao problema que você tem?
- Estou fazendo perguntas demais?
- Há mais alguém que possa fornecer informações adicionais?
- Existe algo mais que eu deva perguntar-lhe?

Essas e outras perguntas ajudarão a “quebrar o gelo” e a iniciar a comunicação essencial para a análise bem sucedida.

### 1.3 GERAÇÃO DE CONHECIMENTO NA ETAPA DE ANÁLISE DE REQUISITOS

Segundo Pressman (1995) “a análise de requisitos é uma tarefa da engenharia de software que efetua a ligação entre a alocação de software em nível de sistema e o projeto de software”. Ela proporciona ao analista de sistemas uma representação da informação e da função que pode ser traduzida em projeto procedimental, arquitetônico e de dados. A meta do analista é o reconhecimento dos elementos problemáticos básicos, conforme percebidos pelo usuário/cliente.

Nesta etapa, as interações entre analista e usuário são muito intensas e de diversas ordens:

- a) o analista efetua diversas entrevistas visando a compreensão do problema e das necessidades do usuário;
- b) o analista efetua um levantamento da existência de procedimentos formais e documentação relacionados ao problema do usuário;
- c) o usuário “tenta” compreender as necessidades do analista e disponibiliza o “arsenal de informações” que possui, relacionado a estas necessidades.

Um texto clássico do cientista convertido a filósofo, Michael Polanyi, *The Tacit Dimension*, publicado originalmente em 1966, é um marco para a conceituação do que vem a ser o conhecimento tácito (Polanyi, 1997). A importância deste tema na literatura organizacional acreditamos, tem sido bem explorada, primeiramente por Mitzenberg (1989), com seus trabalhos sobre o papel desempenhado pela intuição no processo de tomada de decisão gerencial e, mais recentemente por Nonaka e Takeuchi (1995), em sua obra *The Knowledge-Creating Company*, na qual estes autores elaboram um modelo de criação de conhecimento baseado no círculo virtuoso da interação entre conhecimento tácito e explícito.

Polanyi (1997) introduz o tema do conhecimento tácito a partir da frase *we can know more than we can tell*. Com isto ele quer dizer que muito do que sabemos não pode ser verbalizado ou escrito através de palavras. Isto fica mais claro a partir dos exemplos cotidianos e científicos apresentados por ele. Como exemplos mais intuitivos ele cita:

- a. *“nossa capacidade de conseguir distinguir o rosto de uma pessoa conhecida entre tantas outras, mas não sermos capazes de explicitar os particulares que compõem o todo. Daí os artifícios usados pela polícia que, para fazer retratos falados, utiliza uma vasta coleção de fotos de partes específicas do rosto, como narizes, bocas e outros detalhes;*
- b. *o fato de pianistas virtuosos tenderem a ficar paralisados, quando fixam sua atenção ao movimento de seus dedos;*
- c. *a habilidade de pessoas cegas, que usam a bengala como uma extensão de seu corpo”.*

Além disso, Polanyi (1997) relaciona vários estudos em laboratório que mostram a nossa capacidade de reação intuitiva a estímulos externos não compreendidos pela nossa mente consciente. Neste sentido, Polanyi cita os exemplos de testes psicológicos envolvendo "shock syllables" e "shock words". Nestes testes, verificou-se que as pessoas eram capazes de antecipar choques elétricos ao verem palavras e sílabas que, de fato, como parte da lógica do experimento, estavam associadas a choques, mesmo sem serem capazes, a posteriori, de explicitar a lógica do experimento.

Ainda neste estudo, verificou-se, também em laboratório, através de câmaras escondidas de alta resolução, que muitos dos movimentos dos músculos, ditos "involuntários", poderiam ser, na verdade, estimulados externamente, mesmo sem a consciência humana.

Para Polanyi, o conhecimento tácito envolve, pois, uma relação entre duas coisas: um conhecimento específico, como "tocar piano", utilizar uma ferramenta, etc, que ele chama de distal e um outro, que ele chama de proximal, do qual só temos consciência na medida em que ele serve ao anterior. Ele compara ainda seus conceitos de conhecimento tácito aos resultados da *gestalt*. Esta seria o resultado de um esforço ativo e laborioso de busca de conhecimento, que envolveria o nosso corpo e todos os seus sentidos. Neste aspecto, a aquisição de conhecimento, seja intelectual ou prático, seria resultado de envolvimento e compromisso pessoal.

O conceito de conhecimento tácito também ajudaria a explicar por que a compreensão de assuntos complexos pode ser prejudicada, quando se busca fazê-lo, primordialmente, a partir de sua decomposição em partes e análises estritamente racionais. Não que isto não deva ser feito para se obter um aprofundamento sobre o tema em questão. O importante é não perder o foco no todo e isto só ocorre, quando a realidade é interiorizada e todos os detalhes recuperam seu significado e complexos relacionamentos. Dessa maneira, Polanyi expressa total rejeição a qualquer noção de conhecimento absolutamente objetivo, inclusive nas ciências naturais, pois mesmo a formulação de teorias, neste campo, envolveria o conhecimento tácito.

Segundo Polanyi, “o compartilhamento do conhecimento tácito entre indivíduos através da comunicação constitui um processo análogo que exige uma espécie de processamento simultâneo das complexidades dos problemas compartilhados pelos indivíduos. O conhecimento tácito é criado *aqui e agora* em um contexto prático específico. Por outro lado, o conhecimento explícito lida com acontecimentos passados ou objetos *lá e então* e é orientado para uma teoria independente do contexto”.

Nonaka e Takeuchi (1997) em sua *teoria da criação do conhecimento organizacional* afirmam que o conhecimento humano é criado e expandido através da interação social entre o conhecimento tácito e o conhecimento explícito. Esta conversão é um processo social entre indivíduos e não confinada dentro de um indivíduo. Para isto, existem quatro formas bem distintas de criação do conhecimento:

1. **A Socialização**, que cria conhecimento tácito através do próprio conhecimento tácito. Ela é um processo de compartilhamento de experiências. Como exemplo, temos os aprendizes que trabalham com seus mestres e aprendem sua arte não através da linguagem, mas sim através da observação, imitação e prática. O segredo para a aquisição de conhecimento é a experiência.
2. **A Externalização**, que cria conhecimento explícito através de conhecimento tácito. A Externalização deve ser entendida como uma aplicação seqüencial da metáfora, analogia e modelo. A metáfora é a forma de perceber ou entender intuitivamente uma coisa, imaginando outra simbolicamente. A analogia vai harmonizar as contradições inerentes ao fato da metáfora estar tratando de coisas distintas. Os modelos são gerados a partir de metáforas quando são criados novos conceitos.
3. **A Combinação**, que cria conhecimento explícito através de conhecimento explícito. A combinação acontece quando indivíduos trocam e combinam conhecimentos através de meios como documentos, reuniões, conversas ao telefone ou redes de comunicação. A criação de conhecimento realizada através da educação e do treinamento formal nas escolas normalmente assume essa forma.
4. **A Internalização**, que cria conhecimento tácito através de conhecimento explícito. Quando são internalizadas nas bases de conhecimento tácito dos indivíduos sob a forma de modelos mentais ou *know-how* técnico compartilhado, as experiências através da socialização, externalização e combinação tornam-se ativos valiosos. Um

exemplo de internalização pode se dar quando ao se ler ou ouvir uma história de sucesso faz com que alguns dos participantes sintam o realismo e a essência da história, a experiência que ocorreu no passado pode se transformar em um modelo mental tácito. Quando a maioria dos participantes compartilha de tal modelo mental, o conhecimento tácito passa a fazer parte da cultura do grupo.

Na análise de requisitos, tem-se presentes as quatro formas de geração de conhecimento definidas anteriormente, mas devido ao fato de as interações entre usuários e analistas antes e após o desenvolvimento do software serem um processo infinito de compartilhamento do conhecimento tácito e criação de idéias para o aperfeiçoamento, a Socialização e a Externalização são o foco maior deste estudo para uma abordagem diferenciada da abordagem convencional da Engenharia de Software.

#### **1.4 O PROCESSO DE DOCUMENTAÇÃO NA ETAPA DE ANÁLISE DE REQUISITOS**

Segundo Rocha (2001, p.55), “tradicionalmente, pouca atenção tem sido dada à documentação gerada durante os projetos de desenvolvimento de software que tem resultado em documentos mal-elaborados, de difícil compreensão, inadequados ou até mesmo incompletos”. Os maiores especialistas na área de engenharia de software reconhecem que a falta de um projeto de documentação tem atrapalhado a manutenção do software durante toda a história do desenvolvimento de software.

Documentação é o processo responsável pela triagem, organização e conservação da informação, bem como pela viabilização do seu acesso (Cintra, 2002).

Segundo Nair Kobashi (1996, p. 5) “a preocupação teórica com a organização e a representação de informações, com fins documentários, é fato relativamente recente se levarmos em conta as práticas relacionadas a esses processos, que são da época das cidades-estado, que já apresentavam uma forma de tratamento documentário: as obras cunhadas em tábuas de argila eram protegidas por uma espécie de envelopes, sobre os quais se transcreviam informações que cumpriam função semelhante à dos modernos resumos”.

Dentro deste processo evolutivo da análise documentária, o foco passou a estar na indexação, elaboração de resumos e recuperação de informações. Começa-se a atribuir valor aos aspectos comunicacionais dos sistemas documentários e aos instrumentos que possam promover níveis crescentes de interação entre sistema e usuários, tais como os tesouros.

Inicialmente utilizados para servir ao indexador para selecionar o termo mais adequado para descrever o conteúdo informacional dos documentos, os tesouros passam a ser propostos, igualmente, como ferramentas valiosas para auxiliar o usuário a explorar adequadamente os

sistemas documentários. O ato documentário é um ato de comunicação que tem a finalidade de promover a circulação da informação.

Devido ao fato de uma boa parte da atividade de análise de requisitos ser constituída de levantamento de informações, formais e informais, é necessário, que o analista se organize de forma a manter documentado todo esse levantamento, visando dois objetivos principais:

- Recuperar essas informações com acuracidade, de forma que analista e usuário possam entendê-la.
- Representar o levantamento de forma mais próxima da realidade, sem perder os detalhes importantes.

O aspecto da recuperação das informações contidas em um documento é o foco maior da Análise Documentária. Ela se apóia em dois processos principais: a indexação e o resumo. *Na indexação, procura-se obter um grau maior de compactação do texto-base, com fragmentos que procuram caracterizá-lo por meio de palavras ou de sintagmas. No resumo, o texto-base é desestruturado de modo a permitir que certos fragmentos, organizados sob a forma de um novo texto, reconstruam potencialmente o sentido do original (Kobashi, 1996).*

Outro ponto importante é com relação ao método de registro das impressões e observações que o analista teve em contato com o usuário, durante um processo de levantamento de informações. Indiscutivelmente, o melhor momento para registro se localiza na duração do acontecimento. Disso resulta um mínimo de viés seletivo e de deformação pela memória. Nas situações em que não é possível a anotação imediata e minuciosa, a memória do analista pode ser muito sobrecarregada se o registro for transferido para depois do período de observação. Para tais situações, é importante adquirir o hábito de anotar, de maneira quase imperceptível,

palavras chave significativas, que irão orientar posteriormente ao detalhamento da informação sem com isto perder os detalhes da ação em execução.

Qualquer que seja o método de registro de suas impressões imediatas, o analista deve escrever, tão logo quanto possível depois de um período de observação, uma descrição completa de tudo na situação que deseja lembrar. A anotação geralmente será feita sob forma de narrativa.

Segundo Pfleeger (2004) “os requisitos devem ser escritos de maneira compreensível, não somente para os usuários, mas também para os projetistas da equipe de desenvolvimento”. Os requisitos devem ser organizados de tal modo que possam ser acompanhados durante o desenvolvimento do sistema. O ideal é que qualquer característica ou função do sistema possa ser acompanhada até chegar ao requisito que a gerou, e vice-versa.

Pfleeger (2004) define 2 tipos de documentação para a análise de requisitos: **documento de definição de requisitos** e **documento de especificação de requisitos**. O documento de definição de requisitos é escrito em um nível apropriado para o usuário e em termos compreensíveis para este. O documento de especificação de requisitos é escrito a partir da perspectiva do desenvolvedor.

As principais etapas que um documento de definição de requisitos deve contemplar são:

Propósito geral do sistema

Fundamento e objetivos do desenvolvimento do sistema

Visão geral do problema

Características detalhadas do sistema proposto

Ambiente em que o sistema irá operar.

Esta documentação é feita em linguagem natural do usuário e isto normalmente causa diferentes interpretações, entre o analista e o usuário, daí a importância de entrar na esfera do

usuário para se garantir que o documento terá o nível de esclarecimento necessário para que todos os participantes se entendam.

A documentação de especificação de requisitos normalmente utiliza 2 tipos de descrições: **estáticas e dinâmicas.**

A descrição estática lista as entidades ou os objetos do sistema, seus atributos, funções que podem ser realizadas e seus relacionamentos. Nesta categoria a principal técnica utilizada é da abstração de dados. Ela consiste em descrever para que servem os dados, em vez de como eles se parecem. A idéia central é categorizar dados e agrupá-los como elementos. Cada tipo de dado é denominado objeto e recebe um nome e os elementos de dados são associados por tipo ou classe. As ações permitidas com os dados e tipos de dados são chamadas de métodos.

A descrição dinâmica consiste em descrever um sistema em termos das relações entre suas entidades, explicando como o sistema reage às coisas que modificam o seu comportamento em determinado período. Nesta categoria, a principal técnica utilizada é a da descrição funcional e diagramas de transição, que consistem em visualizar um sistema de maneira semelhante a um conjunto de estados, que o sistema reage a certos eventos possíveis. O comportamento do sistema é interpretado como uma série de funções, a entrada para essas funções é um conjunto de condições e um evento é o resultado da ação sobre o sistema que resulta em uma mudança de estado do sistema.

Produzir um documento, sob o ponto de vista da Engenharia de Software, é diferente de organizar informações com fins documentários. A Ciência da informação tem esta preocupação. O documento de definição de requisitos contém todas as ambigüidades da relação entre analista e usuário e consiste na base para a elaboração de todas as definições

subseqüentes que irão respaldar o processo de desenvolvimento de sistemas. Se não houver uma preocupação com a recuperação das informações contidas de forma íntegra e precisa, fatalmente teremos um documento que irá distorcer o resultado final do desenvolvimento do sistema.

## **1.5 PADRONIZAÇÃO DE TERMINOLOGIAS NA ANÁLISE DE REQUISITOS**

O termo ontologia surgiu na transição da Idade Média para Moderna e é a parte da filosofia que trata do ser enquanto ser, isso é, do ser concebido como tendo uma natureza comum que é inerente a todos e a cada um dos seres. Pode ser definida como a teoria dos princípios materiais do conhecimento humano, ou seja, os pressupostos materiais mais gerais do conhecimento científico. Pode ser definida, também, como a teoria do pensamento verdadeiro sobre os objetos do mundo fenomenal (Viera & Chaves, em Abbas, 2004).

Num sentido mais técnico, ontologia pode ser vista como uma ferramenta de engenharia que descreve uma certa realidade com um vocabulário específico, usando um conjunto de premissas de acordo com o sentido intencional das palavras do vocabulário. No sentido filosófico, ontologia é um sistema específico de categorias que reflete uma visão específica do mundo (Abbas, 2004). O conceito consagrado de uma ontologia é uma especificação explícita de uma conceituação.

Entretanto, o conceito que mais se adéqua às necessidades deste trabalho é o de que uma ontologia é uma maneira de se definir um conteúdo específico sobre um conhecimento a ser compartilhado e reusado entre diferentes agentes, estabelecendo convenções em três níveis:

1. Formato de representação da linguagem
2. Protocolo de comunicação entre agentes (humanos ou computacionais)
3. Especificação do conteúdo do conhecimento compartilhado (conceituação)

A análise de requisitos se baseia na aquisição bem sucedida de informações. Esta aquisição normalmente esbarra em informações conflituosas fornecidas pelo usuário-cliente em relação com outras exigências declaradas por pessoas que também estão envolvidas no processo e

noções de funções e desempenho do sistema que estão em desacordo com as metas estabelecidas originalmente.

