UNIVERSIDADE FEDERAL DE SANTA CATARINA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

CLAUDIOMIR SELNER

ANÁLISE DE REQUISITOS PARA SISTEMAS DE INFORMAÇÕES, UTILIZANDO AS FERRAMENTAS DA QUALIDADE E PROCESSOS DE SOFTWARE

Dissertação submetida à Universidade Federal de Santa Catarina para obtenção do grau de Mestre em Engenharia

FLORIANÓPOLIS
AGOSTO 1999

ANÁLISE DE REQUISITOS PARA SISTEMAS DE INFORMAÇÕES, UTILIZANDO AS FERRAMENTAS DA QUALIDADE E PROCESSOS DE SOFTWARE

CLAUDIOMIR SELNER

Esta dissertação foi julgada adequada para obtenção do Título de Mestre em Engenharia, especialidade em Engenharia de Produção e Sistemas e aprovada em sua forma final pelo Programa de Pós-graduação.

_____
Prof. Ricardo Miranda Barcia, Ph. D.
Coordenador do Curso de Pós-Graduação

Banca Examinadora:

_____
Prof. Paulo Maurício Selig, Dr. (Orientador)

_____
Prof. Roberto Carlos dos Santos Pacheco, Dr.

_____
Prof. Carlos Alberto Heuser, Dr.

Deus eterno e Pai querido, como lhe sou grato pelo seu imensurável amor, amor esse que me criou e me mantém seguro em comunhão com o seu Espírito, por causa de Jesus, o Messias prometido, o Cristo que tornou essa comunhão possível através de seu sacrifício na cruz. Amor esse, capaz de presentear-me com a abundância de todas as graças na forma de saúde, capacidade intelectual, alimentos e incontáveis outras coisas necessárias para que hajam condições de vida, desde o ar que respiro até a medida certa da luz do sol, para que não me queimem os olhos. Quantas palavras seriam necessárias para que meu louvor fosse completo? Perdoa-me, ó Deus eterno, porque o meu louvor é miserável, e sonda-me, e conheça os meus pensamentos e o tamanho da satisfação que me concede, e conhecerá o tamanho da intenção do meu louvor.

Ajuda-me, ó eterno, que em todas a situações eu lhe permaneça fiel e nessa fidelidade não me esqueça de repartir as bênçãos que me concede com aqueles a quem o Senhor ama, a saber, a humanidade sobre a face da terra e especialmente aqueles que a abundância do seu amor colocou em meu caminho: aqueles entes queridos que se compadecem de mim e que por mim clamam a ti nos momentos em que as dificuldades me sobrevém. Por eles eu vos suplico: abençoa-os ó Pai de amor, o Senhor os conhece e também às suas necessidades.

Às minhas queridas mulheres: Mara (minha esposa, minha companheira), Ester e Raquel (minhas filhinhas, minhas flores), mãe e oma Schnneider , motivos de sentimentos intensos.

Ao meu pai, que já foi chamado e está sob os cuidados do Deus eterno fora do mundo que conhecemos.

Ao Prof. Salvador Antônio dos Santos, amigo querido que no passado ajudou-me a estabelecer parte do fundamento sobre o qual essa dissertação foi construída.

AGRADECIMENTOS

Muitas entidades e pessoas serviram ao propósito dessa pós-graduação ser realizada e desse trabalho ser concluído. O maior risco, e medo, é o de que alguma delas possa não vir a ser referenciada, ainda assim, não há como deixar de dizer muito obrigado:

- à Universidade do Estado de Santa Catarina, que patrocinou e permitiu-me o acesso a esse curso,
- aos professores da Universidade Federal de Santa Catarina, que aceitaram o desafio de levarem a Joinville esse curso,
- à equipe da Kugel, que manteve essa empresa sob controle, enquanto os estudos eram desenvolvidos, e que serviu como laboratório para os experimentos realizados,
- ao meu primeiro orientador nesse trabalho, professor Plínio Stange, ainda que esse agradecimento humano não lhe seja mais útil, mas em sua memória, até mesmo para que fique registrado à posteridade a relevância de um trabalho bem iniciado,
- ao meu segundo orientador, professor Paulo Maurício Selig, que aceitou o desafio de "embarcar nesse bonde em movimento" (em virtude do falecimento do professor Plínio Stange) e ainda assim conseguiu tomar conta da sua direção,
- à minha querida esposa, Rosana Mara, pelo trabalho que teve na revisão do texto,
- aos membros da banca examinadora, pelas apreciações e considerações feitas, que contribuiram para o enriquecimento desse trabalho.

SUMÁRIO

LISTAS DE FIGURAS

RESUMO

O processo de análise de requisitos é o principal responsável pela falta de qualidade dos softwares de informações gerenciais disponíveis no mercado. A presente dissertação tem como objetivo demonstrar esse fato e apresentar uma metodologia que, levadas em conta algumas restrições, seja capaz de indicar uma possível solução para o problema.

Partindo-se da apresentação dos princípios teóricos que envolvem a matéria, são apresentadas algumas reflexões sobre a importância da análise dos requisitos, como base para a qualidade no desenvolvimento de softwares de informações gerenciais, e algumas estratégias que têm sido adotadas atualmente para resolver o problema em questão. São apresentados, também, conceitos relacionados à qualificação de softwares, bem como as estruturas de alguns padrões de qualidade, como a ISO 9000-3, a ISO/IEC 9126 (NBR13.596) e o modelo SEI-CMM.

Por fim, é apresentada e demonstrada a nova proposta, que resume-se ao aproveitamento dos princípios de algumas ferramentas para o planejamento da qualidade já existentes (como o diagrama de causa e efeito, de Ishikawa e o QFD, de Akao), e a alguns novos dispositivos, desenvolvidos com o fim de apoiar os processos de análise propriamente (planilhas de “respostas desejadas” e metodologia) e suportar a documentação desses processos (“gabarito de processos”).

A orientação específica das questões dessa proposta é para a característica “funcionalidade — adequação”, de acordo com a norma ISO/IEC 9126.

PALAVRAS-CHAVES:

- Qualidade de Software
- Processos de Software

- Análise de Requisitos de Software
- Ferramentas para a qualidade
- ISO 9126, ISO 9000-3, SEI-CMM

ABSTRACT

The requirements analisys process is the main responsible for the quality lack in management information softwares, available in the market. This dissertation has as objective demonstrates that fact and to present a methodology that, considering some restrictions, be capable to indicate a possible solution for the problem.

Departing from the theoretical beginnings presentation that they involve the matter, are presented some reflections on the importance of the requirement analisys, as base for the quality in the development of managerial information softwares, and some strategies that have been adopted nowadays to solve the problem in subject. They are presented, also, concepts related to softwares qualification, as well as the structures of some quality patterns, like ISO 9000-3, ISO/IEC 9126 and the SEI-CMM model.

Finally, it is presented and demonstrated the new proposal, that is summarized to the use of the rudiments of some quality planning tools already existent (as Ishikawa's cause and effect diagram and Akao's QFD), and some new devices, developed in order to support the analisys process properly ("desired answer" spreadsheets and methodology) and to support the documentation of those processes ("processes templates")

The specific orientation of that proposal subjects is for the characteristic "Funcionality — Adequacy", in agreement with the norm ISO/IEC 9126.

KEY-WORDS

- Software Quality
- Software Process
- Software Requirements Analisys
- Quality Tools
- ISO 9126, ISO 9000-3, SEI-CMM

INTRODUÇÃO

I — CONSIDERAÇÕES GERAIS

Já há, pelo menos, duas décadas vêm-se anunciando em cursos, palestras, livros e revistas, os grandes problemas relacionados à qualidade de software. Uma estatística (v. ***fig.I.a***) do ano de 1989, apresentada em [Wilson,1998], indicava que, de todos os projetos de software iniciados nos Estados Unidos, 29% nunca foram entregues, 47% foram entregues mas nunca usados e 19% foram entregues mas intensivamente alterados antes de serem utilizados. Esses problemas foram encontrados em diversos tipos de software, tais como sistemas operacionais, softwares de controles de dispositivos de automóveis e aeronaves e softwares de apoio aos sistemas de informações gerenciais (de planejamento de materiais, de controle de compras, de administração pública).

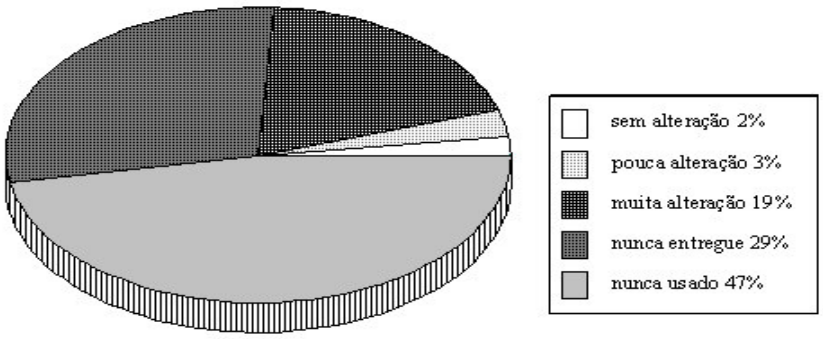


fig.I.a — percentual de aproveitamento dos projetos de software

Quase uma década depois, em 1996, Capers Jones (apud [Magnani,1998]) divulga uma nova estatística (v. ***fig.I.b***) demonstrando que os problemas permanecem. Nessa estatística, 65% dos grandes projetos (projetos com aproximadamente 100.000 pontos de função) haviam sido cancelados.

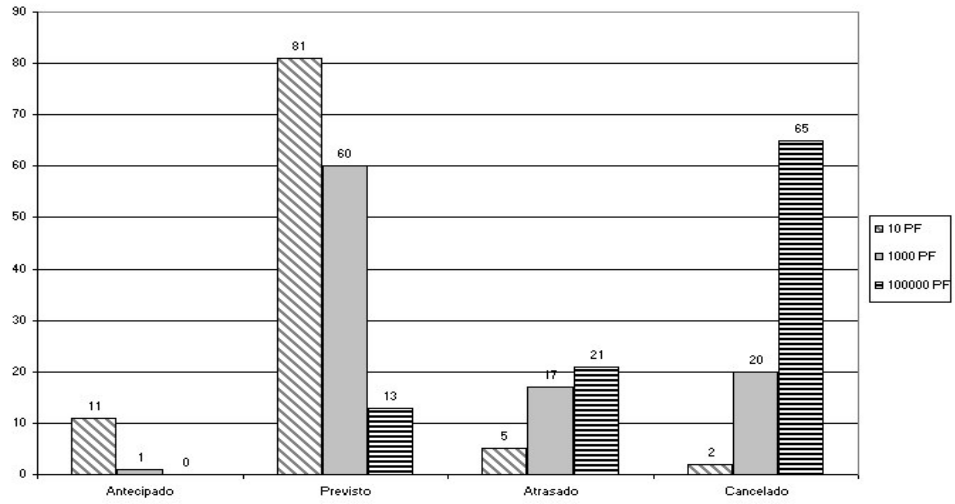


fig.I.b — percentual de aproveitamento dos projetos (por tamanho em pontos de função)

Assim, há pesquisas que evidenciam a existência de problemas de qualidade relacionados ao desenvolvimento de softwares e as evidências parecem indicar que esses problemas têm uma de suas principais fontes de origem na fase inicial do ciclo de vida de um software, que compreende o processo de análise dos requisitos. O presente trabalho visa demonstrar a coerência dessas evidências e pretende apresentar um modelo de processo para a análise dos requisitos de software que pode contribuir para a redução desses problemas.

II — POR QUE VALE A PENA RESOLVER O PROBLEMA?

"***Software é um problema?***" — Com essa pergunta foi dado início ao segundo tutorial (Melhoria de Processos de Software) da Terceira Semana de Engenharia de Software, ocorrida em agosto de 1998 em São Paulo, capital. O sentido da pergunta não era para saber se software "tem" problemas, mas se software "é" um problema. Observe-se:

a) ciclo de vida de desenvolvimento de software pode exceder ao ciclo de vida de desenvolvimento do equipamento que será controlado por esse software (automóvel, micro-computador, aeronave, etc.);

b) softwares têm-se tornado elementos críticos em muitos produtos;

c) conforme [Magnani,1998], o mercado de software deverá atingir, entre os anos 1995 e 2000, crescimentos anuais de 20% na Ásia, 16% na América Latina e 15% nas demais regiões do mundo.

Talvez a expressão “problema” não seja a mais adequada, mas, indubitavelmente, software é um grande desafio a ser enfrentado, dadas as possibilidades e criticidade no seu uso.

Quanto aos problemas “no” software, eles podem levar a conseqüências além daquelas relacionadas ao custo de correção desses problemas, envolvendo, por exemplo, processos judiciais, vidas humanas que se perdem, material desperdiçado, danos ao meio ambiente, o tempo de parada dos usuários, horas extras para recuperar o serviço atrasado, lucros cessantes por decisões que precisariam ser tomadas sobre os dados que deixaram de estar disponíveis, multas contratuais, etc.. Resta saber o que está sendo controlado pelo software (uma empresa? Um avião? Um automóvel? Uma usina nuclear?). Naturalmente, nem todos os problemas de software são introduzidos na fase de análise de requisitos. No entanto, aqueles que são causados por falhas na fase de produção parecem ser mais simples de serem corrigidos do que os causados na fase de análise dos requisitos, uma vez que, basicamente, envolvem somente a recodificação das partes defeituosas dos programas (analisando-se estritamente os erros *no* software, sem considerar-se as conseqüências do erro). Já os problemas de requisitos tendem a afetar a estrutura do software.

Dadas as estatísticas, as possibilidades de aplicações, as tendências de mercado e as conseqüências de problemas com software, é razoável supor-se que um projeto que busque a qualificação dos requisitos de software é relevante, tanto do ponto de vista econômico quanto do social.

III — QUAIS OS RESULTADOS ESPERADOS DO MODELO A SER PROPOSTO?

Antes da apresentação dos objetivos desse trabalho, é necessária a localização sobre para qual espécie de sistemas objetiva-se desenvolver um modelo de

análise de requisitos.

Em 1982, Tom DeMarco, no seu livro “Controlling Software Projects” escreveu o seguinte:

> ... três perspectivas são tudo o que precisamos para especificar a maioria dos sistemas. Analisando os requisitos a partir dessas três perspectivas, teremos três componentes de modelo: um modelo de função (perspectiva do que o sistema faz), um modelo de dados retidos (perspectiva do que o sistema memoriza) e um modelo de transição de estado (perspectiva de diferentes estados comportamentais que caracterizam o sistema). Os três juntos constituem a composição do modelo de requisito. *[DeMarco,1982, p.67].*

O presente trabalho assume que o conceito das três perspectivas apresentado por DeMarco é plausível, entretanto apresentará argumentos que demonstram a excentricidade dessas perspectivas. Em outras palavras, os sistemas não são um equilíbrio das três perspectivas apresentadas, mas eles tendem a, normalmente, uma de duas perspectivas em particular, variando conforme sejam sistemas de informações gerenciais (como um sistema de administração dos estoques) ou de controle de objetos (como um software embarcado num automóvel ou um de usinagem CNC). O aspecto da excentricidade dessas perspectivas pode não ter grande importância nas atividades que compõem os processos de projeto do software, desenvolvimento das ferramentas, produção do software, etc., mas é fundamental a sua percepção nas atividades que compreendem o processo de análise dos requisitos.

A partir do ponto de vista dessas perspectivas e baseado na tendência observada por Peter Drucker, para quem a próxima revolução da informação terá como objetivo a identificação do significado e do propósito da informação [Drucker,1998], serão mérito de aplicação, no presente trabalho, somente os sistemas de informações gerenciais. Esses sistemas têm como objetivo responder a determinadas perguntas formuladas pelos seus usuários (como, por exemplo, “qual foi o faturamento no mês passado?” ou “qual a disponibilidade em estoque do parafuso 1/4 x 7/8 ?”). Dentro desse contexto, os objetivos do presente trabalho estão relacionados às respostas (que devem possuir um significado e um propósito) a essas perguntas formuladas pelos usuários.

Existem, essencialmente, três classes de respostas que são esperadas dos

sistemas de informações gerenciais:

a) as respostas sobre o conteúdo de certas variáveis, que permitem a manutenção do controle sobre a empresa (por exemplo, o que deveria ter sido pago pelos clientes no dia de ontem, mas que não foi pago?);

b) as respostas que apoiam os processos decisórios na empresa (por exemplo, entre os diversos fornecedores potenciais do parafuso 1/4 x 7/8, quais, tradicionalmente, entregam rigorosamente dentro do prazo combinado?) e

c) as informações solicitadas por entidades de controle, externas à empresa (por exemplo, entidades governamentais e empresas com matriz no exterior).

A partir desses aspectos, fica estabelecido como objetivo geral desse trabalho: contribuir para a redução dos problemas causados pela indisponibilidade de informações que atendam às necessidades de negócio de uma empresa, buscando-se um aumento no índice dos projetos de softwares que são concluídos, entregues e utilizados com poucas ou nenhuma alteração, melhorando os resultados das pesquisas apresentadas acima, e como objetivos específicos: permitir, em tempo de análise de requisitos, a identificação e o teste das informações desejadas pelos usuários de um determinado sistema de informações gerenciais em estudo, de tal forma que essas informações, após a produção e a implantação do software, tenham um significado claro e um propósito de utilidade aos interesses da organização onde esse sistema de informações estiver sendo inserido.

IV — ESTRUTURA DO TRABALHO

O objetivo dessa seção é a de fornecer uma visão da estrutura do trabalho. Sendo assim, num primeiro momento serão apresentados alguns conceitos que são parte do fundamento sobre o qual será construída a argumentação adiante e, por ser imprescindível que consiga-se perceber a estrutura assíncrona das idéias a serem transmitidas, representou-se na ***fig.IV.a*** essa estrutura (uma vez que o modelo a ser adotado para a apresentação desses conceitos é, naturalmente, o modelo verbal escrito,

que é totalmente linear).

Num segundo momento será apresentado o problema para o qual deseja-se desenvolver a proposta de solução, alguns métodos que vêm sendo adotados para tentar solucionar esse problema atualmente e, em seguida, serão apresentados os dispositivos a serem utilizados nesse trabalho, a saber, as planilhas de respostas desejadas, o diagrama de causa e efeito de Ishikawa, o diagrama de Pareto, a matriz baseada no QFD e as fichas de processos.

Por último será feita a descrição de algumas aplicações dos dispositivos. Serão apresentadas, também, as conclusões a que pôde-se chegar a partir dessas aplicações e alguns indícios para continuidade do trabalho.