Na análise de requisitos, como já discutido anteriormente, o conteúdo comunicacional é muito intenso entre o analista de sistemas e o usuário. As possibilidades de falhas de interpretação de ambas as partes são muito grande. A ontologia pode ajudar os elementos desta relação a raciocinar e a entender o domínio do conhecimento e atuar como uma referência para a obtenção do consenso neste contexto, sobre o vocabulário técnico a ser usado nas suas interações, além de se constituir em um excelente guia no processo de especificação de conhecimento das diversas fontes envolvidas. Este vocabulário de consenso pode representar o conhecimento envolvido de forma explícita no seu mais alto nível de abstração, possuindo um enorme potencial de reuso. O conhecimento formalizado pode ser especializado em diferentes aplicações, servindo diferentes propósitos, por diferentes equipes de desenvolvimento de sistemas.

A busca de uma padronização na documentação e no vocabulário da etapa de análise de requisitos incorre na dificuldade de encontrar-se processos organizacionais muito variados e complexos, que nem sempre podem ser harmonizados e padronizados. Porém, esta busca é possível, na medida em que para cada projeto se estabeleçam alguns referenciais que serão básicos para a configuração desta padronização, que facilitaria muito as dificuldades de comunicação da análise de requisitos.

Se levado em consideração todas estas dificuldades, o analista poderia criar um conjunto de perguntas ao usuário de forma a obter uma compreensão maior do problema e o usuário verbalizar melhor suas percepções sobre possíveis soluções. Combinando isto com elementos

de solução de problemas, negociação e especificação, podem-se minimizar estas dificuldades naturais do processo comunicacional.

Atualmente uma das principais aplicações das ontologias na Engenharia de Software se dá no uso do padrão XML como uma definição de meta-dados para a troca de informações entre diferentes sistemas de informação baseados na Internet, porém no que diz respeito a análise de requisitos, o mais importante é buscar uma padronização de terminologias utilizadas nas etapas de entrevistas e apresentação de documentos, que expõe as informações levantadas pelo analista de sistemas ao usuário, de forma que ambos possam se entender melhor no uso destas terminologias.

## **1.6 CONHECIMENTO PRÉVIO PARA ANÁLISE DE REQUISITOS**

Segundo Pertti Vakkari (Vakkari, 1998) toda tarefa a ser executada por um agente, quando este não tem um conhecimento prévio sobre o tema, gera um estado anômalo do conhecimento. A falta de entendimento gera ações de informação para resolver a situação problemática a fim de continuar com a tarefa. Os principais elementos nessa situação são ações a serem suportadas pela informação, conhecimento prévio insuficiente do ator e mecanismos de suporte informacional. Quanto mais o ator conhece sobre a dimensão da tarefa, ela se torna menos complexa e mais fácil de ser terminada. Desse modo pode-se ligar o grau de pré-determinabilidade de uma tarefa à estruturação do conhecimento ou aspecto conceitual de desenvolvimento sobre a tarefa. A estrutura do espaço conceitual depende de um conhecimento prévio, que a pessoa tenha sobre a tarefa. Se houver uma severa falta de conhecimento sobre a tarefa, pode-se dizer que a pessoa está em uma situação problemática e está em um estado anômalo do conhecimento.

O conhecimento prévio sobre uma tarefa pelo ator é o maior fator na determinação de que informação é necessária para sua realização. Pesquisas recentes na área de psicologia cognitiva e pesquisa organizacional tem mostrado que a percepção humana e o aprendizado de novas categorias dependem do conhecimento pessoal e teorias sobre o mundo. As teorias referem-se ao corpo de conhecimento que pode incluir princípios científicos, estereótipos, e observação informal de experiências passadas. No aprendizado de novas categorias as pessoas agem como se essas categorias fossem consistentes com o conhecimento prévio. As pessoas parecem agir economicamente de forma que os conhecimentos prévios são reusados quando possível.

Pode-se concluir que conhecimento prévio é vital para determinar que informação é necessária para completar uma tarefa. Isso implica que o grau de conhecimento sobre uma tarefa é o maior fator, que determina que tipo de informação é procurado, como a estratégia de busca é formulada e como a informação descoberta é avaliada e utilizada. Assim, mudanças no conhecimento prévio dirigem as mudanças nas atividades de informação. Quanto mais sabemos, ou quanto maior nosso conhecimento prévio, mais podemos antecipar e predeterminar o desenvolvimento da tarefa e suas dimensões.

Pode-se resumir a relação de nossos conceitos básicos assim: quanto mais complexa a tarefa, mais mal estruturada ela está, e menos conhecimento prévio o ator tem. Quanto mais rica a estrutura conceitual de um ator sobre uma tarefa, mais claramente pode ser expresso que tipo de informação é útil.

A análise de requisitos como já se demonstrou, tenta estabelecer claramente qual é o problema do usuário de forma a estabelecer um caminho viável para se atingir a solução deste problema. Desta forma, o conhecimento prévio do analista sobre o problema pode ajudar a extrair do usuário as informações necessárias para garantir uma identificação de requisitos com maior acuracidade, sintonizada com o problema do usuário, de forma a garantir maior aderência do sistema aos objetivos inicialmente determinados.

## 1.7 AS FUNÇÕES DA LINGUAGEM

Segundo Hayakawa (1972) o ser humano nunca depende de si mesmo para obter informações. Até na cultura mais primitiva, ele podia utilizar-se da experiência dos vizinhos, amigos e parentes e que lhe era transmitida por meio de linguagem. A linguagem é que torna possível o progresso.

Com o advento da escrita a exatidão dos comunicados pôde então ser examinada e reexaminada pelos observadores e a quantidade de conhecimento acumulado deixou de ficar limitada a capacidade das pessoas em lembrar-se daquilo que lhes foi contado.

Desde o grito de alarma do homem primitivo até a última monografia científica ou notícia radiofônica a linguagem é um fenômeno social e a cooperação cultural e intelectual constitui o grande princípio da vida humana.

As palavras – o modo como as emprega e o modo como as recebe quando proferidas pelos outros – plasmas, em larga medida as suas crenças, preconceitos, idéias e aspirações. Constituem, em suma, a atmosfera moral e intelectual na qual se vive (ambiente semântico).

Quando se fala de comunicação humana, existem alguns conceitos de fundamental importância para uma compreensão melhor dos problemas gerados nesta comunicação, que são definidos por Cherry (1968) como:

- a) Inferência – é uma asserção sobre o desconhecido, feita na base do conhecimento.
- b) Julgamentos - são todas as expressões de aprovação ou desaprovação do escritor em face de acontecimentos, pessoas e objetos por ele descritos. O julgamento prematuro impede-nos de ver o que se acha logo à nossa frente
- c) Enviés – seleciona pormenores favoráveis ou desfavoráveis do assunto que está sendo descrito. O enviés torna deliberadamente inevitáveis certos julgamentos nos comunicados.

- d) Significado Extensional – alguma coisa que não pode ser expressa por meio de palavras, pois é aquilo que as palavras representam
- e) Significado Intencional – aquilo que nos é sugerido pela nossa própria cabeça.

Hayakawa (1972) afirma que, mediante o processo de seleção e abstração que os nossos interesses e a formação cultural nos impõem, a experiência já nos vem “de enviés”. Se acontece de sermos pro-catolicismo, nossas idéias a cerca do que é importante ou não, necessariamente diferirão das idéias do homem que é indiferente a este tema. Quando as asserções têm apenas significados intencionais sem os extensionais, as discussões podem prolongar-se indefinidamente.

### **1.7.1 Conotações**

Para explicar o conceito de conotação, Hayakawa (1972) utiliza como exemplo a linguagem de um comunicado. A linguagem de um comunicado é de índole instrumental, isto é, instrumental no sentido de instruir para que se faça o que tem de ser feito, mas a linguagem também se emprega na expressão direta dos sentimentos de quem fala. Do ponto de vista de quem escuta, podemos dizer que a linguagem do comunicado nos informa, mas que, os usos expressivos da linguagem nos afetam, isto é, afetam os nossos sentimentos.

Existe na linguagem um importante elemento afetivo, isto é, a aura de sentimentos agradáveis ou desagradáveis, que praticamente circunda todas as palavras. Idéias, noções, conceitos e sentimentos sugeridos pela mente possuem significado intencional e são exemplos de conotações.

Segundo Hayakawa (1972), as conotações podem dividir-se em duas espécies:

- a) conotação informativa – são os significados impessoais de uma palavra, socialmente aceitos, na medida em que se possa emprestar significados mediante palavras

adicionais. Por exemplo, ao dizermos “porco”, podemos dar as conotações informativas do porco, como: animal quadrúpede criado para se transformar em alimentação para o homem. As vezes as conotações informativas das palavras de todo dia diferem de tal maneira de lugar para lugar e de indivíduo para indivíduo que uma terminologia especial substituta, de conotações informativas mais fixas, tem de ser usada quando se deseja maior concisão (terminologia científica para determinar nomes de plantas e animais)

- b) conotação afetiva – são a aura dos sentimentos impessoais que despertam, como exemplo “porco... hum !, animal imundo e malcheiroso”. Quando estamos fortemente comovidos, exprimimos os nossos sentimentos emitindo palavras de conotações afetivas apropriadas aos mesmos, sem prestar atenção às conotações informativas que acaso tenham.

Todas as palavras, segundo os usos que se lhes dão, são dotadas de certa índole afetiva. Muitas existem mais por seu valor afetivo do que informativo. Pode-se referir àquele homem, dizendo “aquele indivíduo”, “aquela pessoa” ou “aquele gringo” e embora a pessoa mencionada possa ser a mesma em todos os casos, cada um dos termos citados revela uma diferença em nossos sentimentos para com ela.

O fato de algumas palavras despertarem, ao mesmo tempo, não apenas conotações informativas, mas também conotações afetivas, empresta uma complexidade toda peculiar a discussões que envolvam grupos religiosos, raciais, nacionais e políticos. Para muitas pessoas, a palavra “comunista” significa, simultaneamente, “alguém que é pelo comunismo” (conotação informativa) e “alguém que devia ser posto na cadeia” (conotação afetiva).

No tocante a assuntos sobre os quais existem fortes preconceitos, vemo-nos obrigados, para evitar despertá-los, a falar em termos de circunlóquio. Daí utilizar-se expressões como: “criança excepcional” ao invés de “criança retardada”, “delinquentes juvenis” ao invés de “menores criminosos” e assim por diante.

## 1.8 COMUNICAÇÃO HUMANA

Segundo Penteadó (Penteadó, 1986) “Linguagem é Comunicação. Personalidade é Comunicação”. Nesta afirmação está embutido um enfoque importante no qual pretende-se demonstrar a importância da comunicação no relacionamento analista de sistemas e usuário para se atingir uma boa análise e definição de requisitos que contribuam para o sucesso do software final a ser entregue. Cada palavra, cada gesto é ação comunicativa, assim como é Comunicação cada página de livro, folha de jornal, cada som de receptor de rádio, cada imagem de televisão.

A palavra “comunicar” vem do latim “comunicare” com a significação de “por em comum”. Comunicação é convivência; está na raiz de comunidade, agrupamento caracterizado por forte coesão, baseado no consenso espontâneo dos indivíduos.

A Comunicação humana, portanto, através da compreensão, põe idéias “em comum”. Seu grande objetivo é o entendimento entre os homens. Para que exista entendimento é necessário que se compreendam mutuamente indivíduos que se comunicam.

A individualidade é uma característica muito forte na Comunicação humana. Ela impregna a personalidade e a linguagem em cada comunicação. O que se convencionou chamar de “personalidade” imprime seu caráter a toda Comunicação humana. As palavras não significam nada por si mesmas, e só adquirem sentido, quando um ser pensante faz uso delas. A Linguagem é um método puramente humano de comunicação de idéias, emoções e desejos, através de um sistema de símbolos produzidos voluntariamente.

### 1.8.1 A fórmula de Lasswell

Apesar da individualidade ser um aspecto tão marcante na Comunicação humana, existem algumas formas consagradas pelo Prof. Lasswell da Universidade de Michigan (Penteado, 1986: p.12) através de algumas questões em benefício de sua efetividade:

Quem ?

Para a efetividade da Comunicação humana, cumpre a definição inicial de quem é o transmissor, qual o seu papel e em que capacidade se comunica.

Diz o que ?

O que comunicar depende, para a significação, de palavras bem escolhidas, organizadas em frases; palavras e frases que devem traduzir com clareza, o sentido do que se procura transmitir.

A quem ?

Na prática nem sempre acontece que transmissor e receptor falem a mesma língua e coloquem-se no mesmo plano. Pode-se dar a uma palavra, determinado significado, que o outro não compartilha. A condição social diferencia os significados das palavras e uma mesma palavra pode ter significação diferente. Com quem falo condiciona minha maneira de expressar.

Desta forma, é o receptor que condiciona a forma de Comunicação. Para ser compreendido deve-se falar a mesma língua do receptor, ater-se a seu vocabulário e preferir as expressões que lhe são mais familiares.

Através de que meio ?

A escolha do meio adequado pode, em certas circunstâncias, garantir o êxito da Comunicação humana. Formas verbais, escritas, uso de imagens, sons, gráficos, etc, fazem a diferença em situações específicas, conforme a necessidade.

Com que finalidade ?

A finalidade da Comunicação humana deve ser evidente, para prevenir distorções e mal entendidos.

Penteado (Penteado, 1986) cita 3 dependências importantes para a comunicação se tornar efetiva:

- a) A Comunicação humana depende da experiência em comum do transmissor e do receptor;
- b) A Comunicação humana depende da significação em comum dos símbolos entre transmissor e receptor;
- c) A Comunicação humana depende da atenção a mensagem.

Para serem compreensíveis, os símbolos precisam ter significados conhecidos do transmissor e do receptor. Toda palavra é um símbolo – som articulado que representa alguma coisa. Como transmitir a um selvagem, que jamais tenha visto um avião, a idéia de avião ?  
A Linguagem dos gestos, tão expressiva, só é compreensível quando esses sinais têm significação, igual para quem transmite e quem recebe.

Penteado (1986) cita 12 recomendações para a efetividade da Comunicação humana:

- 1) A comunicação humana deve ser sempre individual;
- 2) Entre transmissor e receptor, quem decide o gabarito da Comunicação humana é o receptor;
- 3) Cabe ao transmissor precisar o papel em que faz a Comunicação, assim como a seleção do meio pertinente e da forma apropriada;
- 4) A finalidade da Comunicação humana deve ser evidente e por isso depende da organização de idéias;

- 5) A linguagem utilizada na Comunicação humana deve ser compartilhada entre transmissor e receptor, fazendo comum a significação de experiências e símbolos;
- 6) Sendo a Comunicação humana, processo essencialmente dinâmico, não pode ser abandonado antes de completar-se;
- 7) Completa-se o processo, quando se observa na volta, a reação do receptor;
- 8) A compreensão nasce do significado comum e deve estar presente em todas as fases da Comunicação;
- 9) O objetivo da Comunicação humana não é necessariamente concordância e sim compreensão;
- 10) Embora imperfeita, transmissor e receptor devem esforçar-se em diminuir as deficiências da Comunicação humana;
- 11) Não é tão importante ser compreendido quanto compreender;
- 12) A Comunicação humana efetiva depende da atenção dirigida.

### **1.8.2 Comunicação humana e personalidade**

Penteado (1986) estabelece uma relação importante entre Comunicação humana e personalidade, fundamental para explicar as distorções existentes na relação analista de sistemas e usuário. Para ele, a Comunicação humana é uma forma de comportamento, uma resposta a um estímulo. Esses estímulos podem ocorrer externamente ou dentro do indivíduo; externos ou internos, produzem um impacto no sistema nervoso, sensações visuais, auditivas, tácteis, gustativas, orgânicas, etc. Sendo a Comunicação humana comportamento e estando o comportamento ligado à personalidade individual, são indispensáveis para a compreensão da mesma, alguns conhecimentos sobre personalidade.

Duas correntes da Psicologia procuram equacionar a personalidade: a primeira considera personalidade atributo e a segunda, relação. De acordo com a primeira concepção, define-se

personalidade como o produto das disposições naturais, das inclinações hereditárias, das influências psicofísicas do ambiente, da educação, com a ajuda da própria vontade. Para a segunda corrente, personalidade é a reação do indivíduo ao papel que lhe é imposto pelo grupo.

Na personalidade-atributo, as faculdades intelectuais precisam aparecer, comunicar-se. É necessário que as qualidades individuais se revelem para se estabelecer a comunicação. É a projeção dessas qualidades que importa na vida social e na realização individual. A personalidade não apenas depende da Comunicação humana; ela é a própria Comunicação humana em ação.

Na personalidade-relação, a reação do indivíduo ao papel que lhe é imposto pelo grupo, é resposta essencialmente comunicativa, isto é, reagir de determinada forma, e essa forma, seja ela qual for, será sempre uma Comunicação humana.

Tudo quanto revela a personalidade, pertence à Comunicação humana. Quando simpatizamos com alguém, todas as suas qualidades aparecem-nos iluminadas, enquanto comem na penumbra os defeitos. Quando antipatizamos, todos os defeitos individuais surgem-nos sombrios, ofuscando as qualidades.

A personalidade humana realiza uma síntese individual entre os fatores genéticos e sociais, da mesma forma que a produtividade do trabalhador dependerá do equilíbrio entre vocação e aprendizado. Na Comunicação humana, o indivíduo pode assumir uma tendência inata a comunicar-se bem, como pode apreender a desenvolver essa habilidade.

A Comunicação humana projeta a personalidade real, sublinhando defeitos e qualidades.

### **1.8.3 Obstáculos à Comunicação humana**

A Comunicação humana é essencialmente individual, e sendo individual é afetada pela personalidade, o que torna problemático o seu significado.

A Linguagem, como instrumento de Comunicação humana, sempre foi foco de causas de inadequação chegando o filósofo Bacon a referir-se ao seu caráter diabólico, afirmando que “as palavras atiram tudo à confusão, e afundam a humanidade em vãs e falazes controvérsias sem fim”.

Com o objetivo de estabelecer uma sistemática no estudo dos obstáculos à Comunicação humana, Penteadó (1986) divide em dois grandes grupos:

- 1) Obstáculos devidos à personalidade
- 2) Obstáculos devidos à linguagem.

#### **1.8.3.1 Obstáculos devidos à personalidade**

Os obstáculos devidos à personalidade nascem dos preconceitos, educação, hereditariedade, meio, experiências individuais, estado fisiológico e emocional dos participantes, inclusive os relativos à atenção individual, capacidade de concentração, etc. São os seguintes:

##### **1) Auto-suficiência**

Somos levados a extremos, pela nossa capacidade de afirmar, e por isso, nossas avaliações tomam a parte pelo todo. Para a Comunicação humana, a auto-suficiência, com todas as suas consequências, significa obstáculo por vezes intransponível: é impossível para qualquer pessoa começar a aprender o que ela julga que já sabe. “Ser consciente da própria ignorância é o primeiro passo para a sabedoria”, escreveu o filósofo Disraeli.

## 2) Congelamento das avaliações

Cada um de nós tende a ser um inesgotável repositório de avaliações congeladas, baseadas em que as pessoas não mudam. E não apenas as pessoas, mas as coisas, todas as coisas. As avaliações congeladas passam por cima das mudanças que se operam nas pessoas, nos grupos e nas coisas. Aceitando que tudo muda, fugimos ao risco das controvérsias seculares sobre a “natureza humana”, para aceitar como manifesto da natureza humana apenas a sua permanente mudança.

## 3) Comportamento humano: aspectos objetivos e subjetivos

A Comunicação humana baseia-se na concepção da personalidade projetiva, na evidência de que na sociedade humana, o homem precisa “vender” a sua personalidade. Nada há de pernicioso em uma venda, definida como o esforço consciente de aceitação social. O que se procura na Comunicação humana é a coincidência entre a personalidade subjetiva, a que existe interiormente e a objetiva, que se manifesta exteriormente. A personalidade está ligada ao comportamento, à sua reação individual e ao meio social.

## 4) Geografite

Sofre de “geografite” o indivíduo que se impressiona mais com os “mapas” do que com os “territórios”. Os mapas correspondem aos sentimentos, imaginações, palpites, hipóteses, pressentimentos, preconceitos. Territórios são os objetos, as pessoas, as coisas, os acontecimentos. É o contraste do símbolo com o significado, onde o primeiro vale mais do que o segundo. São pessoas facilmente sugestionáveis.

A Comunicação humana pede uma atitude positiva diante da vida. Por atitude positiva queremos expressar uma forma de equilíbrio na avaliação individual das coisas, dos acontecimentos e das pessoas.

#### 5) Tendência à complicação

Existe uma tendência do indivíduo a restringir seu ponto de vista sobre um problema, tornando a solução para o problema muito mais complicada. Na tendência à complicação, personalidade e linguagem estão juntas, e contribuem para agravar o problema. Como prova disto, temos os tabus e superstições, que influenciam negativamente as pessoas a encontrar caminhos mais claros e objetivos para seus problemas.

#### 1.8.3.2 Obstáculos devidos à linguagem

A linguagem antepõe inúmeras dificuldades à efetividade da Comunicação humana: a polissemia, os pleonasmos, os exageros, as negativas que afirmam e as afirmativas que negam, as graduações, os rodeios, as gírias, e toda a complexidade do significado. Os principais obstáculos provenientes da Linguagem são:

##### 1) Confusão entre fatos e opiniões

Fato é acontecimento; é coisa ou ação feita. Opinião é modo de ver, é conjectura.

Podemos nos agarrar, de tal maneira, às nossas opiniões que, para nós, elas acabam se transformando em fatos; deixamos de distinguir, entre as nossas idéias, as que traduzem fatos e as que significam opiniões. As opiniões podem ser refutadas por fatos.

## 2) Confusão entre inferências e observações

Inferir é deduzir pelo raciocínio. Observar é olhar atentamente. As afirmações devidas a inferências e observações são extremamente difíceis de serem distinguidas, porque a estrutura da linguagem não oferece qualquer indicação de diferença.

As inferências nos oferecem certezas relativas. As observações nos oferecem certezas absolutas.

## 3) Descuido nas palavras abstratas

É muito comum o descuido no emprego das palavras abstratas. A palavra não pode nascer livremente na consciência do homem, fora da realidade objetiva. Não há conceito sem uma coisa que o anteceda. Embora as coisas não sejam palavras, há um nexos entre a palavra e a coisa à qual a palavra se refere.

Definindo as palavras, antes de discuti-las, perceberemos que, ou estamos de acordo, ou pensamos em coisas diferentes, embora lhes emprestemos a mesma forma.

## 4) Desencontros

Existe uma linguagem dos sentimentos, subjetiva e uma linguagem da razão, objetiva. Nada chega ao intelecto sem passar primeiro pelos sentidos. As palavras vêm até a nossa mente, impregnadas de subjetividade; há sempre algo nosso na mais racional das idéias. Os desencontros acontecem na Comunicação humana quando não se compartilha o significado das palavras:

- a) As mesmas palavras significam coisas diferentes
- b) Palavras diferentes significam coisas iguais.

#### 5) Indiscriminação

A indiscriminação ocorre quando falhamos em reconhecer variações, nuances ou diferenças; quando não nos sentimos capazes, ou apenas não queremos distinguir, diferenciar coisas aparentemente iguais. Indiscriminação é negligenciar as diferenças, enquanto se sublinha as semelhanças.

#### 6) Polarização

Há polarização quando tratamos os contrários, como se fossem contraditórios. Polarização é a tendência a reconhecer apenas os extremos, negligenciando as posições intermediárias.

A Comunicação humana não procura harmonizar pontos de vista ou encontrar denominador comum para os opostos; esforça-se para que se não negligencie os intermediários.

#### 7) Falsa identidade baseada nas palavras

A falsa identidade baseada nas palavras é uma ameaça ao entendimento entre os homens, pela força esmagadora dos veículos da Comunicação em massa. Coisas iguais a uma terceira são iguais entre si, não retratam necessariamente uma situação real. Ficam apenas no campo das palavras. A partir de apenas uma característica compartilhada entre duas coisas, salta-se a conclusão que todas as demais características são idênticas.

#### 8) Polissemia

A Polissemia acontece pelo fato de se manter uma multiplicidade de sentidos para cada palavra. A palavra engana porque a Linguagem é um método puramente humano de Comunicação de idéias, emoções e desejos, por meio de um sistema de símbolos produzidos voluntariamente. A palavra engana, porque muitos objetos recebem nomes inadequados, entre o sinal e a coisa assinalada. As palavras enganam porque não significam nada por si mesmas. Só representam algo, quando uma criatura humana faz uso delas.

## **CAPÍTULO 2 – PROCESSO DE REQUISITOS: ESTUDO DA ANÁLISE DE REQUISITOS SOB O PONTO DE VISTA DA CIÊNCIA DA INFORMAÇÃO E DA ENGENHARIA DE SOFTWARE.**

Neste capítulo, com base no processo de desenvolvimento de sistemas de softwares, discutem-se as principais dificuldades encontradas nos modelos adotados pela Engenharia de Software para a definição de requisitos e, identificam-se, sob o ponto de vista das teorias e processos comunicacionais e informacionais emprestados da Ciência da Informação, os dispositivos de controle que podem ser utilizados nos processos de produção do conhecimento e de sua organização na etapa de análise de requisito.

A análise de requisitos se baseia na aquisição bem sucedida de informações. Esta aquisição normalmente esbarra em informações conflituosas fornecidas pelo usuário-cliente em relação com outras exigências declaradas por pessoas que também estão envolvidas no processo e noções de funções e desempenho do sistema que estão em desacordo com as metas estabelecidas originalmente.

Se levado em consideração todas estas dificuldades, o analista poderia criar um conjunto de perguntas ao usuário de forma a obter uma compreensão maior do problema e o usuário verbalizar melhor suas percepções sobre possíveis soluções. Combinando isto com elementos de solução de problemas, negociação e especificação, pode-se minimizar estas dificuldades naturais do processo comunicacional.

## **2.1 ANÁLISE DE REQUISITOS NA ENGENHARIA DE REQUISITOS**

A Engenharia de requisitos de software é definida como todas as atividades voltadas para a identificação de requisitos do usuário; análise dos requisitos para conduzir a requisitos adicionais; documentação dos requisitos como uma definição e validação dos requisitos documentados em relação às necessidades atuais do usuário (Saiedian, 2000). Como parte desta atividade, a análise de requisitos é um processo específico de coleta, extração e determinação dos requisitos de software.

Da perspectiva do usuário, uma boa análise de requisitos resulta em ter um melhor entendimento de suas necessidades e obrigações. A partir disto, estas especificações poderão efetivamente criar soluções alternativas e entender as implicações das decisões envolvidas.

Da perspectiva do desenvolvedor, é necessário construir uma definição de alto nível do problema a ser resolvido, entendendo aqui alto nível como uma definição com um nível de detalhamento maior e mais próximo do usuário. Isto dá segurança que a solução está sendo desenvolvida para o problema certo e é possível de se colocar na prática.

O mais importante é que um bom processo de identificação e análise de requisitos deve construir uma visão comum do problema e uma concepção da solução do software entre usuário e desenvolvedor.

Para ajudar a descrever os requisitos, Pfleeger (2004) classifica-os de duas maneiras: funcionais e não funcionais.

Um requisito funcional descreve uma interação entre o sistema e seu ambiente, ou seja, descreve como o sistema deve se comportar, considerando um certo estímulo. Descreve-se o que o sistema fará, sem discutir sobre que computador específico e linguagem de programação serão utilizados ou estruturas de dados internas envolvidas.

Um requisito não funcional descreve uma restrição no sistema que limita nossas opções para criar uma solução para o problema. Essas restrições geralmente limitam nossa seleção referente à linguagem de programação, plataforma operacional e ferramentas de implementação.

Os requisitos funcionais e não funcionais são identificados com os usuários-cliente de uma maneira formal e cuidadosa, pois nem sempre estes usuários conseguem descrever com exatidão o que querem ou precisam e os analistas nem sempre são capazes de entender todos os aspectos do negócio do cliente.

### **2.1.1 Conhecimento necessário aos analistas de sistemas para análise de requisitos**

Uma das maiores dificuldades no processo de análise de requisitos é identificar o conhecimento tácito em poder do usuário. Como já discutido no item 1.3, sem a externalização do conhecimento tácito não é possível atingir a dimensão do problema pela perspectiva do usuário, o que normalmente compromete os resultados do projeto de desenvolvimento do software.

Sato (1995), define 3 tipos de conhecimento necessários aos desenvolvedores para garantir uma performance melhor do projeto:

- a) Conhecimento básico/comum na área de estudo;
- b) Conhecimento do design, baseado na experiência do desenvolvedor;
- c) Conhecimento sobre os objetivos do trabalho de design atual.

Quanto mais conhecimento formal (conhecimento “a”) o analista de sistemas possui sobre a área de negócio objeto do projeto, mais facilidade ele terá para se comunicar com o usuário, e isso é um fator fundamental para criar uma boa definição de requisitos. Porém, cada área de negócios contém suas particularidades, de uma organização para outra, e por isso, apenas o

conhecimento formal e básico sobre essa área passa a não ser suficiente. Para isto, a experiência do analista de sistemas em projetos que envolvam esta mesma área de negócios em empresas diferentes passa a ser muito importante. Este tipo de experiência (conhecimento “b”) é que irá facilitar a identificação das características particulares da área alvo do projeto, mesmo que não seja explicitamente demonstrado pelo usuário. Porém, algumas destas características particulares fazem parte do conhecimento tácito do usuário (conhecimento “c”) e de difícil explicitação e são de fundamental importância para o projeto. Mesmo com toda expertise do analista de sistemas, não é possível garantir, que será levado em consideração na análise de requisitos os detalhes de negócio que se encontram no domínio do conhecimento tácito do usuário.

Na análise de requisitos existem 2 dimensões importantes a serem consideradas pelo analista de sistemas: dimensão explícita e dimensão tácita (Sato, 1995). Na dimensão explícita, temos o requisito de conhecimento explícito, que consiste nas proposições formais feitas pelo usuário. Na dimensão tácita, temos o requisito de conhecimento tácito, que consiste nas intenções e expectativas do usuário. Por serem muito ambíguos, estes requisitos não são descritos formalmente e acabam ficando de fora da definição final de requisitos.

A experiência do analista no domínio do problema em projetos anteriores juntamente com a cooperação do usuário podem ajudar a compreender e explicitar o requisito tácito de conhecimento.

### **2.1.2 Participantes chaves no processo de definição de requisitos**

O processo de requisitos sofre várias influências de participantes do processo de desenvolvimento de sistemas. Na visão clássica (método “cascata”) do processo de desenvolvimento de sistemas, a análise de requisitos ocorre apenas nas etapas superiores do

processo (figura 2.1), não havendo interferências do restante das etapas. Esta visão, apesar de ainda ser muito utilizada nos projetos de desenvolvimento de sistemas, tem sido confrontada com outros paradigmas da Engenharia de software, que estabelecem revisões em várias etapas do processo e por isto sofrendo várias influências. Desta forma, é importante demonstrar quem são os participantes chaves deste processo e como podem influenciar a identificação de requisitos.

[A1] Comentário: Onde está esta figura ???