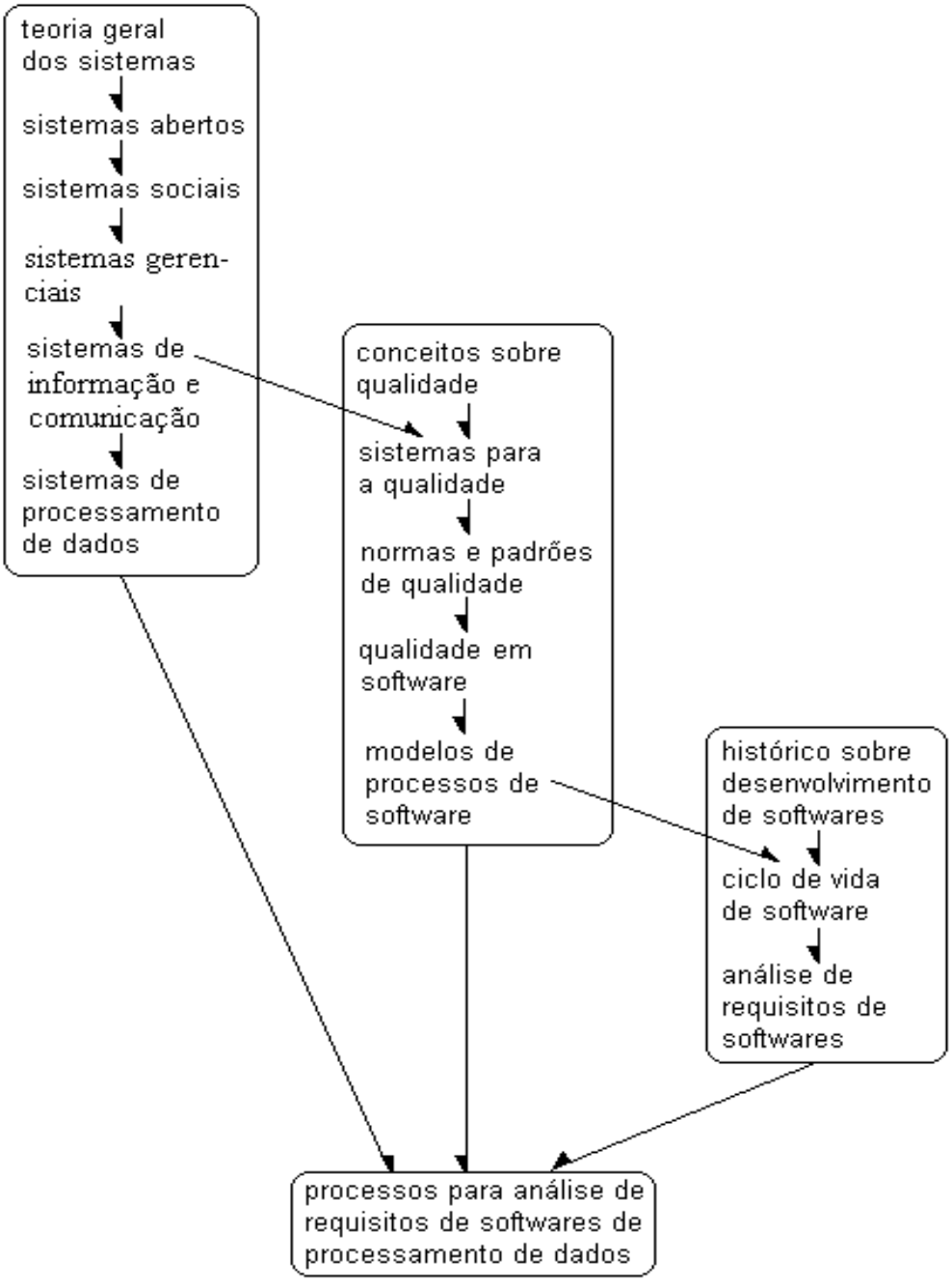


fig.IV.a— divisão dos conceitos do trabalho

CAPÍTULO 1 — CONCEITOS SOBRE SISTEMAS

Existem vários conceitos diferentes sobre sistemas, variando conforme o tipo de sistema e conforme o enfoque desejado pelo seu autor. Esses conceitos deixam transparecer que, aparentemente, praticamente tudo o que encontra-se à nossa volta é sistema. Os conceitos abaixo foram extraídos do “Webster´s New Collegiate Dictionary”, apud [Yourdon,1990,p.13] e estendidos para o presente trabalho:

a) um grupo de corpos que interagem entre si sob a influência de forças relacionadas (gravitacional);

b) uma mistura de substâncias que tende para o equilíbrio (termodinâmico);

c) um grupo de órgãos do corpo que desempenham, em conjunto, uma ou mais funções vitais (digestivo);

d) um grupo de dispositivos ou uma organização em rede, que relacionam-se entre si principalmente para a distribuição de algum produto ou servindo a um propósito comum (telefônico, rodoviário, processamento de dados)

e) um conjunto organizado de doutrinas, idéias ou princípios, habitualmente previsto para explicar a organização ou funcionamento de um conjunto sistemático (mecânica newtoniana);

f) um procedimento organizado ou estabelecido (toques da digitação);

g) uma maneira de classificar, simbolizar ou esquematizar (sistema taxonômico, sistema numérico);

h) organização harmoniosa ou modelo;

i) sociedade organizada ou situação social vista como indesejável.

Todos esses conceitos aplicam-se a sistemas conforme o tipo do sistema a ser analisado e, como pode ser visto nas alíneas acima, existem muitas propostas de classificação dos sistemas. Para os fins desse trabalho, entretanto, os sistemas serão subdivididos em apenas dois grandes grupos: o grupo dos sistemas fechados e o dos

sistemas abertos.

Os sistemas fechados são caracterizados pela ausência da capacidade de evoluírem com o passar do tempo, por estarem sujeitos às leis da física tradicional, particularmente à segunda lei da termodinâmica. Cada um deles é um conjunto particular de relacionamentos entre elementos individuais (grãos de areia, cabos elétricos, elementos químicos, rolamentos, etc.), atraídos, agrupados e mantidos coesos pela imposição de determinadas forças (energia nuclear, gravitacional, trabalho humano, etc.), podendo ter sua origem a partir de objetivos declarados, como a usinagem de peças de aço (para o caso de um sistema "torno mecânico") ou a eliminação de determinados fatores como ruído, calor ou luz (para o caso de um sistema "sala de reuniões" ou um "par de óculos escuros") ou a partir de objetivos não declarados, como o sistema solar.

Os estudos sobre os sistemas abertos começaram a ser formalizados no início do século 20 e esse grupo de sistemas será explorado mais profundamente, uma vez que diz respeito aos sistemas sociais e aos sistemas de informações, inseridos nesses sistemas sociais. As características dos sistemas abertos começaram a ser melhor compreendidas a partir da teoria geral dos sistemas.

1.1 — A TEORIA GERAL DOS SISTEMAS

A teoria geral dos sistemas foi proposta na década de 40 por Ludwig Von Bertalanffy, entretanto as sementes dessa teoria já eram percebidas em 1920, na interpretação de Oswald Spengler sobre "cultura". Para Spengler, "cultura" era um sistema integrado em que qualquer idéia e formas artísticas ***interagiam*** umas com as outras e com as instituições sociais, para simular um determinado estilo comum e, portanto, os estudos para entender um determinado aspecto cultural deveriam ser sobre esse processo de interação, preferencialmente ao estudo sobre cada idéia, expressão artística ou instituição social isoladamente.

Uma evidência da dificuldade que representa a compreensão de um sistema, aparte do relacionamento entre os seus elementos, é o dilema de Blase Pascal (apud [Duailibi,1995]) *"só posso compreender o <u>todo</u> se conheço as partes, mas só posso*

compreender as partes se conheço o todo", para a hipótese do desconhecimento de ambas as variáveis interdependentes, o *todo* e as *partes.* A partir do estudo da interação entre as partes, no entanto, esse dilema pode ter sua releitura como *"posso compreender o todo e as partes a partir do conhecimento da* ***interação*** *dessas partes entre si, pelas funções desempenhadas por cada uma delas nessa* ***interação****"*, reduzindo o dilema a um problema com uma única variável: a interação.

Bertalanffy, biólogo húngaro educado em Viena, tentou reviver a unidade da ciência, propondo um modelo geral de "sistema aberto". Uma grande contribuição de sua obra está na descrição da contradição entre a termodinâmica dos organismos vivos e a segunda lei da termodinâmica, dando origem ao conceito de entropia negativa.

Para compreendermos melhor o conceito de entropia negativa, é interessante que tenha-se uma noção da segunda lei da termodinâmica e dessa função de estado introduzida por Glausius, por ele denominada de entropia (do grego "transformação"). Trata-se de uma função que identifica a variação entre o estado (***i***)nicial e o estado (***f***)inal de um sistema ($\varsigma_i^f = S_f - S_i$), depois de passar por um processo de transformação ou passado um certo período de tempo Δt [Nussenzweig,1981]. Uma das bases dessa função é o fenômeno da desagregação ou de retorno dos elementos de um sistema ao seu estado original, anterior ao de serem reunidos como um sistema, na medida em que sobre esses elementos deixar de ser aplicada a energia (ou pressão) que os mantêm juntos. Segundo [Katz,1974,p.34] "*De acordo com esta última lei, ..."* (referindo-se à segunda lei da termodinâmica) "... *um sistema tende ao equilíbrio; tende a esgotar-se, isto é, a tendência das estruturas diferenciadas é moverem-se para a dissolução, à medida que os elementos que as compõem se acomodam em desordem aleatória*" e isso ocorre por causa da degradação da energia (porque os sistemas, ao passarem de um estado para outro, necessariamente consomem energia) e o comportamento do sistema, durante esse processo, seria totalmente previsível a partir do estudo das propriedades dos seus elementos. O que ocorre entretanto, segundo Bertalanffy (apud [Heylighen,1992]), é que um sistema aberto que importa energia livre de fora, pode legitimamente prosseguir através de estados de incremento de heterogeneidade e ordem, ou seja, os sistemas abertos ***interagem*** com seus ambientes, levando a comportamentos imprevisíveis e podendo adquirir propriedades qualitativamente novas,

não estando, por um lado, sujeitos à segunda lei da termodinâmica, tendo por resultado a evolução contínua (ou entropia negativa, na medida que o estado final do sistema é mais elaborado que o inicial) e por outro lado dificultando a sua análise. Conforme Bertalanffy (apud [Katz,1974]), nós temos que compreender sistemas vivos como sendo sistemas de elementos em interação dinâmica mútua e descobrir as leis que governam os padrões de seus componentes e processos.

1.2 — OS SISTEMAS ABERTOS

Um sistema aberto é um conjunto particular de relacionamentos entre elementos individuais (pessoas, máquinas, instalações, células, órgãos, etc.), atraídos (ou deslocados), agrupados e mantidos coesos pela imposição de determinadas forças (pressão atmosférica, pressões de mercado, pressões sociais, trabalho humano, etc.), extraídas de determinadas fontes de energia (núcleo do átomo, gravidade, madeira, gasolina, motivos éticos, dinheiro, alimentos, etc.) de origem externa, tendo como objetivo o processamento e a transformação de determinados insumos (grãos de areia, cimento, aço, madeira, etc.) em artigos (essa expressão será adotada para representar qualquer espécie de resultado de um sistema ou processo, como, por exemplo: produtos, serviços, oxigênio, humos, etc.) de interesse do sistema maior onde encontra-se inserido. É imprescindível a compreensão do fato de que os sistemas abertos caracterizam-se pela relação que têm com outros sistemas e pela dependência que mantêm com esses sistemas, no que tange à importação e exportação de energia. Segundo [Katz,1974,p.33]:

> *As formulações mais antigas de esquemas de sistema lidavam com os sistemas fechados das ciências físicas, nos quais as estruturas relativamente autocontidas podiam ser tratadas com êxito, como se fossem independentes de forças externas. Mas os sistemas vivos, quer sejam organismos biológicos, quer sejam organizações sociais, se acham agudamente na dependência de seu meio externo e, por isso, precisam ser concebidos como sistemas abertos.*

Desse conceito podem-se derivar algumas observações importantes:

a) um sistema aberto não existe a partir de si próprio. Ele precisa ser alimentado com insumos e fontes de energia, oriundos do sistema maior,

no qual está inserido;

b) um sistema aberto não existe em função de si próprio. O interesse num sistema aberto está em que ele cumpra o seu papel, fornecendo os artigos que dele esperam os sistemas que estão a sua volta;

c) as declarações das alíneas “a” e “b” trazem consigo uma outra característica importante: a relação entre o cumprimento do papel de um sistema aberto e a sua própria continuidade. Na medida em que ele deixar de cumprir o seu papel, pelo não fornecimento das saídas esperadas, também os sistemas que estão à sua volta procurarão deixar de alimentá-lo por considerá-lo um parasita (um relacionamento não simbiótico, no caso dos sistemas biológicos e um relacionamento com mais sansões do que recompensas, no caso das organizações sociais), o que pode levar a uma relação de causa e efeito que culmina com o seu desaparecimento: o sistema deixa de cumprir o seu papel forçando o meio a buscar novas alternativas e na medida em que esse sistema não é mais requerido também não lhe são mais liberadas as fontes de energia nem os insumos. Naturalmente, a energia acaba na medida em que faltarem as suas fontes e, uma vez que essa energia era necessária para manter os elementos desse sistema coesos, ele desaparece (os seus elementos retornam ao estado anterior ao de serem reunidos como um sistema, entrando em equilíbrio com o meio do qual foram extraídos). Conforme [Senge,1990], essa é a principal característica dos sistemas: a sua natureza cíclica, quer dizer, os processos de entrada, transformação e saída de energia de um subsistema completam-se em círculos com os processos de entrada, transformação e saída de energia de outros subsistemas;

d) se os elementos que vão formar um sistema são atraídos a partir da imposição de determinadas forças, fatalmente eles entrarão em choque entre si [Poulin,1998] e, até que consigam acomodar-se (ou adaptar-se), os atritos oriundos desses choques podem ser fontes de problemas.

1.3 — OS SISTEMAS SOCIAIS

Bertalanffy observou que o conceito de estruturação (conexão, ligação) dos elementos, independente da substância concreta dos seus elementos (por exemplo partículas, transistores ou povos), suportaria disciplinas diferentes (física, biologia, sociologia, etc.), fornecendo uma base para a unificação dessas disciplinas, além de levar a uma nova compreensão de um sistema como sendo algo mais do que simplesmente a soma das suas partes componentes. Por isso, o foco da teoria geral dos sistemas está no arranjo que conecta os componentes em um todo e não nos atributos constantes desses componentes.

Uma concepção ainda mais ampla para a conexão de componentes de um sistema, que complementa o raciocínio de Bertalanffy, é a concepção de *estrutura dinâmica* que conecta os componentes de um sistema. Essa concepção é aplicável sobre uma classe especial dos sistemas abertos, composta dos sistemas sociais (famílias, empresas, universidades, etc.). Segundo [Katz,1974]:

> O sistema social é a estruturação de **eventos** ou acontecimentos e não de partes físicas e, por conseguinte, não tem estrutura aparte de seu funcionamento (Alport, 1962). Sistemas físicos ou biológicos, tais como automóveis ou organismos, têm estruturas anatômicas que podem ser identificadas, mesmo que não em funcionamento *... ([Katz,1974,p.47]) e ... s*ão os **eventos**, mais do que as coisas, que se acham estruturados, de modo que a estrutura social é conceito mais dinâmico do que estático *[Katz,1974,p.36] (grifos acrescentados).*

Existe um propósito pelo qual a expressão “eventos” foi colocada em destaque, acima: a visão ampliada dos relacionamentos como sendo ocasionados em eventos internos ou externos aos quais o sistema está sujeito, aliada à constatação de que os atritos entre os elementos que chocam-se nesses eventos podem ser uma fonte de problemas, permite a percepção de que um sistema social específico como uma empresa, deverá ter a capacidade de minimizar esses atritos, primeiramente nos eventos externos, de onde provém as fontes de energia, sob pena de ser rejeitado pelo ambiente onde está inserido, e em segundo lugar nos eventos internos, para economizar energia, sob pena de desaparecer por consumir mais energia do que sua capacidade de importá-la. Conforme [Katz,1974,p.37] “*Os sistemas sociais, porém, não*

se acham vinculados às mesmas constâncias físicas dos organismos biológicos e, por isso, podem ser capazes de deter quase que indefinidamente o processo entrópico. Apesar disso, o número de organizações que deixa de existir todos os anos é grande".

Sendo assim, é coerente dizer-se que a medida da vitalidade de um sistema social é a razão entre a **energia obtida** nos eventos externos em que ele toma parte e o somatório de toda a **energia necessária** para:

a) promover e participar desses eventos externos e

b) atrair e manter coesos nos eventos internos, os elementos que serão estruturados nesse sistema.

1.4 — AS ORGANIZAÇÕES SOCIAIS

Um tipo específico de subsistemas sociais são as organizações sociais. Conforme [Katz,1974], elas podem ser subdivididas em:

a) subsistema técnico ou de produção: responsável pela transformação da energia, e abrange as principais atividades ligadas ao *negócio* principal da organização;

b) subsistema de apoio: responsável pela manutenção do ciclo contínuo de entrada de energia e insumos e de saída de artigos (insumos e energia transformados);

c) subsistema de manutenção: responsável pela entrada de componentes mais "permanentes" ao sistema, tais como equipamentos e pessoal. Também respondem pela eficiência desses componentes, através de atividades tais como lubrificação e conserto de equipamentos e treinamento de mão-de-obra;

d) subsistemas adaptáveis: responsáveis pelas funções de procura e disposição, que permitirão a percepção de mudanças no meio e a adaptação constante da organização a esse meio e

e) subsistemas gerenciais: responsáveis pelo controle, coordenação e direção

dos demais subsistemas da organização.

Especificamente para os fins desse trabalho, os subsistemas mais importantes de serem compreendidos, das organizações sociais, são os subsistemas gerenciais, uma vez que sob a sua responsabilidade serão feitas as coletas de informações sobre os eventos tanto internos quanto externos, objetivando o controle dessa organização social.

Ainda nessa seção, é imprescindível que fique bem evidente a diferença entre função e organização: a organização pressupõe apenas que um determinado sistema organizacional seja subdivisível em órgãos. Já as funções são resultantes do desempenho dos órgãos atuando em conjunto. Essas funções poderão ser percebidas somente no **relacionamento** desses órgãos entre si. Sem estudar-se o relacionamento entre os órgãos, as funções não podem ser percebidas.