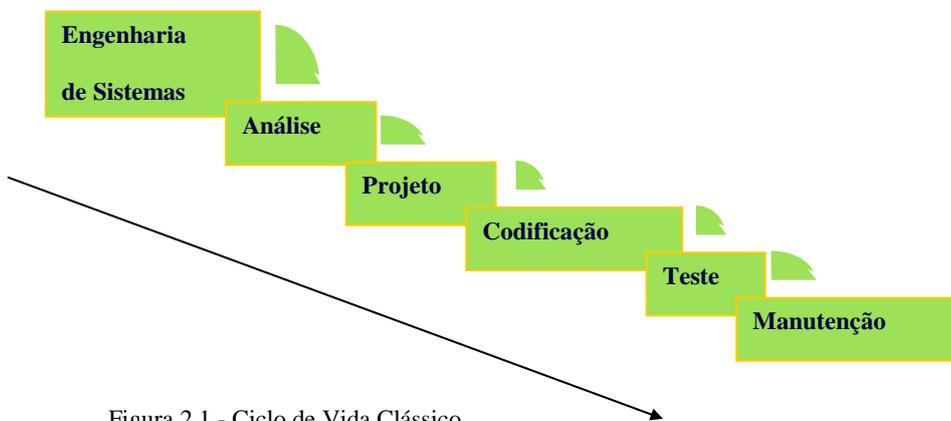


Figura 2.1 - Ciclo de Vida Clássico

Saiedian (2000), define 2 grupos de participantes chaves: do lado do usuário e do lado do desenvolvedor.

Do lado do usuário, os participantes chaves podem ser classificados como:

- **Comprador (buyer)** – pessoas responsáveis pela contratação e pagamento do software. Suas responsabilidades principais incluem controle de cronogramas, orçamentos (budgets) e os compromissos assumidos em contrato. Normalmente são os contatos do projeto com o desenvolvedor.
- **Usuário Final (end user)** – são as pessoas que irão usar o sistema desenvolvido. Suas preocupações estão relacionadas com a usabilidade do sistema bem como a

disponibilidade real. Eles estão mais familiarizados com os processos de trabalho que o analista precisará levantar.

- **Especialistas de domínio (domain experts)** – são as pessoas especializadas no ambiente sistêmico onde o software irá trabalhar. Estão ligados às considerações técnicas do sistema e suas interfaces.
- **Mantenedores de Software (maintainers software)** – em alguns casos, o cliente mantém uma equipe de engenheiros de software interna que fica responsável pela manutenção futura do projeto, que deixa de ser responsabilidade do desenvolvedor.

Do lado do desenvolvedor, os participantes-chaves podem ser classificados como:

- **Gerente de projetos (project managers)** – são os responsáveis pela venda do produto e marketing, tendo contato direto com o cliente.
- **Engenheiro de Requisitos (requirements engineers)** – normalmente são engenheiros de software (ou analistas de sistemas) que são responsáveis pela identificação e documentação dos requisitos.
- **Engenheiro de Software (software engineers)** – são os responsáveis pela produção do software, desenho e prototipação. É mais comum encontrarmos nesta categoria os analistas de sistemas, analistas de banco de dados e programadores.
- **Testadores (testers)** – são os responsáveis pelo desenvolvimento dos testes necessários para validar o software.

Em alguns projetos, é muito comum que o analista de sistemas responsável pela definição de requisitos, não tenha contato direto com o usuário final, recebendo especificações por parte dos compradores ou de membros da equipe de IT (tecnologia de informação) do cliente. Este fato compromete a análise de requisitos, sendo fonte de diversos atritos entre desenvolvedores e usuários.

Mesmo quando a equipe de IT do cliente faz todo o trabalho de análise de requisitos para agilizar o processo, o contato com o usuário final é fundamental para o desenvolvedor, pois entender o contexto do usuário irá garantir requisitos mais bem elaborados e sistemas entregues de acordo com a expectativa do usuário.

### **2.1.3 Problemas mais comuns na identificação de requisitos**

A partir das dificuldades comunicacionais demonstradas no item 1.8, apoiadas pela obra de Penteadó (1986), dos problemas da linguagem no item 1.7, apoiados pela obra de Hayakawa, (1972) e do trabalho de Saiedian (2000) sobre engenharia de requisitos, será exposto a seguir, alguns problemas comuns que ocorrem no relacionamento analista de sistemas e usuário que dificultam a identificação de requisitos na etapa de análise.

#### **2.1.3.1 Comunicação Pobre**

O que comunicar, é tão importante, em relação, ao como comunicar. Muitas vezes se o usuário percebe sinais de falta de interesse ou “não escuta” de seus pontos de vista, isto pode prejudicar a comunicação, tendendo o usuário a restringir o que tem a expressar.

Uma presença muito marcante e dominadora por parte do desenvolvedor também tende a inibir o usuário a expressar suas verdadeiras necessidades e detalhes do processo de trabalho. Neste caso, é melhor uma postura mais “humilde”, que deixe o usuário mais confortável para se expressar.

O uso de intermediários no levantamento de requisitos, tais como pessoas de vendas, marketing, ou Tecnologia de Informação, contribuem para mensagens distorcidas. Estas pessoas não têm uma completa visão do problema, estando mais preocupadas com o gerenciamento de metas e cronogramas, o que pode comprometer o levantamento.

### 2.1.3.2 Resistência

Resistência é um processo emocional em que a pessoa oferece algum tipo de oposição. Resistência a novas idéias é muito comum em qualquer processo de implementação. Ninguém gosta de escutar que, o que está fazendo tem formas melhores de ser feito ou que está obsoleto. O Engenheiro de requisitos deve ser capaz de reconhecer e abrandar estas resistências, destacando questões importantes para o cliente que não foram levantadas adequadamente. Saiedian (Saiedian, 2000) classifica algumas formas de resistências mais comuns:

- a) Time Resistance – as pessoas nunca têm tempo para se encontrar com o analista.
- b) Overload Resistance – Quanto mais informação o analista levanta para o usuário, mais ele identifica que está faltando. Nunca é o suficiente.
- c) Silence Resistance – O usuário não reage ou não responde ao que o analista diz.
- d) Impracticality Resistance – O usuário sempre lembra que vive no mundo real e que as proposições do analista são impossíveis de se colocar em prática.
- e) Compliance Resistance – O que quer que o analista faça está bom. O usuário não expressa suas reservas sobre as proposições do analista.

### 2.1.3.3 Articulação e especialidade para problemas

Um obstáculo comum na comunicação de assuntos técnicos é o uso de terminologias que não são compreendidas por uma das partes. Profissionais adoram fazer uso de jargões e gírias que podem causar confusão e intimidar a outra parte na compreensão real do problema. Outro problema é o desenvolvedor desqualificar o usuário, assumindo que ele seja incapaz de fazer um bom julgamento do que se está ofertando ou incapaz de trabalhar com sistemas de alta tecnologia. Quando se começa um processo de desenvolvimento com esta visão, o resultado pode ser ofensivo ao usuário e não atingir as metas estabelecidas. Vale a pena educar o

usuário para as técnicas mais modernas de desenvolvimento, apresentação e navegação do software e ganhar seu apoio para atingir todas as necessidades.

#### 2.1.3.4 Perspectivas diferentes para o problema

Muitas vezes o analista de sistemas tem pouca ou nenhuma experiência como usuário final da aplicação a ser desenvolvida. Isto resulta em uma solução mais tecnológica não contextualizada no problema a ser resolvido. O analista tende a oferecer um produto que não atenda a atual necessidade do usuário, mas que esteja sintonizado com o que há de melhor em pacotes de hardware e software.

## **2.2 ANÁLISE DE REQUISITOS COM O FOCO NO USUÁRIO**

O desenvolvimento de requisitos de sistema centrado no usuário é uma parte do processo de desenvolvimento de sistemas que se inicia com o usuário e suas necessidades, mais do que com a tecnologia. Essa abordagem cria uma capacitação do usuário de estar no controle de tecnologias avançadas para valorizar as habilidades humanas, superando as limitações humanas. Para alcançar estes objetivos, os desenvolvedores devem entender os usuários e suas tarefas.

Segundo Minocha (2000), “Requisitos do sistema são declarações de situações de uso do sistema”. Numa abordagem centrada no usuário, estas situações não deveriam apenas descrever as necessidades do usuário e seus desejos para um sistema, mas deveriam também capturar o contexto de uso no qual o sistema irá trabalhar e o impacto que o sistema proposto teria nas condições de trabalho atual do usuário.

O processo de análise de requisitos deveria ser participativo de forma que usuários e analistas estejam envolvidos num processo colaborativo de identificação e desenvolvimento dos requisitos do sistema.

Na metodologia tradicional do processo de desenvolvimento de sistema, a cascata, discutido no item 2.1.2, o conceito de análise de requisitos é uma etapa independente do ciclo de vida do desenvolvimento do sistema. Porém o desenvolvimento de requisitos é um processo contínuo de validações e especificações ao longo do processo de desenvolvimento, que deve ser reavaliado a todo o momento.

O desenvolvimento de requisitos no processo centrado no usuário inicia-se com a definição tradicional do usuário e de requisitos funcionais. Entretanto requisitos surgem, mudam e são redefinidos quando eles são derivados de interações do analista e usuário para capturar o contexto do sistema em uso.

Clark e Moreira (1999), propõem um modelo de definição de requisitos centrado no usuário, partindo de representações das interações entre o ambiente e o sistema, definidas como eventos. A idéia é utilizar o processo de análise orientada a objetos para formalizar o comportamento que um sistema oferece para o seu ambiente (system centred mode).

Como o modelo é formal ele não será ambíguo e pode ser analisado para garantir que internamente será consistente. Como é executável, podemos usar protótipos para validar o modelo em relação aos requisitos e para demonstrar seu comportamento ao usuário que poderia prematuramente alimentar o sistema para verificar se os seus requisitos foram corretamente coletados.

Clark e Moreira (1999) destacam que o processo de criar especificações formais é válido mesmo quando o resto do desenvolvimento não é um processo formal. Ele força o analista a considerar todos os casos de uso e eles não podem escapar com generalismos vagos. Entretanto, construir especificações formais de requisitos não é uma tarefa fácil, pois há um “gap” enorme entre os requisitos informais e a notação e estrutura de uma definição.

O modelo centrado no usuário apresentado pelos autores especifica o comportamento observável de um sistema como um conjunto de agentes de cenário (agent views). Um agente de cenário descreve uma forma na qual pode ser usado e é formalmente especificado no LOTOS em termos de interações que podem ocorrer entre o agente e o sistema. LOTOS é uma técnica de descrição formal desenvolvida pela ISO (International Standard Organization) para definições do padrão Open Systems Interconnections (OSI), possuindo uma linguagem formal orientada a evento que é executável.

Os agentes de cenário possuem regras similares aos casos de uso, diagrama clássico da análise orientada a objetos. A grande diferença é que os agentes de cenário incluem apenas as interações entre um agente e o sistema, não incluindo as conseqüentes interações dos objetos

internos do sistema. Os agentes são usuários do sistema que podem ser seres humanos, dispositivos de hardware ou outros sistemas de software.

Eamonn O'Neill (1999) apresenta um modelo cooperativo de desenvolvimento e entendimento compartilhado que define o usuário como a fonte primária de conhecimento de tarefas e requisitos. Ao invés de utilizar o usuário como uma fonte passiva de informação, esse modelo propõe uma interação maior entre usuários e desenvolvedores. Porém, é necessário criar um vocabulário compartilhado entre usuário e desenvolvedor e um diálogo direto do desenvolvedor com o usuário, para criar um entendimento compartilhado.

O ponto principal é envolver analistas e usuários nas atividades iniciais de desenvolvimento fazendo-os sócios no conhecimento geral que é construído nestas etapas. Criar uma rápida interação em torno do desenvolvimento e uma constante reavaliação, é o desafio proposto para se chegar a um software pronto que atenda plenamente aos requisitos do projeto.

## **2.3 TÉCNICAS PARA MELHORAR A ANÁLISE DE REQUISITOS**

Embora todo o processo de desenvolvimento de sistemas venha fazendo cada vez mais uso de ferramentas tecnológicas, que têm levado a processos mais automatizados, o processo de definição de requisitos continua sendo um processo centrado no indivíduo, baseado na experiência humana. É necessário reconhecer e entender os requisitos de conhecimento localizados na dimensão tácita, se deseja-se compreender os reais requisitos do usuário.

Neste capítulo, procurou-se relacionar algumas técnicas e recomendações extraídas de alguns autores que se preocuparam com este tema, bem como de nossa interpretação de experiências pessoais, que procuram melhorar a definição de requisitos em relação ao contato pessoal de analistas e usuários.

### **2.3.1 Características dos requisitos para garantia de qualidade**

Pfleeger (2004) define sete questões para avaliar as características importantes que um requisito deve ter para garantia de qualidade na definição de requisitos. São elas:

1. Os requisitos estão corretos?

Analistas e clientes devem analisar os requisitos para garantir que eles foram definidos sem erros.

2. Os requisitos são consistentes ?

Verificar se não existem requisitos ambíguos ou conflitantes. Considera-se dois requisitos inconsistentes se for impossível satisfazer aos dois simultaneamente.

3. Os requisitos estão completos ?

O conjunto de requisitos é completo, se todos os possíveis estados, mudanças de estado, entradas, produtos e restrições estiverem descritos por algum requisito. Uma descrição de requisitos é internamente completa, se não houver referências indefinidas entre eles.

4. Os requisitos são realistas ?

Deve-se verificar se o que o cliente está pedindo pode realmente ser feito pelo sistema, não gerando falsas expectativas.

5. Cada requisito descreve algo que é necessário para o cliente ?

Algumas vezes, um requisito limita os desenvolvedores desnecessariamente ou inclui funções que não estão diretamente relacionadas com o problema em questão. Deve-se rever os requisitos, de forma a manter somente requisitos que atuam diretamente na resolução do problema do cliente.

6. Os requisitos podem ser verificados ?

Devemos ser capazes de planejar testes que demonstrem que os requisitos foram satisfeitos.

7. Os requisitos podem ser rastreados ?

Cada função do sistema deve poder ser rastreada para um conjunto de requisitos que a exige.

A idéia destas questões é funcionar como um checklist para verificar cada possível requisito, aprimorando e fazendo modificações, à medida que avançamos no projeto de desenvolvimento do software.

### **2.3.2 Comunicação Efetiva e coleta de informações**

Manter o foco na conversação com o usuário, demonstrando atenção, tomando nota, fazendo questões e pedindo para repetir alguns pontos, demonstram que realmente está se escutando o interlocutor e isto é muito importante para começar bem o relacionamento com o usuário.

Saiedian (2000), destaca algumas recomendações para facilitar o processo de comunicação com o usuário:

- *“Fazer conclusões precipitadas sobre o que o usuário está dizendo, deixa-nos com o foco apenas nos comentários que dão suporte a essa conclusão, desprezando detalhes importantes do contexto.*
- *Normalmente, os usuários não entendem o que eles querem requisitar do analista para ajudá-los em seu trabalho. Nas reuniões de levantamento de informações eles estão em desvantagem em relação ao analista, pois eles apenas conhecem seu trabalho e nada sabem sobre projetos de desenvolvimento de sistemas. Frequentemente, o usuário fica estressado quando o analista pede para que documente suas necessidades. Uma maneira de ajudar neste processo é o analista oferecer algo que se assemelhe com o que ele quer. Estabelecer pontos chaves (focal point) de suas necessidades ajudam a esclarecer e organizar os seus requisitos. Porém, isto deve ser feito com cuidado, pois o usuário pode entender que o analista está impondo um conjunto de requisitos para adaptar o usuário em sua solução, que já existe.”*

O analista deve ter uma postura cética na coleta de informações, pois normalmente o usuário não diz o que ele realmente quer. Naomi Karten (Karten, 1994), oferece algumas sugestões para extrair informações dos usuários:

- a. “Nunca faça suposições. Repita questões já perguntadas, refaça-as para conseguir diferentes perspectivas.*
- b. Pergunte para esclarecer. Tenha certeza de entender a linguagem utilizada pelo usuário para descrever o problema. Não fique com receio de admitir que não sabe alguma coisa. Pergunte tudo o que for necessário para formar um bom entendimento das expectativas do usuário.*

- c. Colete informações de múltiplas fontes. Nós podemos conseguir um entendimento mais amplo das diversas necessidades apresentando questões similares para pessoas em entrevistas separadas. Pessoas com diferentes responsabilidades ajudam a preencher lacunas que uma determinada fonte criou.*
- d. Procure inconsistências no diálogo. O usuário está respondendo suas questões com respostas que você quer ouvir ou de sua própria perspectiva ?*
- e. Manter o foco das questões no processo é melhor do que nos indivíduos, pois evita atitudes defensivas.*
- f. Evite questões do tipo sim e não. Envolver o usuário no questionamento do processo, criando um consenso sobre as questões formuladas.*
- g. O aumento no número de linhas de comunicação com o usuário em relação as linhas convencionais (entrevistas, observações, levantamentos) ajudam no processo de definição de requisitos. Atualmente pelo avanço tecnológico disponível na maioria das empresas, é muito comum utilizar-se de recursos tais como: e-mails, chats, vídeo-conferências, etc. “““.*

### **2.3.3 Modelos de representação gráfica que buscam interação com o usuário**

O uso de artefatos não tradicionais para caracterizar o ambiente do usuário, pode ajudar a deixar mais claro os detalhes dos requisitos do usuário, como exemplo: diagramas do ambiente do usuário, hierarquias do problema encontrado, matrizes mostrando os usuários-chaves e suas áreas, listas de benefícios e capacidades desejadas e “story boards” dos produtos.