1.5 — OS SUBSISTEMAS GERENCIAIS

A minimização dos atritos gerados pelo choque entre os elementos que deslocam-se para compor um sistema social, atraídos a partir da imposição de determinadas energias, é da responsabilidade do subsistema de gerenciamento dessa organização. O risco desse sistema social vir a desaparecer é função da reversão dessa capacidade de atrair os elementos, muitas vezes desencadeada sem que o subsistema gerencial perceba a tendência apresentada pelo conjunto histórico dos conteúdos das variáveis que representam esses atritos. Para que o conteúdo dessas variáveis sejam mantidos sob controle, é necessário que identifique-se:

a) as variáveis mais representativas de cada evento (unidades de medida dos indicadores);

b) os valores dos conteúdos, admitidos para cada uma das variáveis (quantidades de unidades);

c) que deverá ser feito para que esses valores sejam respeitados (atividades do processo);

d) os pontos de coleta dos conteúdos dessas variáveis após a execução das

atividades dos processos;

e) os mecanismos que permitam a percepção de eventuais tendências e

f) a forma de atuação, caso alguma tendência seja observada.

À identificação do conjunto de parâmetros das alíneas “a” até “d”, acima, dá-se o nome de planejamento, no qual são estabelecidas as metas a serem atingidas e os caminhos a serem percorridos, para que essas metas sejam alcançadas. Por outro lado, após a execução de um determinado plano, um processo de retroinformações é necessário para que o conteúdo das variáveis que representam os eventos sejam mantidos sob controle, segundo os patamares planejados para essas variáveis, prevendo eventuais tendências que um conjunto histórico desses conteúdos possa representar. Conforme [Katz,1974,p.61], “*O uso sistemático de informação para orientar o funcionamento organizacional é o sine qua non de uma organização*”. Assim, entre os conceitos mais importantes a serem estabelecidos sobre os subsistemas gerenciais, estão os conceitos de planejamento e de controle. Conceitualmente, o gerenciamento estaria mais relacionado à capacidade de atingir um conjunto de metas (orientado para o controle, nesse caso) e a gestão estaria mais relacionada à capacidade de mudança (ou seja, é orientada para o planejamento). Para os efeitos desse trabalho, entretanto, a terminologia para referir-se tanto ao planejamento quanto ao controle será “gerenciamento”.

Para uma melhor compreensão da importância e do significado dos conceitos de planejamento e controle, é conveniente uma reflexão sobre a relação entre eles. Assim, uma vez que por controle entenda-se a *comparação entre os resultados esperados com os efetivos* de um determinado processo, seguida da tomada de ação para a correção de eventuais desvios, o princípio resume-se a que se não houver um planejamento (de onde obtém-se os resultados esperados) também não será possível o controle. Na ausência de um planejamento o que pode-se ter seria algo como um

*acompanhamento***.** O planejamento possibilita o controle e o controle justifica o planejamento. Nesses termos, não faz sentido falar-se de controle sem planejamento, assim como também não faz sentido falar-se de planejamento sem que haja a intenção do exercício do controle, para que sejam atingidas, de fato, as expectativas, ou metas, estabelecidas como base desse planejamento.

A ***fig.1.5.a*** representa o fluxo de informações de um processo para o controle de uma organização social. A essência desse fluxo está no fato de que tanto o planejamento quanto o controle não têm a intenção original de *fazer* alguma coisa, mas sim de permitir que se *saiba* alguma coisa sobre a execução de algum processo dos subsistemas técnico, de apoio, de manutenção e adaptáveis. Esse "saber" pode tanto ser de fatos ainda previstos quanto de fatos já passados, mas, em ambos os casos, nunca do presente, ainda que o lapso de tempo entre o plano e a execução, ou entre a execução e o controle, seja apenas de uma fração de segundos.

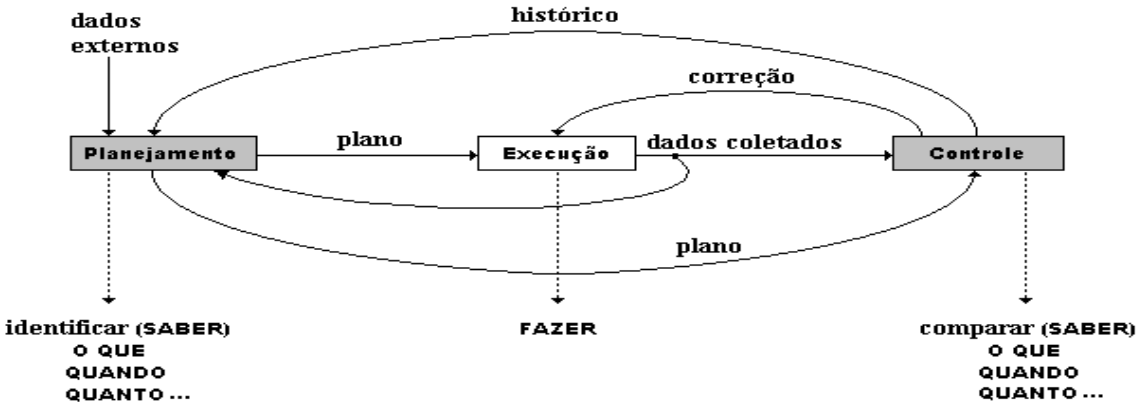


fig.1.5.a — Fluxo de um sistema de informações

Outro aspecto importante, identificável no diagrama da ***fig.1.5.a***, está relacionado aos dados coletados da execução. Observe-se que ao mesmo tempo em que eles servem para serem comparados a um plano feito no passado (portanto para controle) servem também como um novo planejamento de fatos futuros. É o caso de registros de eventos que foram planejados no passado (por exemplo, o registro do evento "entrada de determinados materiais" foi planejado na "compra" desses mesmos materiais, no passado), servindo esses registros de eventos, por sua vez, como plano de eventos que seguramente vão ocorrer no futuro (o registro do evento "entrada de

materiais" funciona como um planejamento do evento futuro "pagamento dos materiais comprados").

1.6 — SISTEMAS DE INFORMAÇÕES

O que é "informar"?" — "Informar", segundo [GDBCP,1979] é "*dar informe, parecer, notícia. Instruir, ensinar, comunicar* ...".

Como pode ser observado, os termos "informar" e "comunicar" podem assumir o mesmo papel numa determinada expressão, ou seja, são sinônimos. O objetivo dessa constatação é menos a discussão sobre até que ponto essas palavras podem ou não ser consideradas sinônimas, mas é interessante observar-se que, independente do estado infinitivo desses verbos, a "comunicação" e a "informação", enquanto substantivos, assumem papéis distintos. Para estabelecer-se uma noção da relação entre um sistema de comunicações e as informações, no contexto das organizações sociais, hajam vistas ao conceito de [Hampton,1983,p.358]: "*para fins de administração, a comunicação é o processo pelo qual as pessoas que trabalham em uma empresa transmitem informações entre si e interpretam o seu significado"*. Dessa forma, "transmitir informação" pode ser o objetivo de um sistema de comunicação (esse conceito é apenas indicativo. Algumas reservas serão reconhecidas mais adiante, conforme [Warnier,1984]).

Um sistema de comunicação segue, genericamente, o esquema da ***fig.1.6.a*** e, segundo [Hampton,1983], obedece um conjunto de etapas: o emissor detém um **significado** a ser transmitido, esse significado é **codificado** para um determinado modelo (detalhes sobre modelos serão analisados adiante, nesse trabalho), esse significado codificado recebe o nome de **mensagem** que é endereçada e transmitida por um determinado **canal** (telefone, documento, etc.) a um receptor. Esse receptor **decodifica** a mensagem e procura **compreendê-la**. Obedecendo um processo semelhante, porém inverso, o receptor assume o papel de emissor e transmite ao emissor original (agora receptor) um sinal de que foi ou não estabelecida a comunicação. O sinal de retorno é chamado de **retroinformação**.

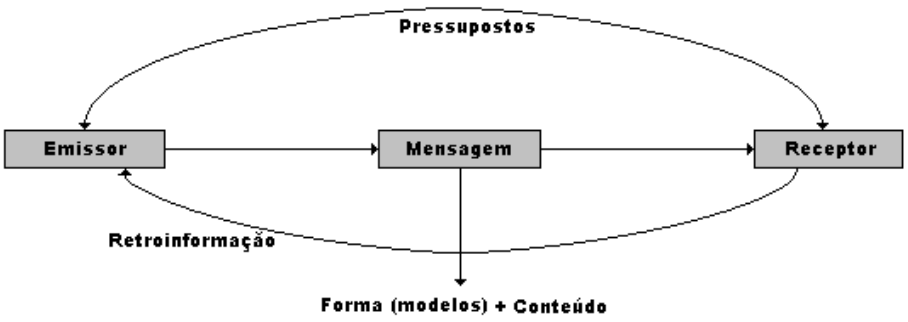


fig.1.6.a — Modelo de um sistema de comunicação

As principais considerações a serem tecidas sobre o esquema apresentado na ***fig.1.6.a*** estão relacionadas às formas e conteúdos da mensagem. A formalidade (ou rigor da forma) na comunicação depende basicamente de dois fatores: do tempo em que a mensagem poderá levar entre a sua emissão e recepção e da medida dos pressupostos comuns entre o emissor e o receptor (um fator que influencia essa medida é o nível de intimidade entre o emissor e o receptor da mensagem). Numa organização social os modelos mais "formais" de comunicação são preferidos, uma vez que a mensagem pode levar anos entre a emissão e a recepção (caso de certos registros fiscais) e, muitas vezes, o emissor nem sabe quem é o receptor da mensagem (caso de uma nota fiscal, que é emitida para comunicar ao oficial da fazenda quais são as mercadorias que estão circulando com um caminhão).

Observe-se que os conceitos acima estabelecidos dizem respeito aos sistemas de comunicações. E quanto às informações? — Para [Warnier,1984,p.14]:

> *A* palavra informações é fonte de confusão na medida em que ela designa, indiferentemente, o conteúdo do espírito humano ou a representação desse conteúdo em uma linguagem ... É muito perigoso confundir o conteúdo do espírito humano, cuja riqueza é incalculável e que se modifica sem cessar, com a expressão parcial e fixa desse conteúdo em uma linguagem armazenada sobre um suporte.

Para [Drucker,1998], a informação deverá, além de ter um significado, servir a um propósito. Se o conteúdo de uma mensagem não faz sentido, não foi estabelecida a comunicação e se o conteúdo não interessa ao destinatário, ainda que a mensagem tenha o seu significado reconhecido, não há informação.

Dentro das organizações sociais, as informações buscam orientar ou provocar a instrução das pessoas. Elas podem atender aos seguintes objetivos:

a) indicar um caminho para atingir-se uma meta: são informações geradas nos planejamentos;

b) indicar se um caminho que está sendo trilhado é o caminho que foi planejado: são informações de controle e

c) indicar uma direção a ser seguida, diante de duas ou mais opções: são informações de apoio aos processos decisórios.

Antes da formulação de um conceito para "sistemas de informação", é importante identificar-se um conjunto de processos que funcionam como um elo de ligação entre a informação e o sistema de comunicação de uma organização social. Tratam-se dos processos de armazenagem, recuperação e processamento de dados (genericamente denominado como "sistema de processamento de dados").

Observe-se que os esquemas mais elementares de comunicação (como o da ***fig.1.6.a***) não levam em conta o aspecto temporal de tramitação da mensagem entre o emissor e o receptor. Esse aspecto acaba por transformar-se num dos propósitos originais de um sistema de processamento de dados. Como pode ser observado na ***fig.1.6.b***, um receptor, para que possa transformar o conteúdo das mensagens em informação, precisa estar de posse desses conteúdos. Como as emissões das mensagens não obedecem, necessariamente, um mesmo padrão de comportamento dependente do tempo, a implementação de dispositivos de armazenagem e recuperação dos conteúdos dessas mensagens (dados) faz-se necessária para o estabelecimento do sincronismo de chegada dessas mensagens. Além disso, esses dados poderão sofrer um pré-processamento (mixagem), para chegar ao receptor num modelo estruturado segundo as suas conveniências.

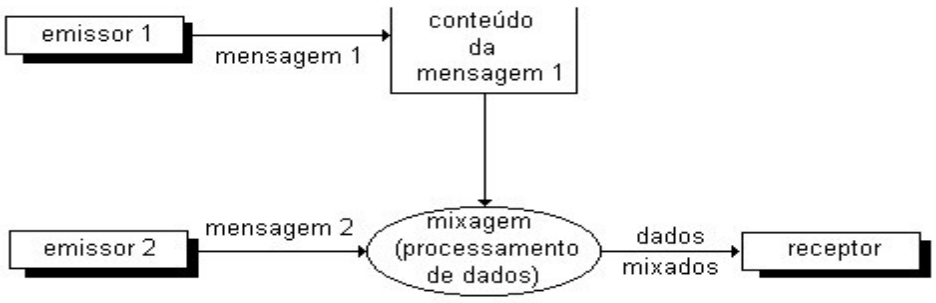


fig.1.6.b — Modelo de processamento de dados

1.7 — CONCLUSÃO

A partir dos conceitos estabelecidos sobre sistemas, comunicação, processamento de dados e informação, um sistema de informações gerenciais pode ser conceituado como "a interação de todos os processos necessários (tanto organizacionais quanto mentais) para que uma determinada pessoa numa organização social seja capaz de identificar:

se essa organização está com os conteúdos das variáveis que representam os eventos a que está sujeita, sob controle;

quais os caminhos a adotar para estabilizar as variáveis desses eventos, caso estejam fora de controle e

como e para onde avançar em busca de novas fontes de energia;

com o objetivo de promover a entropia negativa".

Uma forma mais simplificada disso seria dizer-se que um sistema de informações é um sistema de comunicação (composto de informantes e receptores) de mensagens que prestam-se ao controle de uma organização social. Agora, o que são mensagens que "prestam-se" a esse propósito? O que **qualifica** uma informação como útil ou não?

Antes da resposta a essa pergunta é interessante a instituição dos conceitos sobre "qualificação" de uma forma geral e esse é o objetivo do próximo capítulo.

CAPÍTULO 2 — CONCEITOS SOBRE QUALIDADE

Conforme [Bergman,1995], a história do movimento atual da qualidade iniciou-se na década de 20, quando Shewhart aplicou simples idéias de estatísticas em processos de manufatura. Assim, a partir dessa localização histórica, a pergunta a ser respondida nessa seção é: o que vem a ser “qualidade”? Que fenômeno é esse que vem transformando-se num tormento cada vez mais intenso para milhões de pessoas, principalmente nas últimas quatro décadas, e quais as suas implicações na análise de requisitos de sistemas de informações?

2.1 — DEFININDO QUALIDADE

As definições abaixo foram extraídas do [GDBCP,1979], e objetivam o estabelecimento de alguns parâmetros acadêmicos, menos para provocar discussão sobre o assunto e mais para criar uma fundamentação conceitual sobre a qual será construída a argumentação adiante:

a) “qualidade” é aquilo que caracteriza alguma coisa;

b) “qualificado” é aquele que tem *certas* qualidades; que é *distinto* e

c) “qualificação” é a ação ou efeito de qualificar; apreciação, juízo feito sobre alguma coisa.

A partir dessas definições, dizer-se que um artigo “tem qualidade” passa a não ter grande significado, uma vez que “ter qualidade” é “ter um conjunto de características” não necessariamente ideais ou boas. Já, dizer-se que um artigo é “qualificado” parece estar mais em consonância com os termos atualmente em voga sobre “ter qualidades” (ao menos parcialmente). Ser qualificado significa ter aquelas características que distinguem um artigo de outros, a partir de uma ação de qualificação, ou de julgamento. Um outro termo que também poderia ser empregado seria “virtuoso”: virtuoso seria um artigo que possui o conjunto de todas as virtudes, ou “boas” qualidades esperadas;

aquilo que é próprio (ou apropriado) e eficaz.

Independentemente do compromisso com uma definição de qualidade (ou algo que *definitivamente* qualidade *é*), muitos conceitos têm sido utilizados, sobre qualidade:

a) para Juran (1974, apud [Stange,1996]) “é adequação ao uso através da percepção das necessidades dos clientes”;

b) para Deming (1982, apud [Stange,1996]) “é perseguição às necessidades dos clientes e homogeneidade dos resultados do processo”;

c) para Crosby (1984, apud [Stange,1996]) “é conformidade do produto às suas especificações”;

d) para Feigenbaum (1986, apud [Stange,1996]) “é o conjunto de características incorporadas ao produto através do projeto e manufatura que determinam o grau de satisfação do cliente”;

e) para Ishikawa (1986, apud [Stange,1996]) “é rápida percepção e satisfação das necessidades do mercado, adequação ao uso dos produtos e homogeneidade dos resultados do processo”.

Observe-se a aparente importância que é dada sobre determinados aspectos, nesses conceitos: a satisfação das necessidades do cliente parece ser, para Juran, Deming e Ishikawa dependente da capacidade de capturar correta e completamente essa necessidade. Para Crosby e Feigenbaum, o ato da transformação da necessidade em especificação de projeto e processos parecem ser importantes e a homogeneidade dos resultados do processo parecem ser importantes para Deming e Ishikawa. Analisando-se os termos desses conceitos, percebem-se duas características interessantes ao contexto do presente trabalho: o enfoque sobre a especificação do artigo e o enfoque sobre a constância de resultados dos processos.

O objetivo da diferenciação entre esses enfoques não pode ir além dos objetivos didáticos, uma vez que eles complementam-se entre si na busca pela satisfação dos clientes. Observações encontradas na literatura indicam que as tentativas de concentração sobre um ou outro aspecto, isoladamente, precisam ser vistas com ressalvas. A série de normas ISO 9000 (serão detalhados os aspectos de normalização mais adiante), por exemplo, tem recebido críticas precisamente por concentrar-se

essencialmente nos processos. Segundo Kitchenham (apud [Tsukumo,1996,p.220]), "*esse foco no processo pode levar a um julgamento da qualidade que é virtualmente independente do produto em si. Isto é, a abordagem da manufatura, adotada pela ISO 9001 e pelo Capability Maturity Model (CMM) — advoga a conformidade ao processo mais do que à especificação*". O ideal seria que, assim como os indicadores de qualidade de um artigo são influenciados pelos processos utilizados no seu desenvolvimento, os processos fossem influenciados pelos indicadores de qualidade definidos para o artigo no planejamento.