O uso de vídeos tem sido uma forma eficiente de documentar as atividades e o ambiente do usuário. Isto se torna possível aos usuários falar e serem escutados em seus próprios termos, sem intermediários.

Os vídeos podem ser compartilhados facilmente por outras pessoas da equipe de desenvolvimento de forma a criar novos pontos de vistas sobre o entendimento do problema. É muito mais eficiente e confere maior autoridade um vídeo demonstrando partes do trabalho do usuário do que relatórios resumindo isto.

#### **2.3.4 Técnicas de participação do cliente-usuário.**

Quanto mais cedo o usuário participar da definição de requisitos maior possibilidade de sucesso terá o desenvolvimento do projeto. Existem detalhes do ambiente do trabalho do usuário que estão fora dos padrões regulares de fazer as coisas. Estes detalhes conferem um nível de flexibilidade importante de se levantar, que só a experiência do dia-a-dia pode demonstrar. Uma forma de se conseguir levantar isto é o analista se colocar como aprendiz do usuário. Conversar sobre seu trabalho enquanto se está executando garante que detalhes importantes não serão omitidos ou confundidos.

#### **2.3.5 Etnografia Rápida e Inquisições Contextuais**

A Etnografia rápida e inquisições contextuais são 2 técnicas bastante utilizadas para coleta de dados do usuário, seus requisitos e o contexto do sistema em uso.

Etnografia consiste na observação das atividades diárias do usuário e suas interações com o sistema em uso.

Inquisições contextuais envolvem a observação e a entrevista dos usuários em seus locais de trabalho para identificar requisitos de tarefa, especificações de sistema, fatores ambientais e características do usuário que devem ser suportadas pelo novo sistema.

Ambas as técnicas possuem um forte conteúdo comunicacional, pois envolvem uma série de esclarecimentos verbais que o usuário é obrigado a passar para o analista. Porém, faz muita diferença se o analista já possui ou não experiência com área do usuário e suas atividades, para obter um bom resultado no uso destas técnicas. Quanto menos o analista conhecer, mais possibilidades de interpretações errôneas e subjetividades na sua definição de requisitos.

## **2.4 DISCUSSÃO DOS PARADIGMAS DA ENGENHARIA DE SOFTWARE EM RELAÇÃO A INTERAÇÃO COM O USUÁRIO – ASPECTOS COMUNICACIONAIS**

A Engenharia de Software estabelece mecanismos para se obter economicamente um software que seja confiável e que funcione eficientemente. Segundo Pressman (1995), a Engenharia de software abrange um conjunto de três elementos fundamentais – métodos, ferramentas e procedimentos – que irão possibilitar o controle do processo de desenvolvimento do software.

Os métodos de engenharia de software proporcionam os detalhes de “como fazer” para construir o software. As ferramentas proporcionam apoio automatizado ou semi-automatizado aos métodos. Os procedimentos constituem o elo de ligação que mantém juntos os métodos e as ferramentas e possibilitam o desenvolvimento racional do software.

A Engenharia de Software compreende um conjunto de etapas que envolvem métodos, ferramentas e os procedimentos, sendo também denominadas de “paradigmas de engenharia de software” ou “ciclo de vida do software”, envolvendo todo o processo de desenvolvimento de software, desde a concepção até a implementação, entrega, utilização e manutenção.

Neste item, vão ser discutidos os paradigmas mais populares da Engenharia de software, com enfoque na sua capacidade de interação entre o analista e o usuário, aspectos comunicacionais e pontos críticos na definição de requisitos.

### **2.4.1 Modelo Cascata**

O modelo cascata, também conhecido como ciclo de vida clássico, foi um dos primeiros modelos propostos para regular o desenvolvimento de um software e ainda é amplamente utilizado. Ele se caracteriza por uma sucessão de estágios, em que um estágio deve terminar antes de o próximo começar. Dessa forma, somente quando todos os requisitos forem

expostos pelo usuário e tiverem sido completamente analisados e documentados pelo analista de sistemas é que se poderia iniciar o próximo estágio, Projeto de sistema, pela equipe de desenvolvimento, conforme vemos na figura 2.4.1.

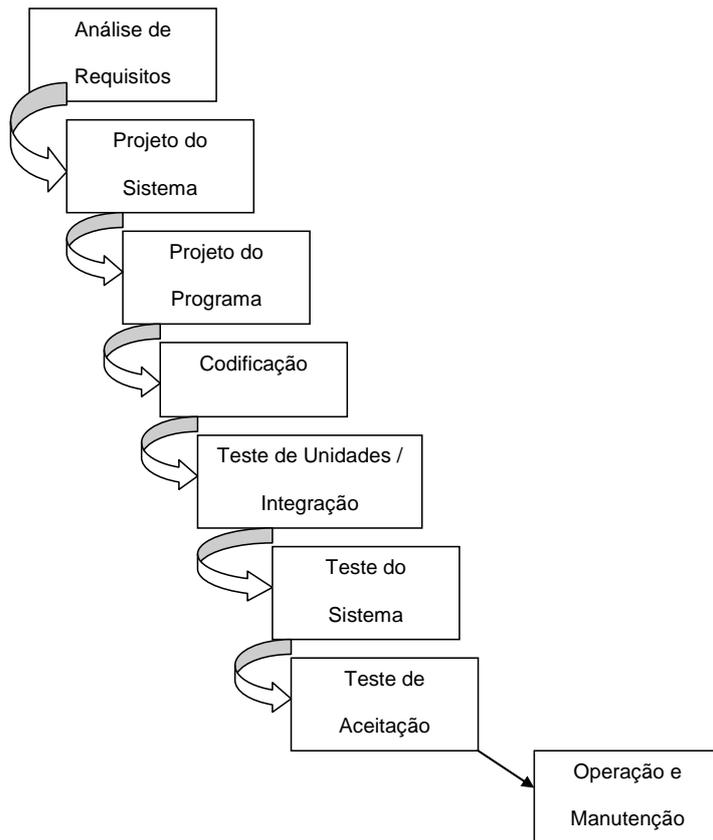


Figura 2.4.1 – Modelo Cascata

Pressman (1995) aponta 3 problemas principais quanto a aplicabilidade do modelo cascata:

1. Projetos reais raramente seguem o fluxo sequencial que o modelo propõe. Alguma interação sempre ocorre e traz problemas na aplicação do paradigma.
2. Muitas vezes é difícil para o cliente declarar todas as exigências explicitamente. O modelo cascata exige isso e tem dificuldade de acomodar a incerteza natural que existe no começo de muitos projetos.

3. O cliente deve ter paciência. Uma versão de trabalho do software não estará disponível até um ponto tardio do cronograma do projeto e um erro crasso, se não detectado até que o programa de trabalho seja revisto, pode ser desastroso.

Pfleeger (2004) destaca que o principal defeito do modelo cascata é a falha em tratar o desenvolvimento de software como um processo de solução de problemas, derivado do mundo do hardware, apresentando o desenvolvimento do ponto de vista do fabricante de hardware. O software não é desenvolvido desta maneira, em vez disso, ele evolui à medida que o problema é melhor compreendido e as alternativas avaliadas. Desta forma, o software é resultado de um processo de criação e não de fabricação.

Neste modelo, a única etapa em que se estabelece uma interação formal com o usuário é na etapa de análise de requisitos. Todas as demais etapas posteriores, até o teste de aceitação, excluem o usuário do processo, o que causa um baixo nível de interação e falhas de entendimento, que se levadas adiante irão comprometer o resultado final do processo. Não existe foco no usuário no decorrer do processo e esta falta de comunicação impede o usuário de acompanhar a evolução do projeto e poder se manifestar em tempo de ajudar a identificar possíveis distorções nos requisitos definidos na primeira etapa.

Os requisitos devem ser re-analisados e verificados junto ao usuário em todas as etapas do processo e isto não é coberto pelo modelo cascata.

#### **2.4.2 Modelo em V**

O modelo em V é uma variação do modelo cascata, que demonstra como as atividades de teste estão relacionadas com a análise e o projeto. Como mostra a figura 2.4.2, a codificação forma o vértice do V, com a análise e o projeto à esquerda, e o teste e a manutenção à direita.

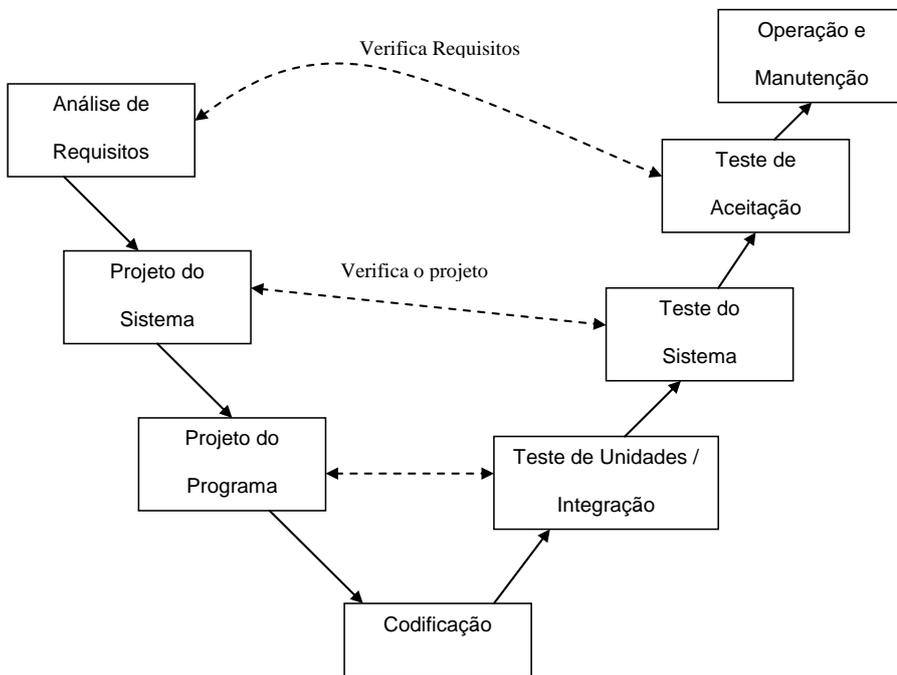


Figura 2.4.2 – Modelo em V

Pfleeger (2004) destaca que o modelo em V sugere que os testes de unidade e de integração também podem ser utilizados para verificar o projeto do programa, isto é, os programadores e a equipe de testes deveriam garantir que todos os aspectos do projeto do sistema foram implementados corretamente no código. A conexão entre os lados esquerdos e direito do modelo em V implica que, caso sejam encontrados problemas durante a verificação e validação, o lado esquerdo do V pode ser executado novamente para corrigir e melhorar os requisitos, o projeto e a codificação, antes da execução das etapas de teste do lado direito do V, ou seja, o modelo em V torna mais explícitas algumas iterações e repetições do trabalho, ocultas no modelo cascata.

Outro ponto importante é que este modelo sugere a possibilidade de se utilizar atividades de prototipação para melhorar o entendimento dos requisitos, do projeto do sistema e do

programa, na medida em que se pode construir protótipos do sistema que ajudarão o usuário e analista a verificar se o resultado final está de acordo com os objetivos inicialmente definidos. Este modelo cria um maior nível de interação com o usuário, em relação ao modelo cascata, exigindo uma participação maior do usuário na validação e verificação dos requisitos do projeto, em relação as etapas iniciais de desenvolvimento, que são fundamentais para o prosseguimento do desenvolvimento do software com maior segurança e integridade. Porém, apesar da significativa melhora nos níveis de interação entre analista e usuário, este modelo ainda releva a participação do usuário em etapas que o projeto já avançou demasiadamente, de forma que se identificado alguma distorção no entendimento dos requisitos, muito trabalho já terá sido desenvolvido, o que irá ocasionar um grande retrabalho.

### **2.4.3 Prototipação**

Segundo Pressman (1995) uma abordagem de prototipação à engenharia de software pode representar a melhor escolha quando, por exemplo, o cliente definiu um conjunto de objetivos gerais para o software, mas não identificou requisitos de entrada, processamento ou saída. Ou então, o desenvolvedor pode não ter certeza da eficiência de um algoritmo, da adaptabilidade de um sistema operacional ou da forma que a interação homem-máquina deve assumir.

A prototipação é um processo que capacita o desenvolvedor a criar um modelo do software que será implementado. O modelo pode assumir uma das três formas:

1. Um protótipo em papel ou modelo baseado em microcomputador PC que retrata a interação homem-máquina de uma forma que capacita o usuário a entender quanta interação ocorrerá.
2. Um protótipo de trabalho que implementa um subconjunto da função exigida do software desejado.

3. Um programa existente que executa parte ou toda a função desejada, mas que tem outras características que serão melhoradas em um novo esforço de desenvolvimento.

Conforme podemos observar na figura 2.4.3, a prototipação inicia-se com a coleta dos requisitos. O desenvolvedor e o cliente reúnem-se e definem os objetivos globais para o software, identificam as exigências conhecidas e esboçam as áreas em que uma definição adicional é obrigatória. Ocorre então, a elaboração de um "projeto rápido" o qual concentra-se na representação daqueles aspectos do software que serão visíveis ao usuário. O projeto rápido leva à construção de um protótipo que é avaliado pelo cliente/usuário e é usado para refinar os requisitos para o software a ser desenvolvido. Um processo de iteração ocorre quando é feita uma "sintonia fina" do protótipo para satisfazer as necessidades do cliente, capacitando, ao mesmo tempo, o desenvolvedor a compreender melhor aquilo que precisa ser feito.

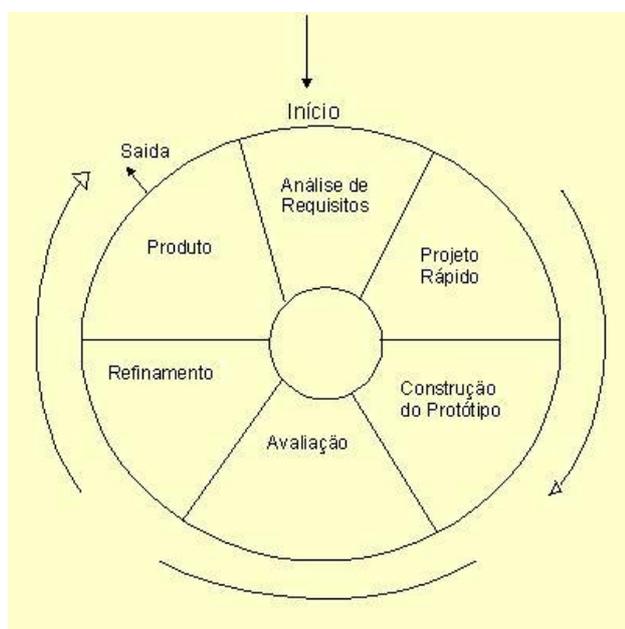


Figura 2.4.3 - Ciclo de vida da Prototipação

Pressman (1995) destaca que como o ciclo de vida clássico, a prototipação como paradigma da engenharia de software pode ser problemática pelas seguintes razões:

1. O usuário vê aquilo que parece ser uma versão de trabalho do software, desconhecendo que o protótipo se mantém unido, sem saber que na pressa de colocá-lo em funcionamento, não levamos em consideração a qualidade global do software e a manutenibilidade a longo prazo.
2. A fim de colocar um protótipo em funcionamento rapidamente, o desenvolvedor muitas vezes faz concessões de implementação. Um sistema operacional ou linguagem de programação imprópria pode ser usado, simplesmente, porque está à disposição e é conhecida; um algoritmo ineficiente pode ser implementado simplesmente para demonstrar capacidade.
3. Depois de algum tempo, o desenvolvedor pode familiarizar-se com essas opções e esquecer-se de todas as razões pelas quais são inadequadas. A opção menos que ideal se tornou então parte integrante do sistema.

Ainda que possam ocorrer problemas, a prototipação é um paradigma eficiente da engenharia de software. O importante é definir-se as regras logo no começo, isto é, o desenvolvedor e o usuário devem ambos concordar que o protótipo seja construído para servir como um mecanismo a fim de definir os requisitos. Ele será descartado (pelo menos em parte) e o software real será projetado, levando-se em conta a qualidade e a manutenibilidade.

Este modelo traz a vantagem de um alto grau de interação entre analista e usuário, na medida em que a interação entre requisitos e projeto é constante. Os requisitos podem ir se completando e se formalizando ao longo do projeto, através de constantes revisões em cima do desenvolvimento do protótipo. Isto causa um grau de certeza muito maior na aceitabilidade

do sistema, pois desde o início o usuário estará interagindo com o analista para validação dos requisitos, vindo surgir aos poucos o sistema planejado.

#### 2.4.4 Modelo Espiral

Pressman (1995) define o modelo espiral como um modelo desenvolvido para abranger as melhores características tanto do ciclo de vida clássico como da prototipação, acrescentando ao mesmo tempo, um novo elemento, a análise de riscos, que falta a esses paradigmas. O modelo representado na figura 2.4.4, define quatro importantes atividades representadas pelos quatro quadrantes da figura:

1. Planejamento: determinação dos objetivos, alternativas e restrições
2. Análise de riscos: análise de alternativas e identificação/resolução dos riscos
3. Engenharia: desenvolvimento do produto no nível seguinte
4. Avaliação feita pelo cliente: avaliação dos resultados da engenharia

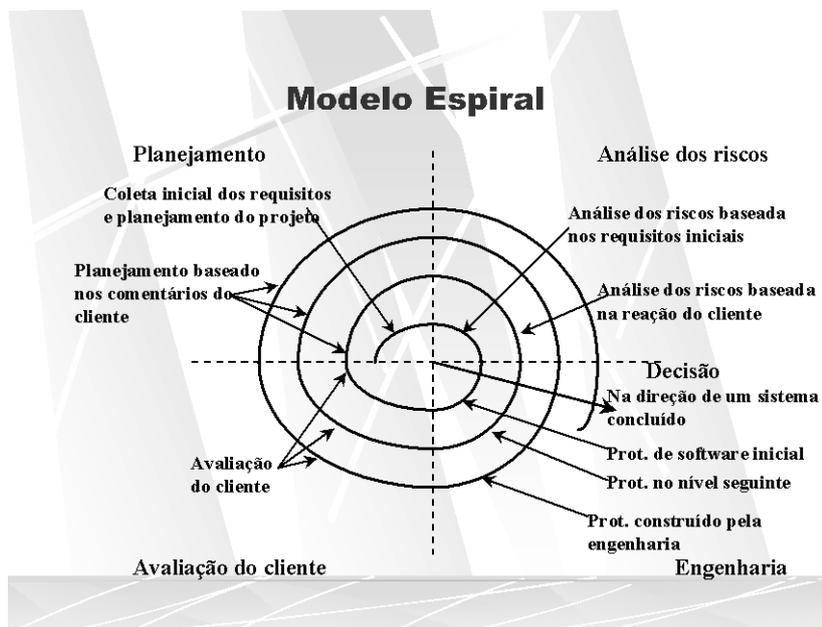


Figura 2.4.4 – Modelo Espiral

Começando com os requisitos e um projeto inicial para o desenvolvimento, o processo insere uma etapa para avaliar riscos e protótipos alternativos antes de ser produzido um documento de concepção de operações, a fim de descrever em alto nível, como o sistema deverá funcionar. A partir desse documento, um conjunto de requisitos é especificado e detalhado para assegurar que os requisitos que estão tão completos e consistentes quanto possíveis. Conseqüentemente, a concepção das operações é o produto da primeira iteração, e os requisitos são o principal produto da segunda iteração. Na terceira iteração, o desenvolvimento do sistema produz o projeto e na quarta, habilita os testes.

Durante o primeiro giro ao redor da espiral, os objetivos, alternativas e restrições são definidos e os riscos são identificados e analisados. Se a análise dos riscos indicar que há incertezas nos requisitos, a prototipação pode ser usada no quadrante da engenharia para ajudar tanto o desenvolvedor quanto o usuário. Simulações e outros modelos podem ser usados para definir ainda mais o problema e refinar os requisitos.

O usuário avalia o trabalho de engenharia e apresenta sugestões para modificações. Baseada na entrada do usuário, ocorre a fase seguinte de planejamento e a análise de riscos. Em cada arco da espiral, a conclusão da análise de riscos resulta numa decisão de prosseguir ou não prosseguir. Se os riscos forem muito grandes, o projeto pode ser encerrado.

O modelo espiral usa uma abordagem evolucionária, capacitando o desenvolvedor e o usuário a entender e reagir aos riscos em cada etapa evolutiva. Este modelo usa a prototipação como um mecanismo de redução de riscos, mas o que é mais importante é que o desenvolvedor pode utilizar a prototipação em qualquer etapa da evolução do projeto.

Este modelo é o que apresenta o maior nível de interação entre analista e usuário, pois força uma reavaliação dos requisitos em todas as etapas do processo de desenvolvimento. Não se

inicia uma nova etapa sem se reavaliar a etapa atual, em relação aos requisitos iniciais e riscos inerentes da etapa. O usuário tem uma participação ativa e constante desde o início do projeto e mesmo nas etapas mais técnicas, ainda assim se torna necessário uma reunião com o usuário para sua avaliação. O nível de comunicação é grande e todos os problemas que discutimos no item 2.1.3 se tornam críticos neste modelo, exigindo uma atenção especial por parte do analista para superar estes problemas.

#### **2.4.5 Interação Humano Computador**

A área de Interação Humano-Computador (IHC) tem por objetivo principal fornecer aos pesquisadores e desenvolvedores de sistemas *explicações e previsões para fenômenos de interação usuário-sistema e resultados práticos para o design da interface de usuário* (ACM SIGCHI, 1992). Com teorias a respeito dos fenômenos envolvidos seria possível prever antecipadamente se o sistema a ser desenvolvido satisfaz as necessidades de usabilidade, aplicabilidade e comunicabilidade dos usuários. Para isto, estudos de IHC visam desenvolver modelos teóricos de desempenho e cognição humanos, bem como técnicas efetivas para avaliar a usabilidade. Mais recentemente algumas propostas têm enfatizado que além de usabilidade, as aplicações devem buscar atingir aplicabilidade e comunicabilidade, oferecendo ao usuário artefatos fáceis de usar, aplicar e comunicar (Souza, 2000).

IHC é uma área multidisciplinar, que envolve disciplinas como: Ciência da Computação; Psicologia Cognitiva; Psicologia Social e Organizacional; Ergonomia ou Fatores Humanos; Linguística; Inteligência Artificial; Filosofia, Sociologia e Antropologia; Engenharia e *Design*.

No contexto de IHC devemos considerar quatro elementos básicos: *o sistema, os usuários, os desenvolvedores e o ambiente de uso* (domínio de aplicação). Estes elementos estão

envolvidos em dois processos importantes: a *interação usuário-sistema* e o *desenvolvimento do sistema*. O *currículo* proposto para IHC identifica cinco enfoques para o estudo destes elementos e para a sua aplicação na melhoria dos processos de desenvolvimento e de interação usuário-sistema (Souza, 2000). Para cada um destes focos, diferentes disciplinas proporcionam os estudos teóricos que podem ser aplicados ao desenvolvimento. São eles:

- *design* e desenvolvimento do hardware e software: estudo de tecnologias de dispositivos de entrada e saída; e tecnologias de software, como ambientes gráficos e virtuais.
- estudo da capacidade e limitação física e cognitiva dos usuários: considera estudos de ergonomia para avaliar limites de esforço físico do usuário, e estudos de psicologia e ciência cognitiva sobre a capacidade humana de memorização, raciocínio e aprendizado.
- instrumentação teórica e prática para o *design* e desenvolvimento de sistemas interativos: envolve o conhecimento teórico a respeito dos fenômenos envolvidos; modelos para o processo de desenvolvimento que descrevam as etapas necessárias e como devem ser conduzidas; diretrizes, técnicas, linguagens, formalismos e ferramentas de apoio a estas etapas.
- modelos de interfaces e do processo de interação usuário-sistema: para desenvolver modelos abstratos do processo de interação compatíveis com as capacidades e limitações físicas e cognitivas dos usuários.
- análise do domínio e de aspectos sociais e organizacionais: para avaliar o impacto que o contexto onde está inserido o usuário exerce sobre seus conhecimentos, sua linguagem e suas necessidades.

A abordagem proporcionada pela IHC cria um novo paradigma na Engenharia de Software que leva em consideração a preocupação com dois aspectos importantes na relação com o usuário: a **interação e a usabilidade**. A interação é um processo que engloba as ações do usuário sobre a interface de um sistema, e suas interpretações sobre as respostas reveladas por esta interface. A usabilidade de um sistema é um conceito que se refere à qualidade da interação de sistemas com os usuários.

No que diz respeito aos aspectos comunicacionais abordados neste trabalho, a IHC tem um enfoque diferente, enfocando comunicabilidade do ponto de vista da relação sistema-usuário. A comunicabilidade de um sistema é a sua propriedade de transmitir ao usuário de forma eficaz e eficiente as intenções e princípios de interação que guiaram o seu *design*. O objetivo da comunicabilidade é permitir que o usuário, através da sua interação com a aplicação, seja capaz de compreender as premissas, intenções e decisões tomadas pelo projetista durante o processo de *design*. Quanto maior o conhecimento do usuário da lógica do *designer* embutida na aplicação, maiores suas chances de conseguir fazer um uso criativo, eficiente e produtivo da aplicação.

Sem dúvida, os pontos críticos discutidos no item 2.1.3 têm um grande peso na abordagem proporcionada pela IHC, mas pelo fato de atuar de forma multidisciplinar, terá mais facilidade de resolvê-los.

## **CAPÍTULO 3 – PROPOSTA DE UM MODELO DE ANÁLISE DE REQUISITOS BASEADO NAS TEORIAS DA CIÊNCIA DA INFORMAÇÃO**

Levando-se em consideração o que se discutiu nos capítulos 1 e 2, neste capítulo descreve-se um modelo para auxiliar na análise de requisitos, apoiado pelas teorias da Ciência da Informação. Este modelo aborda especificamente a etapa de identificação e análise de requisitos do processo de desenvolvimento de software, onde os aspectos comunicacionais são fatores críticos de sucesso para a sua melhor finalização.

### **3.1 APRESENTAÇÃO DO MODELO**

A proposta para o modelo a ser apresentado aborda 3 fases distintas dentro do processo de análise de requisitos, visando estabelecer uma forma melhor de comunicação com os usuários:

1. Conhecimento Prévio;
2. Comunicação;
3. Especificação.

A fase 1, **Conhecimento Prévio**, é de fundamental importância para o bom andamento da análise de requisitos. Como discutido no item 1.6 deste trabalho, a partir da abordagem proposta por Vakkari (1998), e no item 2.1.1 por Sato (1995), o conhecimento prévio do analista sobre o problema pode ajudar a extrair do usuário as informações necessárias para garantir uma identificação de requisitos com maior acuracidade, sintonizada com o problema do usuário de forma a garantir maior aderência do sistema aos objetivos inicialmente determinados.

A fase 2, **Comunicação**, apoiando-se nos conceitos discutidos nos itens 1.7 com Hayakawa (1972) e 1.8 com Penteadó (1986) apresenta-se uma proposta para resolver as questões

cognitivas e de comunicação direta entre o analista e o usuário, envolvendo os seguintes pontos críticos:

- a) efetividade da comunicação
- b) obstáculos devidos à personalidade
- c) obstáculos devidos à linguagem
- d) coleta de informações

A fase 3, **Especificação**, com base nos conceitos propostos por Saiedian (2000) e Pfleeger (2004) no item 2.3, definiu-se formas de apresentação dos requisitos ao usuário, de maneira que ele possa ajudar no processo de validação e aceitação destes requisitos. Os principais pontos envolvidos nesta fase são:

- a) Modelos de representação do problema que busquem interação com o usuário
- b) Prototipação
- c) Documento de definição de requisitos

## 3.2 MODELO PROPOSTO

O modelo proposto constitui uma série de recomendações e definições para se conduzir um processo de análise de requisitos de forma a procurar eliminar os principais problemas encontrados nesta etapa, já discutidos no capítulo 2. Estas proposições serão divididas por fases, conforme apresentado no item 3.1, criando um conjunto formal para facilitar sua aplicação.

### 3.2.1 Fase 1 – Conhecimento prévio

Antes de iniciar a identificação de requisitos, o analista deverá estudar a área de negócio do usuário requisitante, caso ainda não tenha tido nenhuma experiência com essa área, procurando conhecer os principais processos envolvidos.

Na análise de requisitos existem 2 dimensões importantes a serem consideradas pelo analista de sistemas: dimensão explícita e dimensão tácita (Sato, 1995). O grande problema diz respeito à dimensão tácita, que não é descrita formalmente e normalmente acaba ficando de fora da definição final de requisitos. A experiência do analista no domínio do problema em projetos anteriores, juntamente com a cooperação do usuário podem ajudar a compreender e a explicitar o requisito tácito de conhecimento.

Quanto mais informações se conseguir reunir sobre os processos a serem atingidos, melhor proveito se fará no contato necessário com o usuário. Nesse quesito, algumas opções objetivas e efetivas são aconselháveis, tais como:

- a) **Conversar com outros analistas** que já possuem experiência com a área de negócio, reunindo informações sobre os principais processos e pontos críticos além de se familiarizar com as terminologias da área;

- b) **Trabalhar temporariamente como um “estagiário”** do usuário, vivenciando todos os detalhes do dia-a-dia dos processos a serem atingidos;
- c) **Visitar empresas semelhantes**, com a finalidade de conhecer como funciona a área de negócio a ser atingida;
- d) **Levantar soluções já existentes no mercado**, que poderiam facilitar a implementação do sistema ou mesmo direcionar a solução a ser construída;
- e) **Busca de manuais de normas e procedimentos da área a ser atingida**;
- f) **Busca de livros que abordem os processos de negócios da área a ser atingida**;
- g) **Contato com especialistas da área de negócio a ser atingida**;

O objetivo principal dessas recomendações é facilitar a identificação das características particulares da área alvo do projeto, de forma a não só criar um conhecimento formal e básico sobre esta área, mas permitir que se facilite a identificação do conhecimento tácito do usuário, fundamental para o sucesso do projeto.

A fase 1 do modelo proposto está diretamente relacionada com as demais áreas, de forma que o conhecimento prévio irá facilitar não só o processo de comunicação com o usuário, mas também a própria especificação, daí sua importância no contexto.

### **3.2.2 Fase 2 – Comunicação**

Como já discutido nos capítulos 1 e 2, a relação entre analista e usuário no processo de análise de requisitos apresenta grande conteúdo comunicacional, causa de frequentes distorções na definição dos requisitos.

O que comunicar, é tão importante em relação ao como comunicar. Muitas vezes se o usuário percebe sinais de falta de interesse ou “não escuta” de seus pontos de vista, isto pode prejudicar a comunicação, tendendo o usuário a restringir o que tem a expressar.

Nesta fase do modelo, destacam-se algumas recomendações e propostas para os principais pontos críticos, causadores destas distorções.