2.2 — SISTEMAS PARA A QUALIDADE

Para que não sejam dissociados os objetivos estabelecidos na especificação dos artigos, dos objetivos dos processos que geram o artigo, surgem os sistemas para a qualidade. Duas novas expressões podem ser introduzidas nesse ponto: TQM e TQC:

a) TQM — Gerenciamento da Qualidade total (Total Quality Management): pode ser conceituado como um sistema responsável pelo planejamento e controle de todos os fatores, nos relacionamentos entre os elementos de uma organização social, que podem, de alguma forma, influenciar nos objetivos dos interessados sobre essa organização, quais sejam, os interesses dos clientes, fornecedores, funcionários, acionistas, sociedade em geral, governo, etc.;

b) TQC — Controle Total da Qualidade (Total Quality Control): pode ser conceituado como um subsistema do TQM, responsável pelo planejamento e controle de todos os fatores, nos relacionamentos entre os elementos de uma organização social, que podem, de alguma forma, influenciar na satisfação dos clientes, compreendendo todo o ciclo de vida dos artigos, desde a identificação dos seus requisitos até a assistência técnica, após a entrega.

Dos sistemas de qualidade (mais ou menos abrangentes, conforme acima) é derivado o conceito de qualidade total. A qualidade total diferencia-se da definição de qualidade, identificada anteriormente, pela sua abrangência. Conforme

[Paladini,1994,p.17], "... *a "Qualidade Total" está completamente direcionada para o consumidor; pela abrangência do conceito, envolve a todos na organização; pelo nível em que se deve colocar a questão, é uma das grandes metas da empresa, fixada em termos de políticas globais*".

Os processos de um sistema que busca a qualidade total visam, essencialmente, a estabilização e a melhoria de determinadas características dos relacionamentos entre os componentes da organização social onde esse sistema estiver inserido, buscando a satisfação das necessidades (as vezes até expectativas) de todos os interessados. Essas características, primeiramente, precisam ser definidas na forma de indicadores de qualidade (unidades de medida e quantidades de unidades) para, num segundo momento, serem identificados os fatores que exercem influência sobre essas características. Todo os processos deverão ser, então, orientados para o controle dos valores desses fatores. A medida do nível de satisfação das expectativas dos interessados será relativa à medida do ganho ou à medida da perda desses fatores, e o sentido da busca varia dependendo da razão ideal entre os valores dos indicadores de qualidade e os valores desses fatores. Conforme [Stange,1996], o sentido da busca dos valores de um fator pode ser de um dos três tipos a seguir (os exemplos de subcaracterísticas foram extraídos dos requisitos de qualidade de produtos de software segundo a norma ISO/IEC 9126):

a) medida nominal do fator (razão-ideal = 1, podendo, entretanto, ser definida por uma faixa estabelecida entre dois valores limítrofes), como a "presença de todos os dados necessários, e somente os necessários, num determinado relatório exigido pelo governo, conforme um determinado padrão legal definido", se a subcaracterística de qualidade for "conformidade" por exemplo;

b) quanto menor melhor (razão-ideal <= 1), como o "número de projetos sendo implementados ao mesmo tempo, por um projetista", se a subcaracterística de qualidade for "maturidade" por exemplo;

c) quanto maior melhor (razão-ideal >= 1), como o "tempo de experiência do analista na área mérito (contas a pagar, faturamento, controle da produção, etc.)" se a subcaracterística de qualidade for "adequação" por exemplo.

Esses fatores podem estar relacionados tanto à confiabilidade dos artigos que a organização produz, quanto à sua competitividade.

Shewhart sugeriu um método que permite a estabilização e a melhoria sistemática dos valores desses fatores. O modelo geral desse método está representado na ***fig.2.2.a*** e as etapas desse processo são:

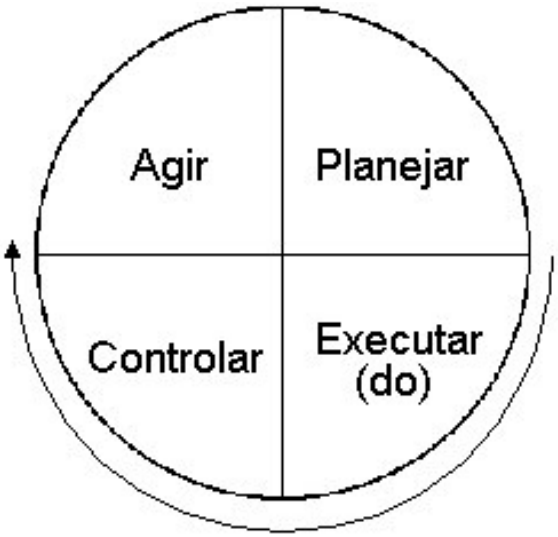


fig.2.2.a — Ciclo de aperfeiçoamento da qualidade de Shewhart (PDCA)

a) planejar: compreende o conjunto de atividades responsáveis:
   a.a) pela identificação e definição dos problemas a serem resolvidos e
   a.b) pelo estabelecimento dos objetivos de aperfeiçoamento;
b) executar: compreende o conjunto de atividades responsáveis:
   b.a) pela identificação das possíveis causas dos problemas (fatores que exercem influência sobre a característica com problema ou a ser melhorada);
   b.b) pela definição de diretrizes (processos com suas atividades e indicadores de qualidade) para atingir-se os objetivos de aperfeiçoamento e
   b.c) pelos testes dos processos, normalmente sobre uma produção piloto.

Caso os indicadores de qualidade das atividades que compõem os processos não tenham sido estabilizados, o controle não pode passar à próxima etapa (esse fluxo de controle pode ser observado na ***fig.2.2.b***);

c) controlar: compreende o conjunto de atividades responsáveis pela coleta dos dados que representam as características de qualidade do artigo em questão e comparação desses dados com os objetivos estabelecidos na etapa "planejar". Caso os objetivos não tenham sido atingidos, o controle retorna à etapa "executar";

d) agir: compreende o conjunto de atividades responsáveis:

d.a) pela implementação do modelo na produção e

d.b) pela institucionalização do modelo, buscando-se que tornem-se eficientes. Nesse ponto o fluxo de controle retorna à etapa "planejar", onde podem ser estabelecidos novos objetivos de aperfeiçoamento para o artigo em questão. Caso esses novos objetivos não possam ser estabelecidos, a resultante do processo é uma incógnita, podendo ser entendida tanto como a estabilidade do artigo quanto como a sua decadência, relativamente à evolução das necessidades dos clientes (veja a ***fig.2.2.b***).

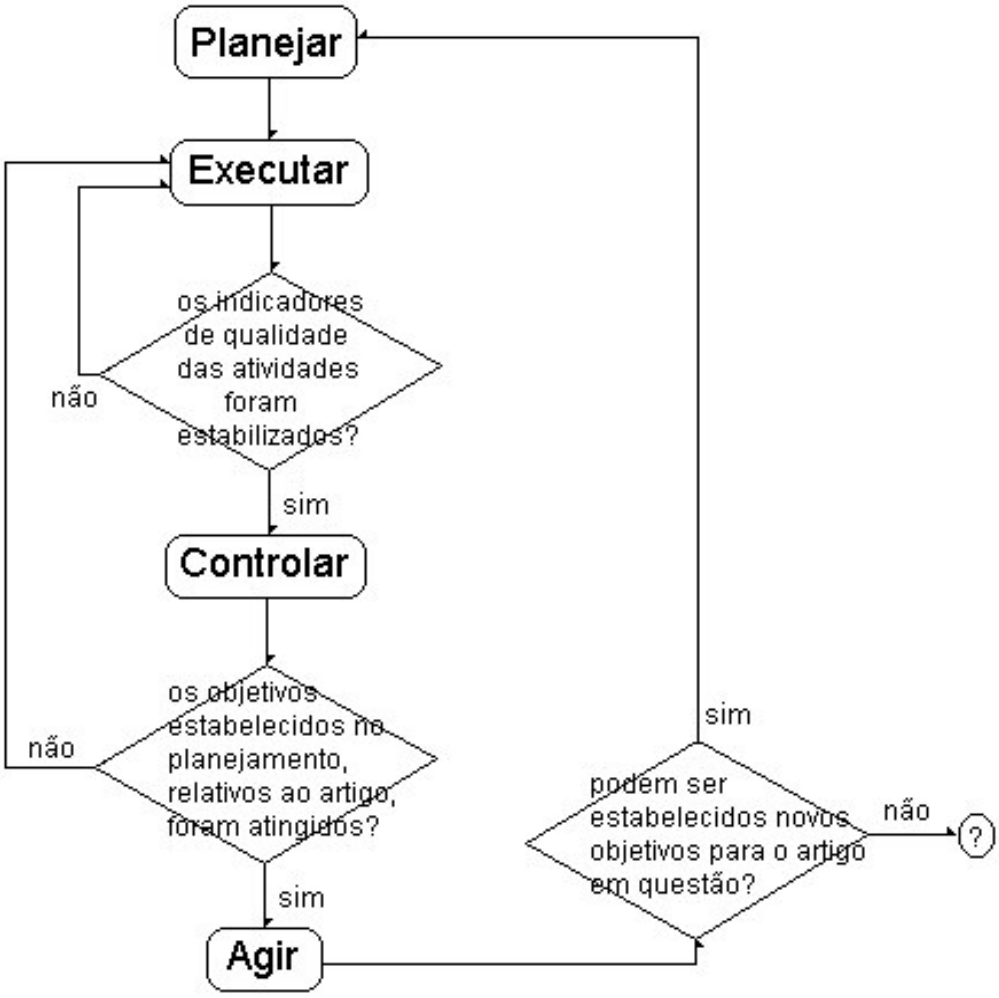


fig.2.2.b — Fluxo de controle entre as etapas do PDCA

2.3 — NORMAS E PADRÕES DE QUALIDADE

As normas e padrões da era contemporânea foram inicialmente desenvolvidas objetivando-se o intercâmbio de componentes. A área de informática teve sua herança na componentização de elementos eletrônicos e mecânicos, usados na construção dos computadores e periféricos.

Padrões mais recentes, entretanto, buscam a uniformização do maior número possível das formas que se prestam aos sistemas de comunicação, indo além da comunicação entre componentes mecânicos e eletrônicos (plug-in), para atingirem

também os padrões de comunicação entre as pessoas (entre si), entre as pessoas e as máquinas e até mesmo entre softwares diferentes. O objetivo principal desses padrões é a redução nos desgastes gerados pelos processos legais, oriundos de discórdias por causa do que foi, ou não foi, dito na formalização dos contratos de negociação de artigos, e nos desgastes gerados pela dificuldade na operação de determinados dispositivos cibernéticos.

Diversas são as instituições de normalização ao redor do mundo. Entre as que possuem prospecção internacional podem ser citadas como exemplos as DIN (Deutches Institut für Normung — Instituto Alemão para a Normalização) e ANS (American National Standard — Padrões Nacionais Estadunidenses). O Brasil também possui um fórum nacional de normalização que é a ABNT (Associação Brasileira de Normas Técnicas).

Além dos padrões internos de cada país, existem movimentos pela universalização desse padrões em nível mundial. Para esse fim, foi criada em 1947 a International Organization for Standardization, uma federação mundial com, atualmente, mais de 100 (cem) organizações nacionais de padronizações (entre elas a ABNT), representando mais de 90 países, responsáveis por mais de 95% da produção industrial mundial. Com sede em Genebra — Suíça, e tendo como principal atividade a elaboração de padrões para especificações e métodos de trabalho nas mais variadas áreas (menos na área da eletrotécnica e eletrônica, estando essa área sob a responsabilidade da International Electrotechnical Comission (IEC)), a International Organization for Standardization, para referir-se a si própria, adotou o nome “ISO”. Conforme [Normandeau,1998], ISO na Grécia antiga significava “igual”, e é o prefixo de centenas de palavras. Por exemplo:

a) isobárico: pressão atmosférica igual;

b) isométrico: de dimensões iguais;

c) isonomia: igualdade civil e política.

Por esse motivo o prefixo ISO veio a ser associado com a concepção de padrão, ou norma de qualidade. Esse prefixo foi adotado pela International Organization for Standardization para referir-se a si própria em todas a línguas (ao invés de usar IOS em inglês ou OIN em francês). O principal objetivo da ISO é o desenvolvimento de

padrões mundiais, com vistas à facilitação do intercâmbio internacional de produtos e serviços e à criação de uma cooperação intelectual, científica, econômica e técnica, entre as nações.

O prefixo “ISO” faz parte das referências de normas desenvolvidas na ISO. Por exemplo, para conceituar qualidade existe uma norma que é a ISO 8402. Segundo essa norma, “*qualidade é a totalidade das características de uma entidade,* (artigo de software, por exemplo) *que lhe confere a capacidade de satisfazer às necessidades explícitas e implícitas*”[i] [ABNT,1996].

Como visto anteriormente, as normas da ISO têm por objetivo encorajar o desenvolvimento de padrões de artigos. Esses padrões são definidos em séries de normas que foram inicialmente orientadas segundo um determinado foco específico: ou sobre a especificação do artigo ou sobre a constância de resultado dos processos (normas de procedimentos que buscam indicadores de qualidade, mantidos sob controle estatístico). A abordagem que tem recebido mais atenção, atualmente, é a abordagem orientada ao processo e o impulso dessa abordagem dá-se principalmente por causa da série de normas ISO 9000.

A série de normas ISO 9000 é um conjunto de normas que representam, atualmente, um consenso internacional sobre as características essenciais de um sistema de garantia da qualidade. Elas foram publicadas em 1987 e revisadas em 1994 e, conforme [Normandeau,1998], são divididas em quatro grupos:

a) ISO 9004: trata-se de um guia básico que contém os procedimentos de qualidade ISO;

b) ISO 9003: limitado aos testes, define um modelo que cobre o controle da produção, verificação e ensaios finais dos produtos;

c) ISO 9002: define um modelo de qualidade considerando a produção e a instalação dos produtos;

d) ISO 9001: define um modelo de gerenciamento que cobre desde a concepção e o desenvolvimento de um produto, passando pelo processo

[i] Conforme [Azevedo,1998], *“as necessidades explícitas são requisitos definindo as condições e objetivos propostos pelo produtor, e as necessidades implícitas englobam: funções básicas do produto, as diferenças entre os possíveis usuários, implicações éticas e de segurança, visões subjetivas e necessidades inconscientes para os usuários”.*

de manufatura e instalação, e atingindo os serviços de assistência técnica. Ela propõe-se a cobrir todos os aspectos do relacionamento entre o fornecedor e o cliente.

2.4 — QUALIDADE EM SOFTWARE

Como não poderia ser diferente, os programas de qualidade em software também estão focados sobre os dois aspectos já identificados anteriormente: a qualidade do produto de software e a qualidade dos processos de software.

2.4.1 — QUALIDADE DE PRODUTO DE SOFTWARE

Para [Tsukumo,1996,p.220]:

> ... a abordagem da qualidade do produto ... ... procura definir fatores ou características que evidenciem a qualidade de um produto. Essas características são definidas para representar necessidades e desejos dos que usam ou são afetados pelo software. Essa abordagem não pretende definir ou julgar como o produto foi desenvolvido, mas se ele atende à definição da qualidade dada pelo ISO 8402.

As normas que tratam sobre qualidade de produtos de software são compostas de dois aspectos principais: as características de qualidade, do produto de software em si, e as técnicas a serem utilizadas para o teste dessas características de qualidade. Conforme [Andrade,1996,p.78]:

> Os aspectos técnicos para avaliação da qualidade do produto de software são abordadas em três Normas:
>
> . ISO/IEC 9126 — *Características* de qualidade de software;
>
> . ISO/IEC 14598 — Guias para avaliação de *produtos* de software; e
>
> . ISO/IEC 12119 — Requisitos de qualidade e testes de pacotes de software.

No Brasil, segundo o relatório [MCT,1998], 73,5% das empresas de software não conheciam as normas ISO/IEC 12119/9126, em 1997, e apenas 5,6% conheciam e usavam-nas. A seguir, serão apresentadas as estruturas dessas normas.

2.4.1.1 — Norma ISO/IEC 14598

Essa norma é composta de um grupo de guias que servem de apoio ao planejamento e ao controle de uma avaliação da qualidade de produtos de software. Esse conjunto de guias é composto de seis partes, sendo:

a) parte 1 — visão geral da estrutura da norma e do processo de avaliação;

b) parte 2 — composta das atividades de planejamento e gerenciamento do processo de avaliação;

c) parte 3 — processo do desenvolvedor: composta das atividades de avaliações que deverão ser feitas durante um processo de desenvolvimento de softwares (quando for o caso de desenvolvimento de um software);

d) parte 4 — processo do comprador: composta das atividades de avaliações que deverão ser feitas durante um processo de compra de softwares (quando for o caso de compra de um software);

e) parte 5 — processo do avaliador: ciclo de vida de avaliação e suas atividades;

f) parte 6 — módulos de avaliação: pacotes de métodos e dispositivos de apoio aos processos de avaliação do desenvolvedor, do comprador e do avaliador.

2.4.1.2 — Norma ISO/IEC 12119

A norma é subdividida em dois grupos: os requisitos de qualidade e as instruções para testes. Quanto aos requisitos de qualidade, eles são subdivididos em:

a) requisitos de descrição do produto: a descrição do produto apresenta as

principais propriedades do pacote, e deverá estar à disposição independente do compromisso de compra, por parte do potencial usuário. Segundo a norma "*a Descrição de Produto é um documento expondo as propriedades de um pacote de software, com o objetivo de auxiliar os potenciais compradores na avaliação da adequação do produto para sua aquisição*" [apud IEES,1997]. Essa descrição deverá conter:

a.a) identificação do produtor/fornecedor;

a.b) identificação do produto (nome, versão, etc.);

a.c) indicação da aplicação geral;

a.d) requisitos mínimos de hardware e software;

a.e) itens a serem entregues na compra do pacote (discos, manuais, caixa, etc.);

a.f) principais funcionalidades disponíveis no pacote;

a.g) maneiras para evitar-se o acesso não autorizado ao software (senhas, etc.);

a.h) valores limítrofes (número máximo de registros, tamanho das chaves, tempos de resposta, etc.);

a.i) procedimentos de segurança em caso de falhas;

a.j) detalhes sobre a interface com o usuário;

a.k) conhecimentos específicos, requeridos para a aplicação do produto;

a.l) dispositivos e métodos para adaptação do produto às necessidades específicas dos usuários;

a.m) serviço de manutenção e suporte técnico;

a.n) interfaces disponíveis, para outros produtos e possibilidade de portabilidade para outros ambientes operacionais e

a.o) proteções técnicas contra infração a direitos autorais;

b) requisitos da documentação do usuário: trata-se do conjunto de documentos a serem fornecidos para a utilização do software. Esses documentos

deverão ter uma apresentação e uma organização que facilitem o seu manuseio, e os requisitos dessa documentação são: completeza, correção, consistência e inteligibilidade;

c) requisitos dos programas e dados: os programas e dados deverão estar em conformidade com todos os requisitos declarados na documentação, sendo levados em conta os mesmos requisitos da norma ISO/IEC 9126, com ênfase nos requisitos: funcionalidade, confiabilidade e usabilidade.