#### 3.2.2.1 Efetividade da Comunicação

Visando tornar a comunicação entre analista e usuário mais efetiva, sugere-se as seguintes recomendações:

- a) **Manter o foco na conversa com o usuário**, demonstrando atenção, tomando nota, fazendo questões e pedindo para repetir alguns pontos, demonstram que realmente está se escutando o interlocutor e isto é muito importante para começar bem o relacionamento com o usuário.
- b) **O que comunicar depende de palavras bem escolhidas**, organizadas em frases; palavras e frases que devem traduzir com clareza, o sentido do que se procura transmitir. Deve-se utilizar uma linguagem de uso comum entre analista e usuário, evitando-se uso de termos que não sejam de domínio comum.
- c) **Evitar fazer conclusões precipitadas** sobre o que o usuário está dizendo, pois isto nos deixa com o foco apenas nos comentários que dão suporte a essa conclusão, desprezando detalhes importantes do contexto.

#### 3.2.2.2 Obstáculos devidos à personalidade

Segundo Penteadó (1986), a Comunicação humana é uma forma de comportamento, uma resposta a um estímulo. Esses estímulos podem ocorrer externamente ou dentro do indivíduo; externos ou internos, produzem um impacto no sistema nervoso, sensações visuais, auditivas,

táteis, gustativas, orgânicas, etc. Sendo a Comunicação humana comportamento e, estando o comportamento ligado à personalidade individual, alguns obstáculos surgem em decorrência disso. Os obstáculos devidos à personalidade nascem dos preconceitos, educação, hereditariedade, meio, experiências individuais, estado fisiológico e emocional dos participantes, inclusive os relativos à atenção individual, capacidade de concentração, etc. É importante destacar algumas recomendações que podem ajudar a superar este tipo de obstáculo:

- a) **Evitar assumir uma postura auto-suficiente.** É impossível para qualquer pessoa começar a aprender o que ela julga que já sabe.
- b) **Não manter avaliações congeladas.** As avaliações congeladas passam por cima das mudanças que se operam nas pessoas, nos grupos e nas coisas. Aceitando que tudo muda, fugimos ao risco de assumirmos pré-conceitos sobre o usuário e o processo a ser definido.
- c) **Procurar “vender” sua personalidade ao usuário.** Nada há de pernicioso em uma venda, definida como o esforço consciente de aceitação social. Manter uma postura simpática ajuda a derrubar barreiras importantes no processo de análise de requisitos.
- d) **Assumir uma atitude positiva.** A Comunicação humana pede uma atitude positiva diante da vida. Por atitude positiva subentende-se uma forma de equilíbrio na avaliação individual das coisas, dos acontecimentos e das pessoas.

#### 3.2.2.3 Obstáculos devidos à linguagem

Segundo Penteadó (1986), os obstáculos devidos a linguagem nascem de inúmeras dificuldades à efetividade da Comunicação humana como: a polissemia, os pleonasmos, os exageros, as negativas que afirmam e as afirmativas que negam, as graduações, os rodeios, as

grias, e toda a complexidade do significado. Algumas recomendações podem ajudar a evitar este tipo de obstáculo:

- a) **Flexibilizar suas opiniões na conversa com o usuário.** Pode-se agarrar, de tal maneira às nossas opiniões que, elas acabam se transformando em fatos, deixando de distinguir, entre as nossas idéias, as que podem realmente traduzir fatos.
- b) **Evitar o uso de palavras abstratas ou de dupla interpretação.** Definindo as palavras, antes de discuti-las, percebe-se que, ou estamos de acordo, ou pensamos em coisas diferentes, embora lhes emprestemos a mesma forma.
- c) **Cuidado com a indiscriminação na comunicação com o usuário.** Indiscriminação é negligenciar as diferenças, enquanto se sublinha as semelhanças.
- d) **Procurar detalhar melhor as características apresentadas pelo usuário** na descrição de um processo. A partir de apenas uma característica compartilhada entre duas coisas, evidencia-se a conclusão que todas as demais características são idênticas.

#### 3.2.2.4 Coleta de Informações

A coleta de informações é uma das atividades principais na etapa de análise de requisitos. Ela inicia um processo de comunicação importante entre analista e usuário que irá consolidar uma definição de requisitos sustentável para se iniciar o projeto de desenvolvimento do software.

Normalmente, os usuários não entendem o que eles querem requisitar do analista para ajudá-los em seu trabalho. Nas reuniões de levantamento e coleta de informações eles estão em desvantagem em relação ao analista, pois eles apenas conhecem seu trabalho e nada sabem sobre projetos de desenvolvimento de software. Algumas recomendações podem facilitar muito este processo:

- a) **Expor ao usuário a abrangência do projeto,** justificando que o nível de informação necessário implica em detalhes que vão além do dia-a-dia do usuário. Isto facilita o

entendimento do usuário para requisições que o analista pode fazer, que aparentemente não digam respeito diretamente ao problema do usuário.

- b) **Estabelecer pontos chaves das necessidades do usuário** para ajudar a esclarecer os seus requisitos.
- c) **Adotar uma postura cética em relação as informações fornecidas pelo usuário.** Pergunte de formas diferentes as mesmas questões para avaliar se as respostas não se confundem.
- d) **Coletar informações de múltiplas fontes.** Pode-se conseguir um entendimento mais amplo das diversas necessidades apresentando questões similares para pessoas em entrevistas separadas.
- e) **Manter o foco das questões no processo** e não nos indivíduos de forma a evitar atitudes defensivas por parte dos entrevistados.
- f) **Evitar questões do tipo sim e não.** Envolve o usuário no questionamento do processo, criando um consenso sobre as questões formuladas.

### 3.2.3 Fase 3 – Especificação

A identificação dos requisitos é uma parte especialmente importante do processo de desenvolvimento de software. Deve-se determinar o que os usuários e os clientes realmente querem, com o objetivo de entender o problema quando ainda não se dispõe de uma solução.

Saiedian (2000), destaca que um bom processo de identificação e análise de requisitos deve construir uma visão comum do problema e uma concepção da solução do software entre usuário e desenvolvedor.

O objetivo desta fase é sugerir recomendações para definir formas de apresentação dos requisitos ao usuário, de maneira que ele possa ajudar no processo de validação e aceitação

destes requisitos, criando maior envolvimento e participação do usuário no resultado final da definição de requisitos.

#### 3.2.3.1 Modelos de representação do problema que busquem interação com o usuário

Criar uma forma de expor ao usuário o levantamento efetuado pelo analista do problema alvo é muito importante para criar uma forma de participação efetiva do usuário na compreensão do problema e nas possibilidades de solução. A utilização apenas do modelo narrativo, descrevendo de forma textual os processos envolvidos no problema, é sabidamente causador de profundas ambigüidades. Diagramas complexos, tais como o DFD (diagrama de fluxo de dados), voltados para especificações técnicas da equipe de desenvolvimento não são adequados para se interagir com o usuário. Desta forma, o objetivo é ter-se uma representação do problema que permita uma maior interação entre analista e usuário. Algumas recomendações podem ajudar neste sentido:

- a) **Criar uma hierarquia do problema.** Em conjunto com o usuário defina níveis e sub-níveis para demonstrar os pontos chaves do problema que deverão participar dos requisitos.
- b) **Utilizar vídeos para demonstrar como o processo operacional alvo está funcionando,** de forma a deixar mais claro os detalhes do processo que podem ser problemas em potenciais a serem resolvidos.
- c) **Construir matrizes relacionando as áreas envolvidas do problema** com seus respectivos usuários e operações chaves do processo, facilitando identificar prioridades e gargalos.
- d) **Construir fluxogramas hierárquicos,** em nível macro, utilizando-se de terminologias comuns ao usuário e textos explicativos em cada objeto do fluxo.

### 3.2.3.2 Prototipação

A prototipação é um processo que capacita o analista para criar um modelo do software que será implementado. Desta forma é feita a elaboração de um "projeto rápido" (resumo do projeto principal) o qual concentra-se na representação daqueles aspectos do software que serão visíveis ao usuário. O projeto rápido leva à construção de um protótipo que é avaliado pelo usuário e é usado para refinar os requisitos para o software a ser desenvolvido. A prototipação permite profunda participação do usuário junto ao analista de forma a facilitar o entendimento de ambos sobre o melhor caminho para a solução do problema e formalização dos requisitos.

No modelo aqui proposto, a prototipação tem um papel muito importante para ajudar na análise e definição dos requisitos, pois irá permitir a diminuição de falhas de entendimento e expectativas sobre o escopo do projeto.

Algumas recomendações para a realização de protótipos:

- a) **Começar criando protótipos que demonstrem como as informações entrarão e sairão do sistema proposto.** É muito importante ao usuário conseguir visualizar como irá ocorrer a entrada de dados no sistema de forma a conseguir identificar se alguma informação importante estará faltando e o nível de dificuldade para se conseguir realizar esta tarefa. Visualizando também as saídas de dados do sistema, se consegue perceber se algo importante ficou de fora, ajudando a refinar os requisitos.
- b) A partir de uma hierarquia das funções esperadas do software para a solução do problema, **criar um protótipo que demonstre como estas funções estarão sendo atendidas sem com isto executar as funções mais complexas.** É importante que o usuário tenha uma idéia de como as principais funções do sistema estarão sendo sistematizadas sem com isto entrar no detalhe de sua programação, apenas exibindo os seus resultados.

- c) **Reprogramar e refazer o protótipo tantas vezes quanto forem necessárias** para satisfazer o usuário. É importante que ele esteja convencido de que o caminho a ser seguido é certo e seguro, mesmo que detalhes pouco significativos tenham que ser considerados.

### 3.2.3.3 Documento de definição de requisitos

O documento de definição de requisitos é escrito em um nível apropriado para o usuário e em termos compreensíveis para este. Esta documentação é feita em linguagem natural do usuário e isto normalmente causa diferentes interpretações, entre o analista e o usuário, daí a importância de entrar na esfera do usuário para se garantir que o documento terá o nível de esclarecimento necessário para que todos os participantes se entendam.

Este documento deve conter algumas partes importantes que são fundamentais para respaldar o entendimento do analista e do usuário, bem como o alcance da solução. São elas:

#### **1. Solicitação / Problema**

Solicitação do usuário, em sua linguagem, do problema / requisição. Esta descrição deve ser de acordo com o que o usuário identificou nas reuniões de levantamento inicial.

#### **2. Descrição da Solicitação / Problema**

Descrever o problema / requisição na linguagem do analista. Itemizar de acordo com os pontos-chaves identificados do problema.

Aqui deve vir a descrição do problema ou os Requisitos devidamente itemizados / numerados. Em ambos os casos, esse campo contém os requisitos na visão do analista, ou seja, não deve ser copiada aqui a descrição do problema do usuário no campo 1, mas sim uma interpretação do problema já sob a visão do analista.

### **3. Escopo**

Descrever a abrangência da solicitação / problema.

Aqui se deve descrever a abrangência da solução. Não devem ser mencionados programas ou tabelas a serem alteradas, mas o escopo lógico da solução (do tipo: que casos esse projeto ATENDE e que casos esse projeto NÃO ATENDE, quando o projeto vai ser aplicado, etc). Deve ser mencionado qual a correção definitiva da causa original do problema, ou seja, qual a ação que está sendo tomada para evitar a repetição do problema.

### **4. Impacto Técnico**

Aqui podem vir uma primeira análise técnica do projeto. Coisas que permitam dimensionar o que está sendo feito (e.g. vamos criar uma tabela nova, uma aplicação de manutenção dessa tabela, e 3 relatórios que combinem dados dessa tabela com dados previamente existentes no sistema. Além disso, tem-se que modificar nosso processamento de notas fiscais para levar em conta os dados dessa tabela. Tem-se ainda impacto na performance, com aumento estimado em 1 segundo no processamento de cada nota - isto é, mudança de performance em coisas que já existem ). Aqui também cabe mencionar o impacto lógico do projeto (e.g. esse projeto terá impacto na geração de Nota Fiscal e na geração de arquivos para clientes através de Packing Lists).

### **5. Riscos Técnicos**

Riscos possíveis decorrentes da solução do problema.

Aqui devem vir, se houver, os riscos inerentes ao projeto (e.g. estamos tratando apenas determinados casos de erro. O que acontecerá se outros casos de erro acontecerem ? Ou o

projeto está abrindo uma brecha de processo ?). Riscos com recursos humanos, computacionais, custos também devem ser considerados.

Probabilidade e Impacto devem ser classificados em Alto, Médio e Baixo.

## **6. Solução**

Atividades / passos adotados para solucionar o problema. Itemizar correspondentemente aos itens da “Descrição da Solicitação/Problema” – item 2.

## **7. Esforço Despendido**

Tempo efetivamente utilizado para solucionar o problema. Destacar se parte desse tempo foi usada em preparação – e.g. treinamento para realização das tarefas.

Em caso de quaisquer atrasos em relação às datas comprometidas com os usuários no momento da definição, devem ser enviados e-mails ou realizadas reuniões notificando e justificando os atrasos para todos os envolvidos no projeto, e cópias desses e-mails / atas de reuniões devem ser anexadas nessa seção.

## **8. Testes – Desenvolvedor**

Aqui novamente deve ser garantida a correspondência com os itens de Requisitos (correspondência com os itens do campo 2 - Descrição da Solicitação / Problema). Não esquecer de anexar Evidências dos Testes aos campos "Resultados Obtidos". É importante, ao descrever o Caso de Teste, de fazer testes REPETÍVEIS - com um nível de detalhe suficiente para que outra pessoa possa reexecutá-lo, e com os "Resultados Esperados" explícitos, para que uma outra pessoa que execute o teste consiga avaliar se o resultado está correto ou não.

### 8.1 Caso de Teste

Faça referência aqui ao número do requisito – da seção “Descrição da Solicitação/Problema” – sendo testado

### 8.2 Resultado Esperado

Descreva o resultado que deve ocorrer para cada caso de teste.

### 8.3 Resultado Obtido

Descreva o resultado obtido para cada caso de teste.

### **3.3 FINALIZAÇÃO DO MODELO**

Como ficou demonstrado, a ênfase deste modelo está sobre as pessoas envolvidas na etapa de análise de requisitos. Esta etapa não é um processo estruturado como outras etapas do desenvolvimento de software, onde é possível se estabelecer uma certa linearidade. Exatamente por isto, não é um modelo que se instala ou se auto-implanta, pois a experiência do analista de sistemas é fundamental para se conduzir este modelo.

A fase mais crítica do modelo é a fase 2 – Comunicação. Esta fase, por apresentar muitos aspectos subjetivos, pode gerar interpretações errôneas do modelo por parte do analista, implicando em uma definição de requisitos inconsistentes. Por isto, apresenta-se a seguir uma tabela, que tem como objetivo facilitar a interpretação e aplicação do modelo. Destacaram-se as principais recomendações de cada fase. Para cada recomendação, se apresentam duas colunas para se responder SIM ou NÃO. Cada recomendação que esteja assinalada como SIM, significa que o analista já tomou o cuidado de planejá-la e executá-la a contento, não deixando de atender aos principais pontos do modelo. Recomendações assinaladas como NÃO, implicam em uma tarefa que ainda deve ser executada, ficando resguardado a sua futura implementação. Isto irá possibilitar ao analista rever os pontos que ainda estão pendentes.