Quanto às instruções para testes, são formalizados os métodos de teste, que são, essencialmente, planejados segundo os requisitos de qualidade, identificados anteriormente:

a) fase de pré-requisitos de teste: identifica a presença dos itens e dos componentes do software, prevê o acesso ao material de treinamento, caso esse material seja mencionado na descrição do produto;

b) fase de atividade de teste: todos os requisitos especificados nas documentações e nos programas deverão ser testados;

c) fase de registros e relatórios de teste: deverão ser coletados os resultados dos testes e esses resultados precisam ser tabulados e analisados;

d) fase de teste de acompanhamento: para o caso de teste em programas já testados anteriormente. Nesse caso, as partes que não estavam em conformidade deverão ser testadas, bem como todas as partes que podem sofrer influências dessas alterações.

2.4.1.3 — Norma ISO/IEC 9126 (NBR 13.596)

Os requisitos de qualidade de produtos de software podem ser agrupados em características e subcaracterísticas, da seguinte forma [ABNT,1996]:

a) funcionalidade: trata-se das funções que satisfazem as necessidades dos usuários:

a.a) adequação: é apropriado ao uso, conforme especificado;

a.b) acurácia: geração de resultados nos níveis conforme acordado;

a.c) interoperabilidade: capacidade de interação com outros softwares, conforme especificado;

a.d) conformidade: de acordo com normas e leis em vigor;

a.e) segurança de acesso: capacidade de evitar o acesso não autorizado;

b) confiabilidade: indica se o software mantém um determinado nível de desempenho, sob condições pré-determinadas:

b.a) maturidade: freqüência de falhas, causadas por defeitos no software;

b.b) tolerância a falhas: capacidade de manter um determinado nível de desempenho em caso de falhas;

b.c) recuperabilidade: capacidade de restabelecimento aos níveis de desempenho especificados, em caso de falhas;

c) usabilidade: atributos que indicam o esforço necessário ao uso do software:

c.a) inteligibilidade: esforço necessário para o usuário compreender o conceito lógico da aplicação;

c.b) apreensibilidade: esforço necessário para o usuário aprender a usar o software;

c.c) operacionalidade: esforço necessário para o usuário operar o software;

d) eficiência: relacionamento entre o nível de desempenho do software e a quantidade de recursos utilizados:

d.a) comportamento em relação ao tempo: refere-se aos acordos sobre tempos de resposta e de processamento do software;

d.b) comportamento em relação aos recursos: refere-se aos acordos sobre a quantidade de recursos usados durante o uso do software;

e) manutenibilidade: atributos que evidenciam o esforço necessário à execução de modificações especificadas:

e.a) analisabilidade: identifica o esforço necessário à identificação de problemas;

e.b) modificabilidade: identifica o esforço necessário a remoção de problemas ou adaptação à mudanças;

e.c) estabilidade: evidências sobre os riscos de efeitos inesperados em caso de mudanças;

e.d) testabilidade: identifica o esforço necessário para a execução de testes em caso de modificações;

f) portabilidade: capacidade de operar em ambientes operacionais diferentes:

f.a) adaptabilidade: identifica a capacidade de adaptar-se a ambientes diferentes, conforme especificado;

f.b) capacidade para ser instalado: identifica o esforço necessário à instalação do software;

f.c) conformidade (quanto à portabilidade): atributos do software que identificam o nível de padronização no que se refere à portabilidade;

f.d) capacidade para substituir: identifica o esforço necessário para usar o software em substituição a outro já instalado.

2.4.2 — CONSIDERAÇÕES SOBRE MODELOS

Os modelos são representações, ou a abstrações, de determinados pontos de vista de uma realidade. Conforme [Wille,1986] os modelos são uma tentativa de imitar os sistemas. Dessa forma, servem ao propósito de materializar uma mensagem e também prestam-se para testar certos princípios físicos da realidade que representam (a um custo e risco menores do que se o teste fosse efetuado sobre o sistema real) buscando a predição de determinados padrões de comportamento, constituindo-se, portanto, em importantes ferramentas para o analista de sistemas. Entretanto, apesar do desenvolvimento de sistemas ser facilitado com o uso de modelos (como é o caso da UML (Unified Modeling Language)), para [Furlan,1998, pag.98] "*criar modelos é um trabalho altamente criativo, e não há uma solução final ou uma resposta correta que possa ser verificada ao final do trabalho*".

Na ***fig.2.4.2.a*** estão representados os principais tipos de modelos. Esses tipos servem aos seguintes propósitos:

a) modelos mentais: por serem uma capacidade natural do ser humano, constituem-se no modo pelo qual são formuladas a maioria das abstrações para o estabelecimento de planos e teste de hipóteses para tomadas de decisões;

b) modelos simbólicos: dividem-se em modelos verbais (oral e escrito) e matemáticos. Os modelos verbais possuem uma ampla gama de aplicações e, provavelmente, são os mais utilizados depois dos modelos mentais. Os modelos matemáticos são bastante atrativos (desde que os seus procedimentos sejam conhecidos) por duas características principais: são muito precisos e concisos;

c) modelos físicos: divididos entre icônicos e análogos, podem ser exemplificados com mapas rodoviários (gráficos), maquetes (reduzidos) e medidores de combustível (análogos).

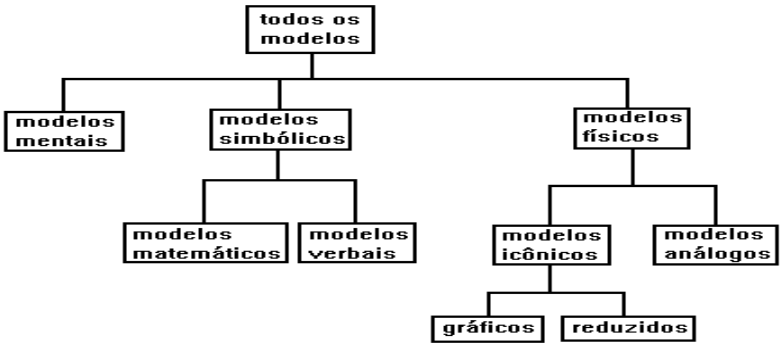


fig.2.4.2.a — principais tipos de modelos

2.4.3 — MODELOS DE PROCESSOS DE SOFTWARE

Conforme [GDBCP,1979], processos são:

a) uma série de fenômenos que se sucedem e são ligados pelas relações de causa e efeito;

b) um conjunto dos atos por que se realiza determinada operação;

c) uma maneira de operar, resolver ou ensinar.

Conforme [Poulin,1998], processos são meios pelos quais pessoas, procedimentos, métodos, equipamentos e ferramentas são integrados para produzir um resultado final desejado.

Observando-se as aplicações dos conceitos sobre sistemas e processos, em determinados casos essas terminologias são empregadas indistintamente. Entretanto, nesse trabalho, buscou-se a diferenciação em dois aspectos básicos:

a) sistemas são definidos por princípios e leis que regulam uma determinada ordem de fenômenos, processos são definidos apenas como a implementação da relação de causa e efeito desses fenômenos e

b) sistemas buscam uma relação de influências em malha fechada com o meio no qual encontram-se inseridos, onde todas as causas e todos os efeitos são, ao mesmo tempo, causa e efeito; processos podem ser compreendidos como um conjunto linear de influências em que o último efeito não é a primeira causa.

Processos de software podem ser conceituados como a definição dada por [Poulin,1998], aplicada ao desenvolvimento e manutenção de softwares e produtos relacionados, ou seja, “processos de software são meios pelos quais pessoas, procedimentos, métodos, equipamentos e ferramentas são integrados para produzirem um determinado artigo de software”.

Existem diversos modelos de processos de software, entre eles podem ser citados: a norma ISO 9000-3 e o modelo SEI-CMM. Adiante, segue a descrição das estruturas desses modelos.

2.4.3.1 — Norma ISO 9000-3

Trata-se de um guia para a aplicação da norma ISO 9001 para o desenvolvimento, fornecimento e manutenção de softwares. Essa norma foi criada em

junho de 1993, e é dividida em três partes:

a) estrutura: trata de questões relacionadas ao sistema de qualidade na organização, sendo definidas as responsabilidades tanto do comprador quanto do fornecedor. Essas questões são: responsabilidade da administração, sistema de qualidade e auditorias internas desse sistema, ações corretivas;

b) atividades do ciclo de vida: são as atividades relacionadas ao desenvolvimento de software propriamente. O ciclo de vida em si não é imposto pela norma, entretanto, os seguintes grupos de atividades precisam ser contemplados: análise crítica do contrato; especificação dos requisitos do comprador; planejamento do desenvolvimento; planejamento da qualidade; projeto e implementação; ensaios e validação; aceitação; cópia, entrega e instalação; manutenção;

c) atividades de suporte: nessa parte, a norma define quais as atividades que servirão de apoio às atividades do ciclo de vida (acima): gestão de configuração; controle de documentos; registros da qualidade e medição; regras, práticas e convenções; ferramentas e técnicas; aquisição; produto de software incluído; treinamento.

2.4.3.2 — Modelo SEI-CMM

Desenvolvido pelo “Software Engineering Institute” (SEI) da “Carnegie Mellon University”, a criação desse modelo foi motivada principalmente por projetos do Departamento de Defesa dos Estados Unidos da América (no Brasil, segundo o relatório [MCT,1998], 71% das empresas de software não conheciam esse modelo em 1997). Conforme [Tsukumo,1996], ele diferencia-se da norma ISO 9000-3, essencialmente, porque preocupa-se explicitamente com o contínuo aperfeiçoamento do processo, o que não ocorre com a ISO 9000-3.

É definido com uma estrutura que compreende 5 níveis de maturidade, baseados nos princípios da qualidade estabelecidos por Shewhart, Deming, Juran e

Crosby. Cada nível de maturidade compreende um conjunto de características e áreas chaves de processo (Key Process Areas — KPA's):

a) **nível 1 — inicial:** não há processos definidos; o gerenciamento, quando existe, é para fins específicos;

b) **nível 2 — repetitivo:** o planejamento e o controle de projetos são estabelecidos e definidos na forma de processos, principalmente orientados à administração de custos, cronogramas e funcionalidades do software. Nesse nível são implementados, também, os processos de engenharia de software, permitindo que tornem-se repetitíveis. **KPA's:** gerenciamento de requisitos, planejamento de projeto de software, acompanhamento de projeto de software, gerenciamento de contratos, qualidade assegurada de software, gerenciamento de configuração;

c) **nível 3 — definido:** nesse nível são formalizados (padronizados e documentados) e institucionalizados os processos implementados no nível anterior. Nenhum projeto (seja de desenvolvimento, seja de manutenção) é implementado sem a utilização desse processo organizacional. **KPA's:** foco no processo da organização, definição do processo da organização, programa de treinamento, gerenciamento integrado do software, engenharia de produto de software, coordenação entre grupos, revisões;

d) **nível 4 — gerenciado:** focalizado na qualidade do produto e do processo, leva a organização à condição de ser capaz de planejar a qualidade (implementando especificidades de projetos aos processos padronizados já existentes), a partir da medição dos indicadores de qualidade (tanto de produtos quanto de processos). **KPA's:** gerenciamento quantitativo do processo, gerenciamento da qualidade do software;

e) **nível 5 — otimizado:** refere-se ao melhoramento contínuo do processo, a partir da realimentação sistemática dos processos com os indicadores colhidos nos controles, e a partir do gerenciamento das mudanças tecnológicas. **KPA's:** gerenciamento de mudanças tecnológicas, gerenciamento de mudanças nos processos, prevenção de defeitos.

2.5 — TÉCNICAS PARA A QUALIDADE TOTAL

Conforme [Paladini,1994,p.66], as técnicas para a qualidade total envolvem ferramentas “que são dispositivos, procedimentos gráficos, numéricos ou analíticos, formulações práticas, esquemas de funcionamento, mecanismos de operação, enfim, métodos estruturados para viabilizar a implantação da Qualidade Total” e estratégias “que são metodologias para implantar mecanismos destinados a produzirem qualidade em qualquer atividade, processo, serviço ou produto da organização” (p.93).

O objetivo dessa seção é a identificação de algumas ferramentas para o planejamento e o controle da qualidade que já estão disponíveis. A estratégia de implantação do novo modelo, bem como algumas novas ferramentas específicas à análise de requisitos, serão apresentadas no capítulo 4.

2.5.1 — DIAGRAMA DE CAUSA E EFEITO

Criado em 1943 por Kaoru Ishikawa (por isso também é conhecido como diagrama de Ishikawa), tem por objetivo principal a visualização de um processo, ou seja, o mapeamento entre uma série de fenômenos que se sucedem e que são ligados entre si pelas relações de causa e efeito. Segundo [Ishikawa,1993]:

> A análise de processo é a análise que esclarece a relação entre os fatores de causa no processo e os efeitos como qualidade, custo, produtividade, etc., quando se está engajado no controle de processo. O controle de processo tenta descobrir os fatores de causa que impedem o funcionamento suave dos processos. Ele procura assim a tecnologia que possa efetuar o controle preventivo. Qualidade, custo e produtividade são efeitos ou resultados deste controle de processo.

Com a forma de uma espinha de peixe, o modelo original sugeria quatro grandes grupos de causas que deveriam ser analisadas. Esses quatro grupos (também conhecidos como quatro M’s) eram: materiais, mão-de-obra, métodos e máquinas. Versões mais recentes desse diagrama sugerem a análise orientada por seis grandes grupos de causas: materiais, mão-de-obra, métodos, máquinas, medidas e meio-ambiente (conforme [Vieira,1994]). A ***fig.2.5.1.a*** representa um modelo geral do

diagrama. Dele pode-se compreender o funcionamento da ferramenta:

a) na extremidade direita do eixo central é apresentado o sintoma que representa o problema a ser resolvido, ou o efeito desejado do processo;

b) a esse eixo central estão ligadas as diversas causas que de alguma forma cooperam para que o sintoma ou efeito ocorra;

c) cada causa, por ocasião de sua análise, transforma-se também num eixo central e a esse eixo são ligadas causas menores. Essa iteração pode ocorrer indefinidamente, até que as causas mais elementares sejam identificadas.

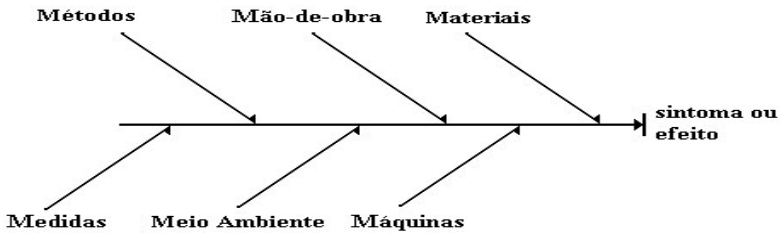


fig.2.5.1.a — Diagrama de causa e efeito de Ishikawa

O processo de decomposição dos efeitos em suas causas poderá ser feito com o apoio de ferramentas adicionais, como:

a) diagrama de dispersão, que permite a visualização gráfica da correlação entre as diversas causas, explicitando se essa correlação é positiva (p.exe.: se aumentarmos a rotação do motor, aumenta-se o consumo de combustível), negativa (p.exe.: se diminuirmos os preços, aumentam-se as vendas) ou nula (não há correlação entre as causas);

b) os histogramas, que facilitam o estudo de variações de comportamento do sistema, por classes de freqüências, representadas sobre um sistema de eixos cartesianos, onde sobre o eixo das ordenadas representam-se as classes de freqüências e sobre o eixo das abscissas representa-se o aspecto sob estudo (tempo de espera numa fila, velocidade, custo, etc.).

A importância desse dispositivo como instrumento de análise de requisitos de sistemas de informações é caracterizada por dois motivos principais:

a) a percepção de determinados sintomas é a principal forma de identificação de que existe algum problema no sistema. Por esse motivo, os usuários desse sistema tendem a sugerir determinadas soluções, orientados por esses sintomas;

b) essas sugestões dos usuários configuram-se como hipóteses de solução e a ferramenta auxilia no teste dessas hipóteses, na medida em que as verdadeiras causas dos problemas forem identificadas. Quando identificadas, elas podem ser mapeadas, podendo levar, eventualmente, ao controle sobre as variáveis que têm influência sobre os efeitos desejados ou sintomas indesejados.

As causas que geram um determinado resultado podem atuar de três formas diferentes:

a) um efeito é gerado por uma causa isolada, dentre um conjunto de causas possíveis (por exemplo: sintoma: o microcomputador não liga, causa: falta energia elétrica **ou** a fonte do micro está queimada **ou** ...);

b) um efeito é gerado pela coincidência de uma combinação de causas, o que caracteriza um fenômeno (por exemplo: efeito: chuva ciclônica, causas: havia uma depressão barométrica **e** essa diferença de pressão causou um deslocamento de ar quente **e** a umidade relativa desse ar era superior a um determinado percentual **e** nesse deslocamento houve um encontro do ar quente com uma massa de ar frio **e** a partir desse encontro foi desencadeado um processo de condensação favorecido pela ação de soluções de cloreto de sódio evaporadas do mar **e**...);

c) um efeito é gerado por um conjunto de causas simultâneas, em que cada uma delas participa com uma parte do valor total do efeito (por exemplo: sintoma: a pizza não está com um sabor agradável, causa: muito salgada **e/ou** a massa é muito fina **e/ou** com pouco queijo **e/ou** ...).

A importância da atuação de cada causa sobre os efeitos em estudo, muitas vezes não é conhecida. Assim, torna-se necessária a medição da amplitude da responsabilidade de cada causa sobre o efeito desejado para, então, proceder-se à ponderação dessas responsabilidades. A terceira forma de atuação das causas,

apresentada na alínea “c” acima, caracteriza-se precisamente pela dificuldade que apresenta no momento em que é necessária uma tomada de decisão a respeito de sobre quais causas deve-se exercer controle. Nesse momento mostra-se importante uma outra ferramenta: o gráfico de Pareto.

2.5.2 — GRÁFICO DE PARETO

Batizado como gráfico de Pareto por causa de seu criador, Vilfredo Pareto, essa ferramenta permite uma visualização estruturada em ordem de importância das causas, sobre um determinado resultado que deseja-se atingir (vide ***fig.2.5.2.a***).

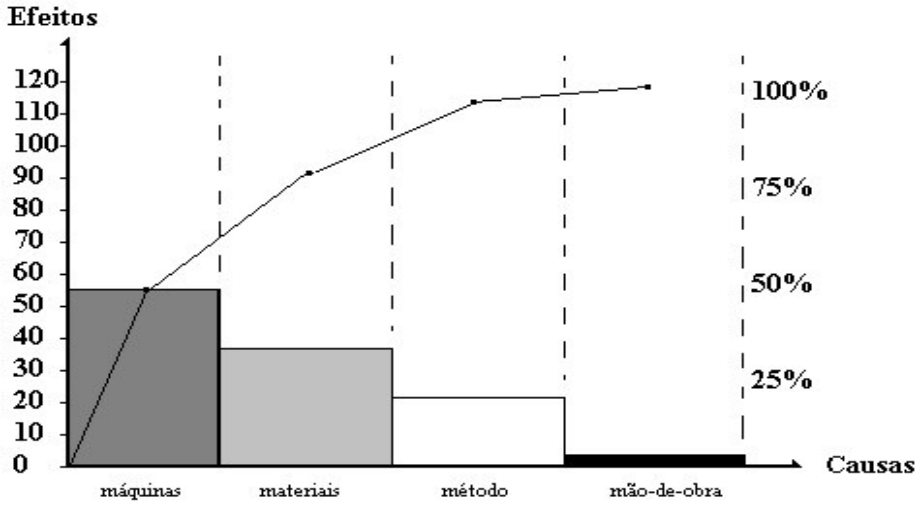


fig.2.5.2.a — Gráfico de Pareto

Esse aspecto é particularmente útil quando o sintoma ou efeito é gerado por um conjunto de causas simultâneas (como visto anteriormente para o diagrama de causa e efeito) e os recursos disponíveis (tempo, dinheiro, etc.) não forem suficientes para o controle de todas as variáveis que compõem essas causas. Assim, a identificação das causas mais influentes torna-se necessária.

Pode-se aplicar o gráfico de Pareto a partir das causas identificadas no

diagrama de causa e efeito (exemplificadas na ***fig.2.5.2.a*** com as mesmas classes de causas apresentadas na ***fig.2.5.1.a***), ponderando-se essas causas pela unidade de medida do resultado desejado do processo (custo, lucro, etc.). Esse gráfico apresenta, essencialmente, dois conjuntos de informações, representados por dois "sub-gráficos" sobrepostos: o primeiro conjunto ("sub-gráfico" de colunas) representa as causas com seus valores, e o segundo conjunto ("sub-gráfico" de linhas) representa as causas com suas importâncias relativas (percentuais).

2.5.3 — QFD — DESDOBRAMENTO DA FUNÇÃO QUALIDADE

Identificados os resultados desejados e as variáveis mais importantes a serem controladas, resta identificar-se o momento mais adequado para que o conteúdo dessas variáveis sejam coletados, de tal forma que possam transformar-se em informação de controle. Isso é simples de ser compreendido, mas, dependendo do volume de variáveis a serem controladas, difícil de ser implementado na prática. É comum que as variáveis exerçam influência sobre mais de um efeito (muitas vezes para efeitos diferentes as variáveis têm sentido de busca diferente) e, quando o volume tanto de variáveis quanto de efeitos é na casa das centenas, o mapeamento entre o registro e o controle dessas variáveis, mapeamento esse que tornará possível o projeto do produto, precisa de uma ferramenta que o suporte. Assim, ao longo da cadeia produtiva de um resultado (ou ao longo do processo que gera um resultado), é necessário que o resultado desejado seja preservado, em outras palavras, a voz do cliente, que estabelece o resultado desejado, precisa ser ouvida ao longo do processo, com o significado que aquele resultado representa em cada ponto do processo.

Para resguardar-se a voz do cliente, surgiu no Japão da década de 70 uma metodologia, idealizada pelo Professor Yoji Akao, que conta hoje com simpatizantes em todo o mundo. Trata-se do QFD - Quality Function Deployment, ou Desdobramento da Função Qualidade. O QFD é um conjunto formal de disciplinas e tecnologias que permitem, de forma sistemática, o mapeamento e a comunicação das funções impostas pelos usuários, às diversas atividades num processo produtivo, de forma que cada uma dessas atividades permaneça orientada por essas funções.

Um exemplo da aplicação dessa metodologia na área de software pode ser vista em [El Boushi,1994]. Trata-se de um software desenvolvido no Laboratório de Propulsão a Jato (Jet Propulsion Laboratory — JPL) da NASA, dentro do projeto AMMOS (Advanced Multi Mission Operation System), projeto esse que visa todas as futuras missões envolvendo o JPL, desde missões na Terra, até viagens interplanetárias a Marte e Saturno. Conforme os autores, o QFD possibilitou uma abordagem metódica para a captura da voz dos usuários e tratamento dos seus requisitos.

2.5.3.1 — Conceito geral

Para Akao (apud [Fiates,1995,p.63]), o QFD *"é uma metodologia para a conversão de demandas dos consumidores em características de qualidade*" Essa conversão ocorre através de desdobramentos sucessivos desde a identificação dos requisitos do projeto, percorrendo todo o ciclo de vida do produto até a aquisição da matéria-prima. [Stange,1995] evidencia dois aspectos próprios do QFD: um é a sinergia obtida pelo comprometimento de todos os setores que participam de um processo produtivo e outro é o desenvolvimento da qualidade desde a concepção de um artigo e, indo além da sua utilização por parte do cliente, até a preocupação com o meio-ambiente.

O QFD remonta à área de conhecimento da Análise do Valor, indo porém além da análise do artigo (suas funções, a importância dessas funções e suas relações com os componentes do artigo) e da busca de uma redução do custo ou a melhoria no desempenho das funções desse artigo. Ele preocupa-se, além dessas coisas, com a relação de toda a organização com o meio no qual ela está inserida através desse determinado artigo e com a relação dos componentes dessa organização entre si, através desse mesmo artigo, tudo isso antes que ele seja produzido.

O modelo original do QFD era formado por dois componentes:

a) desdobramento da qualidade: para os aspectos relacionados ao artigo em questão. Esta fase pode ser caracterizada por descritores e medidas de ***qualidade*** e

b) desdobramento da função: para os aspectos relacionados aos meios para a obtenção desse artigo. Esta fase pode ser caracterizada por descritores e medidas de ***produtividade***.

Desenvolvimentos mais recentes, entretanto, (conforme [Akao,1993]) acrescenta ao modelo já existente mais dois componentes:

a) desdobramento da confiabilidade: [Cheng,1995] conceitua confiabilidade como sendo a probabilidade e a habilidade de um item desempenhar uma função exigida, sob condições estipuladas, durante um determinado período de tempo e

b) desdobramento do custo, objetivando oferecer uma nova visão para a análise de custos, antes fundamentada na premissa de que o valor de um artigo poderia ser obtido pela relação entre a função e o custo (valor = função / custo) e agora essa relação seria dada entre a qualidade e o custo (valor = qualidade / custo).

2.5.3.2 — Um modelo de QFD

O QFD é desenvolvido em diversas fases. Cada fase é representada por uma matriz em que, pelo lado esquerdo (linhas) entra-se com um conjunto de requisitos oriundos do cliente ou da fase anterior, representando “o que” deseja-se, e pelo lado superior (colunas) busca-se “como” atender esses requisitos, através de medidas e descritores técnicos, para que sejam atingidas as qualidades esperadas com o artigo em questão.

Com o intuito de transmitir uma noção geral do funcionamento dessa metodologia, será apresentado um modelo simplificado do QFD, derivado do original de Akao, conhecido como modelo de Makabe, modelo esse que tem popularizado o QFD no ocidente. A ***fig.2.5.3.2.a*** representa o modelo conceitual de Makabe, sendo esse modelo composto por quatro fases: desdobramento dos requisitos do cliente em requisitos de projeto, desdobramento dos requisitos de projeto em requisitos de componentes, desdobramento dos requisitos de componentes em requisitos de

processos e desdobramento dos requisitos de processos em requisitos de produção.

O princípio do relacionamento entre as quatro fases está em que os requisitos mais relevantes (somente os mais importantes) obtidos na fase anterior servirão como entrada para a fase seguinte. Por exemplo, os requisitos mais relevantes de projeto, obtidos na primeira fase, serão os requisitos de entrada na segunda matriz (vide as setas e a semelhança dos títulos na ***fig.2.5.3.2.a***).

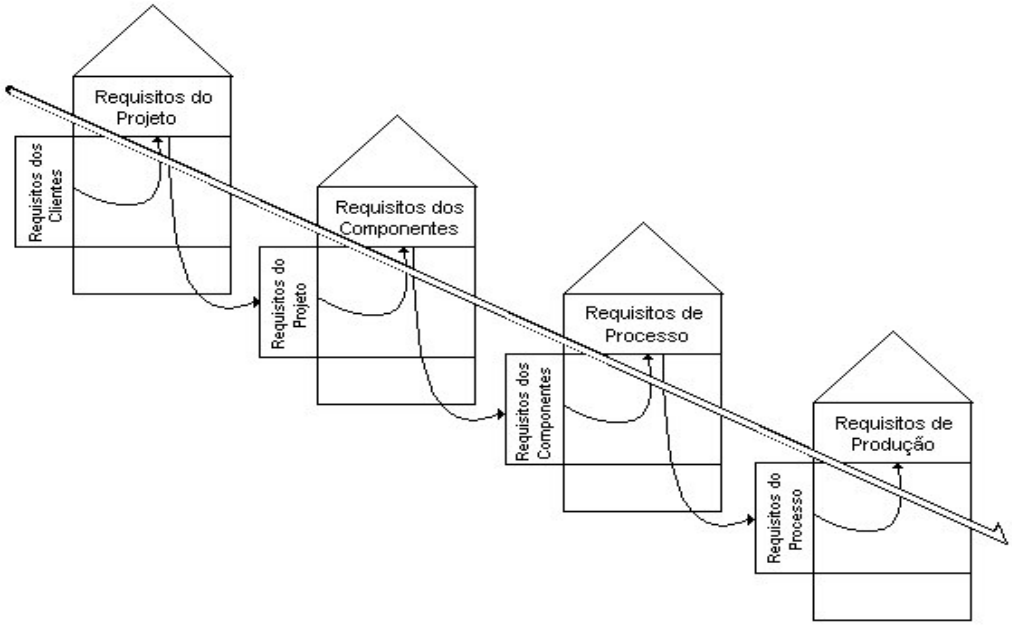


fig.2.5.3.2.a —Modelo conceitual (fases) do QFD

2.5.3.3 — Primeira matriz — Requisitos do cliente x requisitos do projeto

A primeira das quatro matrizes do QFD é também conhecida como "a casa da qualidade", por sua forma bastante peculiar (***fig.2.5.3.3.a***). O objetivo dessa matriz é identificação das características do artigo na forma de funções (finalidades), oriundas dos clientes e usuários e a tradução dessas funções em termos de descritores técnicos e suas medidas. Para dar-se início a essa fase, faz-se necessário que a voz do cliente tenha sido ouvida, seus problemas identificados e definidos e suas hipóteses de solução testadas (hipóteses essas que geraram as funções identificadas). Assim, e para que

surta o efeito desejado, o número de opiniões deverá ser representativo do universo de clientes que deseja-se atingir, uma vez que requisitos irrelevantes, mal descritos ou ausentes nessa fase representarão para as próximas fases o mesmo que se estar no andar certo do prédio errado.

No caso de desenvolvimento de um artigo para um cliente específico, que também utilize o QFD como metodologia de apoio para o planejamento da qualidade na sua empresa, os requisitos de produção no QFD desse cliente (quarta matriz, a ser vista mais adiante) poderão servir como entrada para essa primeira matriz do QFD do fornecedor. Em outros casos, o ideal seria que as metas do artigo (redução de custo, aumento na participação no mercado, aumento de produtividade, etc.) tivessem origem no plano estratégico da empresa, ou num processo de análise de requisitos (que é o caso mérito desse trabalho).

Essa fase divide-se em, basicamente, 8 passos:

a) identificação dos requisitos dos clientes;

b) comparação com a concorrência;

c) estabelecimentos dos objetivos em relação à concorrência;

d) tradução (extração) dos requisitos de projeto e identificação dos valores objetivados;

e) correlação dos requisitos do cliente com os de projeto, estabelecendo suas importâncias relativas;

f) correlação dos requisitos técnicos entre si. Esse é um aspecto interessante, uma vez que, como nas correlações entre os requisitos podem-se identificar ambigüidades, o método prevê que as ambigüidades possam ser repassadas para a fase anterior, caso não se encontre uma forma aceitável de contorno, para que o cliente (ou fase anterior) opte por qual requisito deseja sacrificar, se for o caso;

g) comparação dos requisitos técnicos com a concorrência e

h) estabelecimento das exigências para os requisitos técnicos.

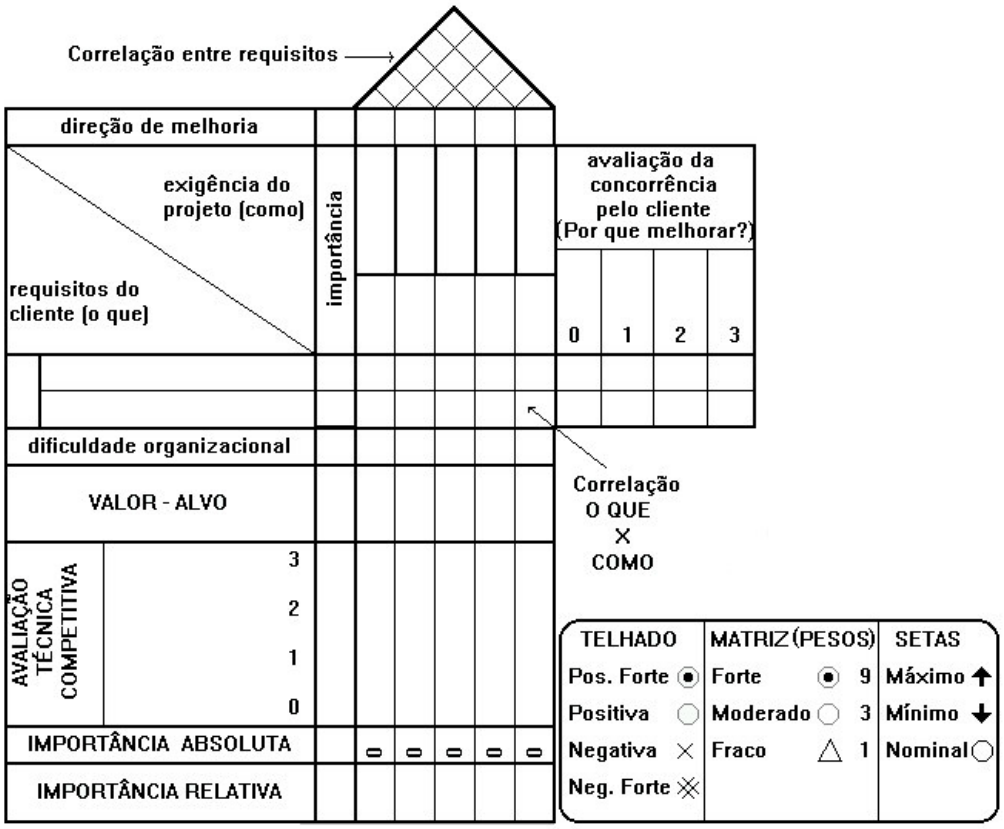


fig.2.5.3.3.a — Primeira matriz do QFD — a casa da qualidade

2.5.3.4 — Segunda matriz — Requisito de projeto x requisito de componente

Essa fase caracteriza-se pela relação estabelecida entre os requisitos de projeto identificados na primeira fase com os componentes do artigo a ser desenvolvido (no caso de software: objetos, tabelas, formulários, etc.). Quando o artigo já existe, o processo é puramente de correlação com as partes afetadas, caso contrário o processo é de extração, a exemplo da primeira fase. As exigências de projeto mais significativas são documentadas nas linhas (oriundas da fase anterior) e os componentes são documentados nas colunas (vide ***fig.3.5.3.4.a***).