<b>FASES</b>	<b>SIM</b>	<b>NÃO</b>
<u>CONHECIMENTO PRÉVIO</u>		
Levantar subsídios junto a outros analistas		
Trabalhar como um estagiário do usuário		
Efetuar benchmarking sobre a área de negócio		
Verificar soluções existentes no mercado		
Obter manuais de normas e procedimentos da área		
Obter livros que abordem os processos de negócios da área		
Fazer contato com especialistas da área		
<u>COMUNICAÇÃO</u>		
Foco na conversação com o usuário		
Linguagem comum entre analista e usuário		
Descongelar avaliações sobre o tema		
Manter postura simpática com o usuário		
Estabelecer pontos chaves das necessidades do usuário		
Coletar informações de múltiplas fontes		
Manter o foco das questões no processo e não nos indivíduos		
<u>ESPECIFICAÇÃO</u>		
Hierarquizar o problema		
Construir matrizes de áreas do problema com operações chaves		
Construir fluxograma hierárquico das funções do processo		
Criar protótipo das funções esperadas		

Tabela 1 – Facilitando a aplicação do modelo proposto

Na experiência de colocar o modelo proposto em prática, obteve-se contribuições muito significativas da troca de informações com os usuários do projeto, que inicialmente não sabiam que estavam participando da validação de um modelo para um trabalho de mestrado. Apenas se apresentou uma diretriz de como o projeto seria desenvolvido, no que dizia respeito a etapa de análise de requisitos, com a condição de que estas diretrizes estavam em aberto para poderem ser reavaliadas e alteradas de acordo com a necessidade de se aperfeiçoar o modelo para atingir com mais precisão os objetivos do projeto. A participação ativa dos usuários, conseguida com muitas dicas que estão na fase 2 do modelo, foi fundamental para a coleta de informações importantes na definição e revisão do modelo proposto. Tinha-se a preocupação de não impor um modelo rígido e fechado aos usuários do projeto, de forma a permitir sua adaptação e correção ao longo da sua aplicação, sem com isto causar a impressão de que se estava conduzindo um projeto sem rumo, sem certezas de onde se queria chegar. Isto só foi conseguido graças ao bom relacionamento que se criou entre o analista, autor deste trabalho, e os usuários, fato fundamental para se chegar ao modelo final. Contribuiu também o fato de se ter um cronograma flexível e adequado a condução do projeto sem as pressões inerentes da falta de tempo e da busca imediata de soluções aos problemas do usuário.

Foi possível verificar, que alguns requisitos identificados com a ajuda do modelo proposto foram fundamentais para o resultado final do projeto. Se estes requisitos não estivessem claros desde o início, o resultado final do software não teria atendido os usuários do projeto e teria-se criado uma forte tensão entre os coordenadores do projeto.

Ficou claro, que planejando-se um tempo maior nas atividades de análise e identificação de requisitos consegue-se um desenvolvimento de software mais sintonizado com as expectativas do usuário. Comparando-se com os cronogramas tradicionais deste tipo de projeto, acrescentou-se algo em torno de 25% a mais de horas na etapa de análise de requisitos para se

chegar na implantação do software com exigências de retrabalho insignificantes. Como se sabe, a execução de retrabalhos ao final do projeto demandam um custo muito maior, por afetarem toda a estrutura do software desenvolvido. Normalmente, por conta deste alto impacto de custo, acabam-se procurando soluções que não irão atender completamente o usuário, mas que irão permitir um trabalho regular, com atividades extras e complementares na periferia do software. Isto constitui um consagrado fato gerador de insatisfações.

Os analistas de sistemas e desenvolvedores de software têm a tendência a formar uma compreensão parcial dos requisitos do projeto, “suficiente” para se iniciar o desenvolvimento. Normalmente, estando estes requisitos formalizados por escrito, se tem a falsa impressão de que foi construída a garantia necessária para se desenvolver o software.

O modelo proposto força uma interação entre analistas e usuários para se reavaliarem pressupostos falsos obtidos no início do projeto, mesmo que escritos e assinados pelos responsáveis.

A grande vantagem do modelo é exatamente esta: criar uma interação sistemática entre analista e usuário para se chegar a um resultado final satisfatório. A limitação de aplicação do modelo está no fato de exigir maior experiência por parte do analista para poder identificar e resolver os principais aspectos comunicacionais conflituosos. Esta experiência só se consegue com a vivência em desenvolvimento de software. Apesar disto, apenas pelo simples fato de se concentrar maior atenção às recomendações descritas no modelo, já se espera conseguir uma abordagem diferente ao desenvolvimento tradicional, que implicará em resultados mais satisfatórios.

### 3.3.1 Visão da Análise de Requisitos

Baseado no modelo discutido, definiu-se um diagrama, que mostra uma visão da análise de requisitos em que as fases do modelo se inter-relacionam de forma a buscar a criação de requisitos válidos e mais alinhados com a necessidade do usuário.

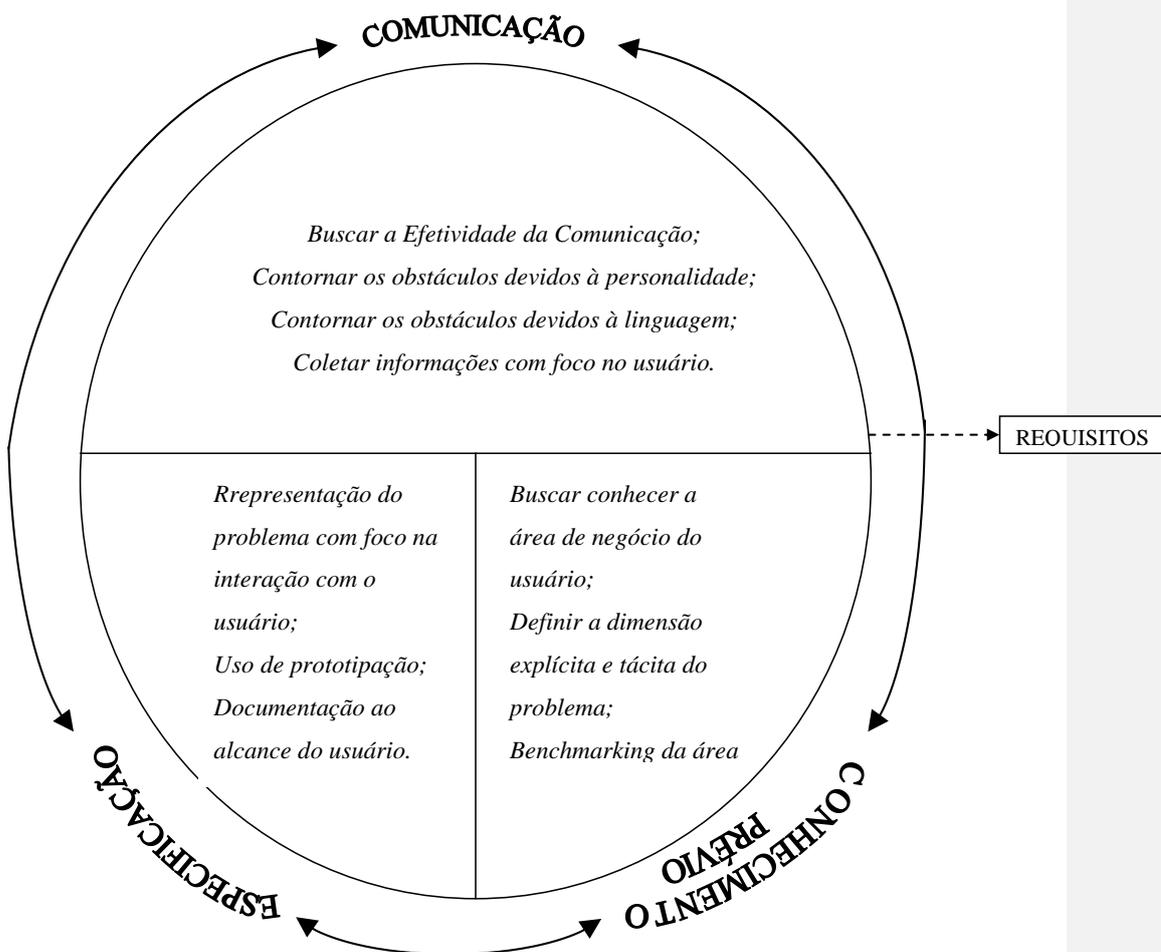


Figura 3.3.1 – Visão da Análise de Requisitos

## CONCLUSÃO

Na formulação inicial deste estudo, o objetivo principal era a discussão dos problemas relacionados a etapa de análise de requisitos e todos os transtornos que são gerados no desenvolvimento de um software quando não se dá a devida atenção ao equacionamento destes problemas. A luz de um conhecimento predominantemente empírico, buscou-se o respaldo de teorias e estudos da Engenharia de Software e da Ciência da Informação, com foco nas hipóteses levantadas deste trabalho. A revisão bibliográfica na temática sobre análise de requisitos permitiu descobrir uma variedade de autores e estudos, diretos e indiretos, sobre as dificuldades da análise de requisitos, que nos permitiram definir um marco teórico muito importante para o respaldo necessário à elaboração de um modelo voltado para ajudar na solução destas principais dificuldades. Face a este encaminhamento, foi possível conciliar o conhecimento prático e intuitivo de anos de experiência em desenvolvimento de software com teorias e estudos consagrados, que deram muito mais consistência ao resultado final.

Dos levantamentos efetuados, chamou a atenção o fato de poucas pesquisas e trabalhos no Brasil, estarem preocupados com esta temática, sendo que as fontes principais de referências encontradas, estarem concentradas na Europa e Estados Unidos. Alguns contatos que se fizeram no decorrer deste trabalho com pesquisadores brasileiros da UFRJ, UNICAMP e Universidade de Brasília que estavam trabalhando na área de concentração da Ciência da Computação com Engenharia de Requisitos, demonstraram a falta de interesse em abordar os aspectos comunicacionais ligados ao tema, sendo a ênfase principal destas pesquisas, os aspectos técnicos ligados ao software ou a usabilidade deste.

A complexidade e subjetividade dos aspectos comunicacionais na etapa de análise de requisitos exigem uma abordagem mais próxima das pessoas envolvidas no processo e a Ciência da Informação por sua característica multidisciplinar, fornece maior respaldo para este tipo de abordagem.

Como demonstrado no capítulo 3, os ganhos que se podem obter com um maior planejamento e estruturação da etapa de análise de requisitos, concentrando mais atenção nos aspectos comunicacionais, são muito significativos, fazendo a diferença sobre um software desenvolvido que atenda as necessidades do usuário ou não. Isto tem um impacto econômico muito importante neste tipo de projeto, desenvolvimento e manutenção de softwares, que implica numa forma diferente de se montar e gerenciar os cronogramas tradicionais de desenvolvimento, criando-se etapas novas na fase de análise de requisitos. É uma quebra de paradigmas, que se ocorrer, trará vantagens evidentes. Para cada hora a mais que se gasta na etapa de análise de requisitos, utilizando-se do modelo proposto, podemos economizar, em média, o dobro de horas de ajustes e retrabalho na fase final do projeto. Em projetos de longa duração isto tem uma implicação significativa.

O modelo proposto, no entanto não tem a pretensão de esgotar o assunto nem de apresentar uma solução definitiva a um problema clássico do desenvolvimento de software. Apenas começou-se a discutir, com uma abordagem mais ligada à Ciência da Informação, as principais dificuldades da etapa de análise de requisitos e a partir da mudança de foco, do software para o usuário, tentar buscar soluções mais apropriadas para estas dificuldades.

Ainda resta um campo vasto a ser explorado, principalmente na pesquisa das teorias psicológicas, com foco nos estudos dos aspectos cognitivos e estrutura da mente, da teoria da atividade, uma teoria que busca elucidar a formação dos processos mentais humanos a partir

do estudo da atividade, na Engenharia Semiótica, através do estudo dos fenômenos da Interação Humano Computador e do marco teórico discutido nos capítulos 1 e 2, que podem vir a ter um papel fundamental na resolução destes problemas.

## BIBLIOGRAFIA

ABBAS, Claudia Barenco. **Ontologias na Engenharia de Software**, Trabalho publicado na Internet, <http://qos.tecnolink.com.br/doc/Ontologia.htm>, Acesso em 10/07/2004

ACM SIGCHI (1992) **Curricula for human-computer interaction**. Technical report, ACM, NY, 1992. Disponível on-line em <http://www.acm.org/sigchi/>.

BARRETO, A. A. **A questão da informação**. São Paulo em Perspectiva, São Paulo, v. 8, n. 4, p. 3-28, 1994.

BATISTA, A. C. F. **Especulando a comunicação organizacional**. UNIVALE/ES. In: <[www.pucrs/famecos/geacor/texto10.html](http://www.pucrs/famecos/geacor/texto10.html)>. Acesso em: 19 nov. 2003.

CHERRY, Colin, **A comunicação Humana**. São Paulo: Editora Cultrix - USP, 1968

CHIZZOTTI, Antonio. **Pesquisa em Ciências Humanas e Sociais**. Cortez Editora, 4ª ed, 1987, p.89-106

CINTRA, Ana Maria et al. **Para entender as linguagens documentárias**. 2ª ed. ver. e ampl. São Paulo: Polis, 2002

CLARK, Robert G. e MOREIRA, Ana M. D., **Formal Specifications of User Requirements**. Netherlands, Automated Software Engineering n. 6, p. 217-232, 1999

FIDALGO, António **Semiótica Geral**. Covilhã: Universidade da Beira Interior, 1999.

HAYAKAWA, S. I., **A linguagem no pensamento e na ação**. São Paulo: Pioneira Editora, 1972

KARTEN, N. **Managing Expectations**, Dorset House Publishing, New York, NY, 1994

KOBASHI, Nair Yumiko. **A elaboração de Informações Documentárias: em busca de uma metodologia**. Tese apresentada a Escola de Comunicações e Artes da USP. São Paulo, 1994

KOBASHI, Nair Yumiko. **Análise documentária e representação da informação**. Rio de Janeiro: Informare – Cad. Prog. Pós Grad. Ci. Inf., v.2, n.2, p.5-27, jul/dez. 1996

LAUDON, Kenneth C. **Sistemas de Informações Gerenciais**. São Paulo: Prentice Hall, 2004

LE COADIC, Y.-F. **A ciência da informação**. Brasília: Briquet de Lemos, 1996. 119 p. Cap. 1: O objeto: a informação, p. 4-13.

MINOCHA, Shailey. **Requirements Development in User-Centred System Design**, London, artigo da Centre for HCI Design – City University, 2000

MINTZBERG, H. **An emerging strategy of direct research**. Administrative Science Quarterly, v. 24, Dec. 1989

MOLINA, J. Carlos Fernandez, **Enfoques objetivo y subjetivo Del concepto de informacion**, Revista Esp. Doc. Científica nº 17, vol. 3, Granada: Universidade de Granada, p. 320-329, 1994

NONAKA, Ikujiro e TAKEUCHI, Hirotaka. **Criação de conhecimento na empresa**. Rio de Janeiro: Campus, 1997

O'NEILL, Eamonn. **Cooperative Development: Under specification in External Representations and Software Usability**, London, Human Computer Interaction Laboratory – University of London, p.918-923, 1999

PENTEADO, José R. Whitaker. **A técnica da comunicação humana**. 9ª ed. São Paulo: Pioneira, 1986

PFLEEGER, Shari Lawrence. **Engenharia de software: teoria e prática**. 2ª ed. São Paulo: Prentice Hall, 2004

POLANYI, M. **The Tacit Dimension**, in: Laurence Prusak (ed), **Knowledge in Organizations**. Butterworth-Heinemann, Newton, MA, 1997

PRESSMAN, Roger S. **Engenharia de Software**. São Paulo: Makron Books, 1995

ROCHA, Ana R. C. ; MALDONADO, José C. ; WEBER, Kival C. **Qualidade de Software – Teoria e Prática**. São Paulo: Prentice Hall, 2001

SAIEDIAN, H. e DALE R., **Requirements engineering: making the connection between the software developer and customer**, USA, Information and Software Technology n. 42, p. 419-428, 2000

SATO, Yoshihiro, **Requirement Acquisition in System Development: A human-centred perspective of the Tacit Requirements**, London, AI & SOCIETY n. 9, p. 208-217, 1995

SELLTIZ, JAHODA, DEUTSCH, COOK. **Métodos de pesquisa nas relações sociais**. Editora da USP, 1ª reimpressão, 1971, p.225-261

SOUZA, Clarisse Sieckenius; LEITE, Jair Cavalcanti; PRATES, Raquel Oliveira; BARBOSA, Simone D.J, **Projeto de Interfaces de Usuário: Perspectivas Cognitivas e Semióticas**, Rio de Janeiro, PUC-RJ, 2000

VAKKARI, Pertti, **Task complexity, problem structure and information actions – Integrating studies on information seeking and retrieval**, Information Processing and Management n° 35, 1999