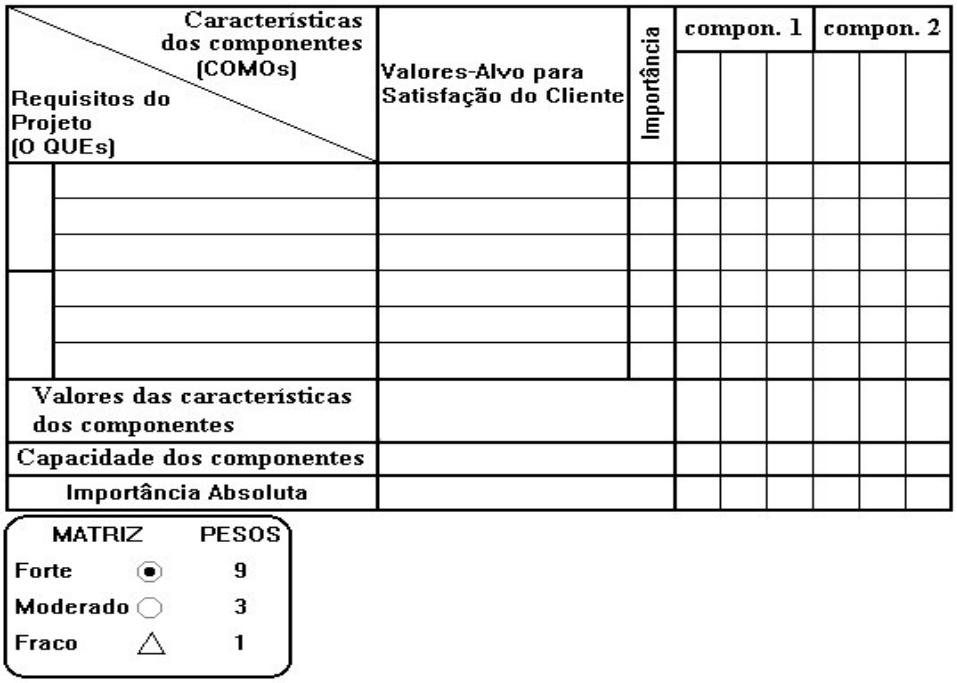


fig.2.5.3.4.a — Segunda matriz do QFD — requisito de projeto X requisito de componente

2.5.3.5 — Terceira matriz — Requisitos de componentes X requisitos dos processos

A próxima fase consiste na identificação de como proceder para a obtenção dos componentes, de tal forma que esses componentes consigam atingir as exigências do projeto, quer dizer, é a fase da definição dos processos que geram os componentes (no caso de software: procedimentos de implementação, revisão, métodos de ensaio, etc.) e os indicadores de qualidade e produtividade após cada atividade ou conjunto de atividades (quando não for possível o estabelecimento dos indicadores após cada atividade individualmente) que compõem os processos. Os processos afetados não são somente os da produção do artigo. Leva-se em conta, também, os processos de apoio (como compras e recepção de materiais, por exemplo) e de manutenção (como a ferramentaria, por exemplo). O modelo da matriz pode ser observado na ***fig.2.5.3.5.a.***

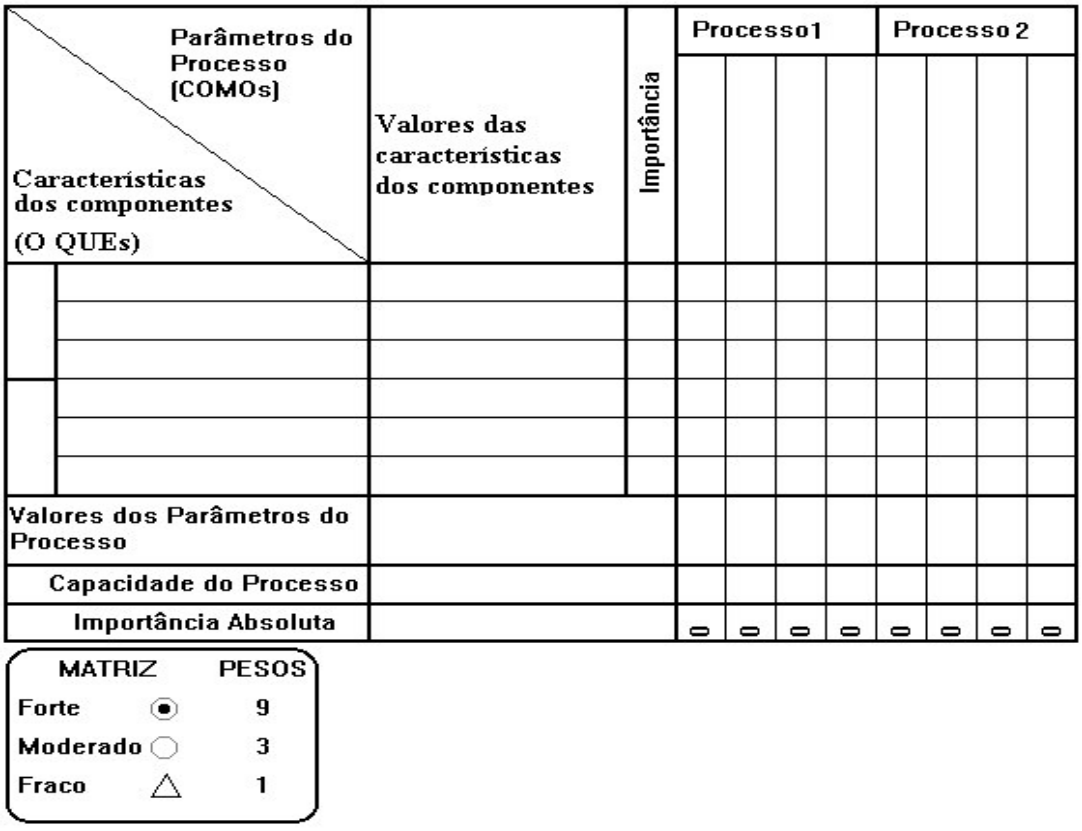


fig.2.5.3.5.a — Terceira matriz do QFD — requisito de componente X requisito de processo

2.5.3.6 — Quarta matriz — Requisitos de processos X requisitos de produção

Se a fase anterior definiu quais processos serão afetados, nessa fase deverão ser identificados os requisitos necessários para que os processos possam ser executados com sucesso (vide ***fig.2.5.3.6.a***). Por exemplo:

a) requisitos de habilidades necessárias (treinamento de mão-de-obra);

b) requisitos de recursos (dispositivos, ferramentas, utensílios, equipamentos, etc.);

c) requisitos de materiais que se consomem no processo (epecificações de marca, modelo, fornecedor, dimensões, etc.).

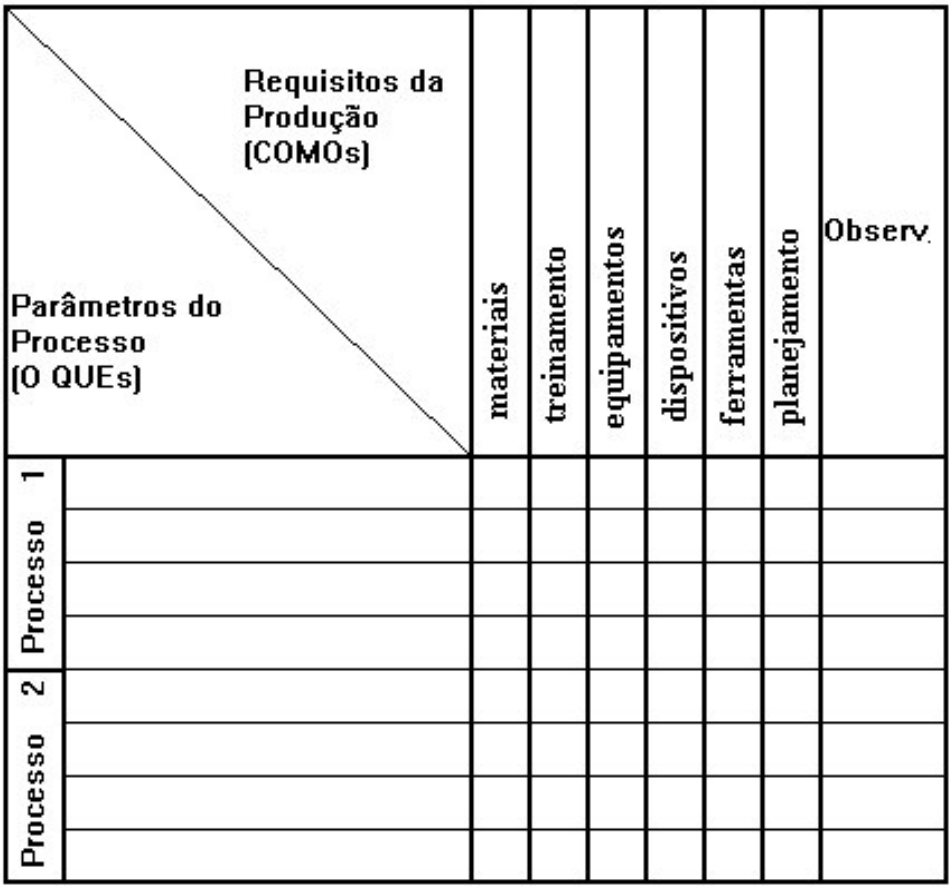


fig.2.5.3.6.a — Quarta matriz do QFD — requisito de processo X requisito de produção

2.5.4 — FOLHA DE VERIFICAÇÃO (OU CHECAGEM)

Trata-se de uma planilha através da qual podem ser documentados os dados identificados nos levantamentos de determinadas características de qualidade, sobre as quais deseja-se manter controle. Conforme [Paladini,1994,p.70], "*Não existe um modelo geral para as folhas de checagem — elas dependem de cada aplicação feita*". De qualquer forma, elas normalmente apresentam:

a) uma data ou período em que foi feito o levantamento dos dados;

b) artigo que está sob análise;

c) tipo de problema que está ocorrendo e

d) a freqüência com que o problema ocorreu, no período especificado.

A grande vantagem da folha de verificação é que ela permite uma identificação imediata dos problemas que ocorrem com maior freqüência num determinado artigo, dispensando a aplicação do gráfico de Pareto, para as situações em que as causas não necessitem ser traduzidas para uma outra unidade de medida que não seja a própria freqüência em que ocorrem.

2.6 — CONSIDERAÇÕES FINAIS SOBRE ESSE CAPÍTULO

Tendo indentificados os conceitos relacionados à qualidade (gerais e específicos da área de software), algumas ferramentas para o planejamento e o controle da qualidade e, anteriormente, os conceitos sobre sistemas de informações gerenciais, resta a necessidade de localizar-se o mérito desse trabalho, que é a análise de requisitos para sistemas de informações gerenciais. Isso será feito a seguir para, adiante, apresentar-se uma forma de combinar-se o conjunto dos conceitos dessas três disciplinas, como base da nova proposta.

CAPÍTULO 3 — ANÁLISE DE REQUISITOS DE SOFTWARE

O objetivo desse capítulo é permitir a localização dos conceitos de análise de requisitos de software e, principalmente, sobre análise de requisitos de softwares que prestam-se como apoio aos sistemas de informações gerenciais.

3.1 — UM BREVE HISTÓRICO SOBRE DESENVOLVIMENTO DE SOFTWARES

Desde que o computador foi formalmente reconhecido como útil para o processamento de dados em organizações sociais, até meados da década de 70, o fator mais crítico era a sua baixa capacidade tanto na armazenagem quanto no processamento desses dados. A palavra de ordem entre os desenvolvedores de software era "eficiência dos programas" para que os resultados pudessem ser obtidos em tempo hábil. Havia na época uma dependência muito forte das organizações aos indivíduos que desenvolviam o software, uma vez que, por causa das artimanhas de programação (muitas vezes necessárias por causa da pouca disponibilidade de “memória” dos computadores), somente o próprio autor de um programa seria capaz de manuteni-lo e muitas vezes nem ele próprio.

Em meados da década de 60 (1966 com Wirth, Hoare, Böhm e Jacopini) o mundo da informática começava a viver uma de suas fases mais ricas em termos de produção de idéias para a formalização do desenvolvimento de software. O problema de hardware ainda permanecia, mas já vislumbrava-se um horizonte bastante promissor nessa área e muitos pensadores do meio acadêmico começaram a estabelecer uma nova ordem para o desenvolvimento de sistemas para computador. Nessa época começaram a ser lançados os fundamentos da programação estruturada que passaria a basear-se em três estruturas básicas de controle: a seqüência, a seleção e a repetição (inicialmente foram somente duas essas estruturas básicas). Em 1968 Edsger W. Dijkstra publicou um artigo intitulado "Go To Statement Considered Harmful" onde ele

procurou argumentar sobre o uso indevido do comando "Go To" e as suas implicações. Esse artigo (publicado na edição de março de 1968 do periódico "Communications of the ACM") acabou por traduzir-se como a própria definição da programação estruturada no meio vulgar. Programação estruturada passou a ser encarada, de certa forma, como a programação sem o uso do comando "Go To".

A partir da década de 70 a palavra "estruturação" começou a ser empregada, difundida e expandida a outras fases do desenvolvimento de sistemas, não permanecendo, portanto, restrita somente à programação. As diversas fases do desenvolvimento de sistemas começaram a ser observadas do ponto de vista das estruturas de controle. Em 1972 D. L. Parnas publicou um artigo (Communications of the ACM) intitulado "On the Criteria to Be Used in Decomposing Systems into Modules" (sobre os critérios a serem usados na decomposição de sistemas em módulos). Esse artigo é uma evidência de que uma nova forma de se encarar os sistemas estava sendo empregada, a partir da formalização de critérios para a sua decomposição. Esses critérios foram utilizados dois anos depois (em 1974) por Stevens, Myers e Constantine numa publicação do "IBM Systems Journal" intitulada "Structured Design" (projeto estruturado). A partir daí os projetos de programas ganharam novos componentes e a estrutura básica de controle entre os módulos passou a ser a hierárquica. O objetivo final era a obtenção de um conjunto de unidades básicas, chamadas de “módulos funcionalmente coesos”. A expressão “estruturado” passou a fazer parte, agora, também da etapa de projeto de um sistema. Conforme [Stevens,1988], dois benefícios principais eram esperados através desse modelo: a capacidade de manutenção tanto para implementar novas exigências quanto para recuperar um erro e a reutilização de módulos, do qual derivaria o benefício da redução no tempo para o desenvolvimento de programas futuros. Nas palavras de [Stevens,1988,p.223]: "*Não há técnica de desenvolvimento que habilite uma pessoa a escrever e depurar um código mais rápido do que obter um código já existente*". Serão feitas algumas considerações sobre esse comentário mais adiante, ainda nessa seção.

Em 1979 Tom DeMarco escreveu um livro "Structured Analysis and System Specification" (análise estruturada e especificação de sistemas) onde são formalizados alguns princípios e estabelecidas algumas técnicas que introduzem a atividade de análise de sistemas na "era estruturada". Nessa mesma época outros estudiosos do

assunto, como Gane e Sarson, desenvolveram pesquisas e chegaram a conclusões que mais tarde se somariam às de Tom DeMarco. O modelo que caracterizava, por excelência, a análise estruturada era o “Diagrama de Fluxo de Dados” (DFD). Esse diagrama passou a representar em forma de rede os processos de um sistema e o fluxo dos dados dos processos entre si e com as entidades externas ao sistema.

O próprio gerenciamento de um projeto foi incluído entre as “coisas” estruturadas, em 1988 por Edward Yourdon, que escreveu um livro intitulado “Managing de System Life Cycle” (administrando o ciclo de vida do sistema). Nesse âmbito (do gerenciamento de projetos), estruturação passou a designar o paralelismo das atividades de planejamento e de controle dos projetos.

Paralelamente a essa "revolução estruturada" (ou revolução da decomposição funcional) estavam sendo desenvolvidas pesquisas por uma outra dimensão dos sistemas para computador. Pensadores como Hoare, Codd, Bachman e Chen defendiam a idéia de que a saída para a melhoria na qualidade dos sistemas começava com a modelagem da informação. Esse processo (conhecido em alguns meios como Engenharia da Informação) teve um impulso bastante significativo na década de 70 com a formalização na IBM de uma linguagem conhecida como SQL (Structured Query Language — linguagem estruturada, para consulta) para manipulação de bancos de dados e em 1976 com a publicação por Peter Chen de um artigo intitulado "The Entity-Relationship Model, Toward a Unified View of Data" (o modelo entidade-relacionamento, rumo a uma visão unificada dos dados — Communications of the ACM), formalizando um modelo que representa a “malha de memória” do sistema.

Havia, ainda, uma terceira espécie de sistemas, cuja implementação era derivada de um modelo de análise diferente dos modelos funcional e da informação: tratavam-se dos softwares baseados nos modelos de estado orientados para circuitos digitais (conforme [DeMarco,1989]). O método de estados de dispositivos e transição entre estados, ganhou impulso entre as organizações que desenvolvem os chamados “softwares embarcados” em máquinas, elevadores, automóveis, aeronaves, etc., e são conhecidos, também, como “sistemas em tempo real” (sistemas operacionais e para controle de dispositivos).

A diferença fundamental entre os três modelos (o funcional, o da informação e

o de estados) está na perspectiva inicial da análise e do projeto:

a) funcional: parte da decomposição dos processos (funções) da organização em estudo, para daí serem derivadas as estruturas de dados necessárias para o estabelecimento do sincronismo entre essas funções;

b) da informação: inicia o processo de análise identificando-se o modelo conceitual de dados (modelo que representa a visão dos usuários sobre as informações necessárias), que é decomposto nos conjuntos de dados do software (utilizando-se uma técnica conhecida como “normalização”) e, a partir desses conjuntos de dados, são implementadas as operações que aplicam-se sobre eles;

c) de estados: inicia o processo de análise na identificação dos possíveis estados que um determinado dispositivo pode assumir (aberto, fechado, ligado, etc.) para, daí, serem obtidos os processos que levariam esses dispositivos de um estado para outro e os eventos que fariam com que esses processos fossem disparados.

Em 1982, Tom DeMarco juntou essas três visões sobre sistemas. A idéia era a de representar os sistemas em perspectivas (ou em três dimensões) e isso trouxe um novo fôlego às técnicas e ferramentas criadas até então. Ao invés dos defensores de cada idéia competirem entre si, teve início um movimento de coesão de conhecimentos, que serve como base às ferramentas e metodologias atuais. Durante a década de 80 e no início da década de 90, esses conceitos foram ampliados (vide, por exemplo, [McMenamim,1984], [Ward,1987] e [Yourdon,1990]). Entretanto as dificuldades na manutenção dos modelos que eram gerados nas diversas fases do desenvolvimento e a falta de cultura com os modelos assíncronos malogrou a sua popularização, principalmente nas atividades de análise e projeto do sistema pela perspectiva das funções. Se o objetivo era o reaproveitamento máximo de funções já desenvolvidas (vide o comentário de [Stevens,1988] sobre “*obter um código já existente*”, anteriormente), isso, apesar de teoricamente fundamentado, não mostrou-se muito produtivo, até mesmo quando da aplicação desses modelos em projetos considerados pequenos (com três ou quatro centenas de pontos de funções) pela dificuldade apresentada na localização de uma função específica (estima-se que os softwares de

gestão empresarial de mercado tenham entre 50.000 e 150.000 pontos de função). Assim, uma forma mais eficiente de estruturação dessas funções mostrou-se necessária. Atualmente, essa forma tem sido apresentada como um conjunto de técnicas e disciplinas denominadas de “orientação por objetos”.

Conforme [Coad,1992,p.4], “a programação baseada em objetos foi discutida pela primeira vez no final dos anos sessenta por aqueles que trabalhavam com a linguagem SIMULA. Nos anos setenta, ela era uma parte importante da linguagem Smaltalk desenvolvida na Xerox PARC”. O conjunto total das técnicas orientadas para objetos tinham por objetivo, inicialmente, “auxiliar no gerenciamento da complexidade de grandes softwares de tempo real ” [Booch,1986].

No início da década de 80, as previsões sobre as técnicas orientadas para objetos eram de que elas estariam para os anos 80 o que a programação estruturada representou para a década de 70 (T. Rensch 1982, apud [Booch,1986]) (“an passant”, a diferença fundamental entre as técnicas estruturadas e as orientadas para objetos está na forma de decomposição das operações a serem executadas pelo software. Uma fundamenta-se na decomposição das funções dentro de uma unidade estruturada hierarquicamente, que é o programa, e outra baseia-se na distribuição das funções por objetos e classes de objetos, sobre os quais essas funções podem ser aplicadas). Isso de fato foi assim, mas não somente a programação orientada para objetos seguiu os passos da programação estruturada, do ponto de vista das semelhanças nas pesquisas sobre técnicas e na produção de software. Da mesma forma como a “estruturação” transformou-se num paradigma, afetando os processos de projeto, análise e gerenciamento do desenvolvimento de software, os “objetos” têm se transformado num paradigma desses mesmos processos. Entretanto, a aplicação desses princípios como uma panacéia, para todos os males relativos ao desenvolvimento de software (assim como foi a “estruturação”), independentemente do alerta de [Booch,1986]: “*devemos chamar atenção sobre o fato de que o desenvolvimento orientado para objetos é um método que atinge o ciclo de vida* ***apenas parcialmente****. Ele está focado sobre os estágios de* ***projeto e implementação****, no desenvolvimento de software ... é imprescindível o acoplamento de* ***métodos apropriados de requisitos e análise****, ao desenvolvimento orientado para objetos*” (grifos acrescentados), pode desviar ainda mais a atenção dos que buscam uma solução para o controle de projetos de software, a

uma direção que pode não ser a mais adequada, hajam vistas, por exemplo, a um relatório da KPMG, divulgado em abril de 1997 (apud [Poulin,1998]). Nesse relatório dá-se contas de que:

a) 87% dos projetos de software analisados sofreram desvios de cronograma;

b) 56% dos projetos excederam o custo estimado e

c) em 45% dos projetos foi constatado que o resultado dos produtos de software eram inadequados às necessidades de negócio a que propunham-se apoiar.

Assim, se por um lado a evolução do conjunto de todas as técnicas permitiu uma melhoria na qualidade intrínseca dos produtos de software (por causa da evolução das técnicas e ferramentas de projeto e programação), por outro os problemas relacionados ao controle sobre os projetos de software e à aplicação desses softwares permaneceu.

Considerando-se estritamente os problemas relacionados à adequação do software às necessidades dos usuários, duas hipóteses poderiam ser estabelecidas: ou os requisitos dos usuários não são completa e adequadamente coletados, ou esses requisitos são distorcidos ao longo do ciclo de desenvolvimento do software. Para que isso seja melhor compreendido, será apresentado, a seguir, o ciclo de vida de desenvolvimento de um software, seguido das questões relativas à análise dos requisitos de software propriamente.

3.2 — CICLO DE VIDA DO DESENVOLVIMENTO DE SOFTWARE

Na ***fig.3.2.a*** é representado um exemplo de ciclo de vida para desenvolvimento de um software. Como pode ser observado, os seguintes processos fazem parte desse ciclo:

a) análise de requisitos de mercado: responsável pela identificação de tendências tecnológicas que, se não percebidas em tempo, podem fazer com que uma instalação de software fique com sua capacidade de migração deficiente, podendo forçar a empresa a recomeçar o processo de

informatização, quando poderia fazer um reaproveitamento do que já possui;

b) análise dos requisitos de usuários: responsável pela identificação dos problemas dos usuários. Esse processo é o mérito desse trabalho e será visto em detalhes, mais adiante;

c) projeto do software: a partir dos requisitos técnicos, requisitos de mercado e normas técnicas, é o processo responsável pela idealização da solução em termos de: projeto dos programas, projeto de banco de dados, projeto de ferramentas de produção, processos de produção dos programas e processos necessários à implantação do software;

d) ferramentaria: responsável pela implementação das ferramentas idealizadas pela fase anterior (projeto do software). Esses dispositivos podem ter vida além de um projeto especificamente, ou podem aplicar-se a uma situação específica do projeto em questão;

e) produção do software: é o processo responsável pela implementação dos elementos que compõem o software, a saber: os programas, manuais de operação, etc.;

f) implantação do software: responsável pela instalação do software, conversão das bases de dados do software antigo para o novo (quando houver), treinamento dos usuários e definição dos procedimentos de utilização.

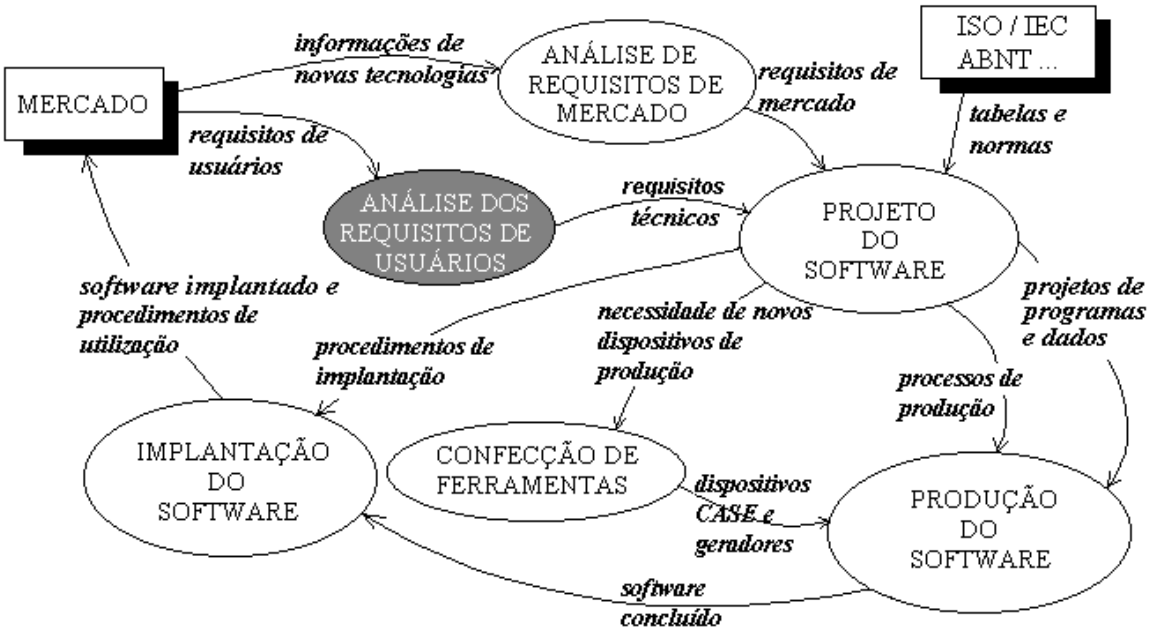


fig.3.2.a — modelo de ciclo de vida para o desenvolvimento de software

3.3 — COMPREENDENDO OS REQUISITOS DOS USUÁRIOS DE SOFTWARE

Afinal, o que são requisitos de software? — Segundo [Dean,1994], "*é qualquer coisa que restringe o sistema*" e, conforme uma publicação do Software Productivity Consortium Inc. (S.P.C.I.), "*Os requisitos definem o problema. Eles lhe dizem o que o software deverá fazer. Os demais passos do processo tradicional de desenvolvimento de software criam a solução"* [SPCI,1996].

E qual seria o objetivo da análise de requisitos de software? — Conforme Leite (1987, apud [Zirbes,1996]) *"a Análise de Requisitos é um processo, onde o que deve ser feito é eliciado* (expulsar, fazer sair) *e modelado. Este processo envolve-se com diferentes visões e utiliza uma combinação de métodos, ferramentas e atores. O produto deste processo é um modelo, a partir do qual um documento denominado Requisitos do Sistema é produzido*", para [Thayer,1996] engenharia de requisitos de software é a disciplina usada para capturar correta e completamente os requisitos de software e expectativas dos usuários do software e as técnicas e disciplinas da engenharia de requisitos de software têm como objetivo a elicitação de requisitos do macrossistema,

para [Breitman,1998].

É importante salientar-se, para que não haja confusão de terminologias, que na bibliografia pesquisada, os termos “engenharia de requisitos” e “análise de requisitos” foram encontrados referindo-se ao mesmo conjunto geral de atividades e objetivos, indistintamente (isso pode ser verificado observando-se as referências citadas acima), ou seja, os objetivos entre as atividades da “engenharia de requisitos” e as da “análise de requisitos” são semelhantes. Para os efeitos desse trabalho optou-se pelo termo “análise de requisitos”.

Assim, do que trata-se em análise de requisitos de software de informações? — Precisamente da identificação das necessidades dos usuários de informações e comunicação dessas necessidades aos processos de construção do software (projeto e produção).

3.3.1 — CONSEQÜÊNCIAS DE ERROS NA FASE DE ANÁLISE DOS REQUISITOS DE SOFTWARE

Como pode ser observado na ***fig.3.2.a***, o processo de “Análise dos Requisitos de Usuários” é responsável pela identificação dos objetivos a serem atingidos pelo software. Os erros cometidos nessa fase serão transmitidos para as fases de projeto e produção do software na forma de requisitos inconsistentes. A estrutura do software, que é determinada na fase de projeto, pode ficar comprometida na medida em que os requisitos técnicos não forem consistentes. Para [Bender,1997], definir os requisitos é o primeiro, e mais crítico, passo em desenvolvimento de sistemas de software. Se a definição dos requisitos é pobre, o projeto resultante é desajeitado.

Segundo o S.P.C.I., “o impacto, se você não identificar um erro na fase de análise de requisitos e projeto do software, será de um custo 4x maior para eliminar esse erro na fase de testes e 100x maior para eliminá-lo na fase de manutenção do software” (vide **fig.3.3.1.a**) e “de fato, requisitos de software são o passo mais importante no processo ... e são conhecidos como o **principal problema** da indústria de software" [SPCI,1996]. [Thayer,1996] afirma que as especificações de requisitos incorretas ou incompletas, são a maior causa de erros de software.

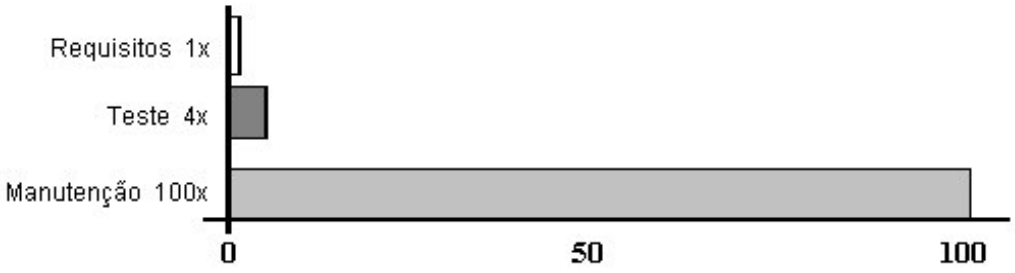


fig.3.3.1.a — custo relativo para eliminar problemas (conforme a etapa)

Uma expressão de John Von Neumann, apud [Gause,1991,p.249], pode ser perfeitamente aplicada nesse ponto: "*não faz sentido ser exato em relação a algo se você nem mesmo sabe o contexto do que está falando*", resumindo a incoerência das soluções perfeitas para problemas irreais.

3.3.2 — HISTÓRICO DE DESENVOLVIMENTO E ESTÁGIO ATUAL SOBRE REQUISITOS DE SOFTWARE

Em 1980, Edward Yourdon, prefaciando o livro "The Practical Guide to Structured Systems Design" de Meilir Page-Jones [Jones,1980], escreveu o seguinte:

> O projeto estruturado preocupa-se com a seleção e organização de módulos e suas interligações, de modo a conseguir a melhor solução para um problema bem formulado. Mas de onde vem essa boa formulação de problema? Como podemos ter certeza de que as necessidades do usuário foram apropriadamente especificadas? ... Infelizmente essas questões foram um tanto ignoradas ... Assim, a "ponte" entre a análise e o projeto tem sido motivo de confusão e trauma para muitos profissionais de processamento de dados.

Pelas evidências nesse texto, sabe-se que já em 1980 havia sido identificado o problema da qualificação dos requisitos de software, tanto na sua interpretação (*problema bem formulado*) quanto na sua comunicação para a fase de projeto do software (*ponte entre a análise e o projeto*). O mesmo Edward Yourdon escreveu em 1990, em parceria com Peter Coad, um dos principais “best-sellers” atuais sobre análise

de sistemas. Trata-se do livro “Object-Oriented Analysis”. Nesse livro eles reafirmam o problema de uma década anterior, dizendo: “*A transição da análise para o projeto tem sido uma fonte constante de frustração. Muitos trabalhos foram escritos; muito pouco progresso foi alcançado”* [Coad,1992,p.23]. A intenção desse novo livro era a de resolver o problema e o desfecho parecia resumir-se a uma simples questão de uniformização na representação dos resultados das modelagens do problema (análise) e da solução (projeto). Como foi afirmado “*Tudo é uma questão de usar a mesma representação básica na análise e no projeto”* [Coad,1992,p.23]. As evidências, entretanto, indicam que esse problema permanece ainda hoje. Essas evidências são tanto pelas pesquisas que são desenvolvidas sobre o assunto, por exemplo:

a) departamento de engenharia de software da Universidade de Stuttgart, na Alemanha, declara que engenharia de requisitos de software está entre os seus mais importantes assuntos de pesquisa [Stuttgart,1997];

b) no Brasil, pesquisadores da COPPE/UFRJ, por exemplo, estão testando a aplicação da teoria de Fuzzy para esse fim [Belchior,1996],

quanto pelos resultados das estatísticas que são realizadas sobre o assunto. Além do resultado apresentado anteriormente, do relatório da KPMG, pode ser citado como outro exemplo, o resultado de uma pesquisa de Alan Daves (apud [Rocha,1998]), onde é constatado que: 56% de todos os erros encontrados nos softwares são originados na fase de análise de requisitos e 75% desses erros são detectados somente depois das etapas de implementação e teste, ou seja, 42% dos erros são percebidos somente pelos usuários, por ocasião da implantação ou uso do software (só para relembrar, a pesquisa da KPMG chegou a um percentual de 45% de inadequação no uso). Segundo essa pesquisa, os erros de requisitos são distribuídos da seguinte forma:

a) 49% devido a fatos incorretos;

b) 31% omissões;

c) 13% inconsistências;

d) 5% ambigüidades e

e) 2% localização errada do requisito.

3.3.2.1 — Qual é o "estado da prática", no Brasil?

Em meados de 1996, o Ministério da Ciência e da Tecnologia (MCT) publicou uma pesquisa intitulada "Qualidade no Setor de Software Brasileiro - 1995" [MCT,1996]. Entre as questões analisadas na pesquisa, uma versava sobre técnicas de engenharia de software adotada e, entre as técnicas, uma referia-se à análise de requisitos. Das empresas que responderam ao questionário, apenas 47,4% disseram fazer uso dessas técnicas.

Em agosto de 1998 foi divulgado um novo relatório, dessa vez de uma pesquisa referente ao ano de 1997, nos mesmos moldes da pesquisa feita em 1995. Nessa nova pesquisa, apenas 35,5% disseram fazer uso das técnicas de análise de requisitos [MCT,1998]. Presumindo-se que as respostas ao questionário do MCT foram totalmente fiéis e levando-se em conta a margem de erro[ii] , cerca de 37% das empresas brasileiras produtoras de software, em 1997, ainda não utilizavam um método formal de análise de requisitos e, por conseqüência, também não utilizavam nenhum método formal para a transição entre o processo de análise dos requisitos e o processo de projeto do software.

3.4 — ALGUNS MÉTODOS POPULARES PARA ANÁLISE DE REQUISITOS

Serão apresentados, a seguir, alguns dos principais modelos de análise de requisitos para sistemas de informações, que têm sido apresentados ao mundo, pelas últimas três décadas.

[ii] De um total estimado de 2.500 empresas que atuam no setor de software brasileiro, com atividade de desenvolvimento, 589 responderam ao questionário em 1997. Segundo os relatórios (tanto o de 1996 quanto o de 1998), a amostra tem margens de erro máxima de 5%.

3.4.1 — ANÁLISE ESTRUTURADA DE SISTEMAS

O modelo de análise estruturada foi formalizado na década de 70 e popularizado principalmente por [GANE,1983] e [DeMARCO,1989]. Fundamentada no princípio da decomposição funcional, ele tem como objetivo a modelagem da organização em estudo utilizando-se, para esse fim, de um conjunto de ferramentas, sendo o Diagrama de Fluxo de Dados (DFD) e o Dicionário de Dados (DD) as principais delas.

O método de decomposição recomendado, conforme [DeMARCO,1989], é baseado na busca pela menor taxa de transferência de dados entre os sub-processos dos processos que estão sendo analisados, o que levaria, por conseqüência, a um modelo da organização decomposta em módulos funcionalmente coesos, ou seja, módulos que têm uma única função.

A principal crítica à análise estruturada é a não replicabilidade dos requisitos resultantes da análise, caso fosse feito um experimento em que diversos analistas fossem incumbidos de analisarem o mesmo sistema. Isso porque o processo de decomposição utilizado depende da abordagem de cada analista particularmente (dos seus pressupostos e de suas experiências anteriores ao projeto em questão) e das coincidências de informações oriundas das pessoas que compõem a organização social sob análise.

3.4.2 — ANÁLISE ESSENCIAL DE SISTEMAS

Definida em 1984 por Stephen M. McMenamim e John F. Palmer ([McMenamim,1984]) a abordagem da análise essencial de sistemas utiliza-se das mesmas ferramentas de modelagem da análise estruturada, mas os mecanismos são diferentes. Ao invés de uma decomposição do mais geral para o mais específico (“top-down”) o método prevê que sejam identificados, inicialmente, os eventos externos aos quais espera-se que a organização social em questão responda, sendo derivadas então as ações (ou funcões) em resposta a esses eventos e, posteriormente, os eventos gerados internamente e também as respectivas ações. A expressão “essencial” deve-se

ao fato de que não serão consideradas quaisquer restrições tecnológicas, ou seja, como se a tecnologia disponível fosse perfeita o suficiente para suportar quaisquer questões relacionadas à captura ou mixagem dos dados, permitindo que seja possível concentrar-se somente sobre as questões essenciais do sistema sob análise.

3.4.3 — ENGENHARIA DA INFORMAÇÃO

Os conceitos fundamentais da engenharia da Informação foram estabelecidos em 1981 por James Martin e Clive Finkelstein e, para Martin (apud [Neto,1988,p.2]), "*A Engenharia da Informação é um conjunto integrado de técnicas formais pelas quais modelos de empresa, modelos de dados e modelos de processos são construídos a partir de uma base de conhecimentos de grande alcance, para criar e manter sistemas de processamento de dados*".

O modelo da engenharia da informação é mais abrangente que os de análise estruturada e essencial, apresentados anteriormente, abrangendo, além das atividades de análise, as atividades de projeto e implementação do software. A análise propriamente está presente em duas das fases desse modelo: a fase do planejamento estratégico da informação e a fase da análise das áreas de negócios da empresa.

O planejamento estratégico da informação é responsável pela identificação das informações necessárias à consecução dos objetivos estabelecidos no planejamento estratégico empresarial. Nesse sentido, a engenharia da informação leva vantagens, quando comparado aos outros modelos de análise (como as estruturada e essencial), na medida em que possui como ponto de partida um conjunto bem definido de objetivos organizacionais, não limitando-se apenas à automação de operações burocráticas, que podem nem estar contribuindo para os objetivos da empresa. Também é responsável pelo desenvolvimento do modelo corporativo de dados (que, mixados, geram as informações táticas e estratégicas) e do modelo funcional da organização sob análise.

A análise das áreas de negócios refere-se à decomposição do modelo corporativo de dados em dados operacionais e à decomposição funcional dos setores da organização que respondem diretamente pelos resultados esperados e que geram os

dados necessários à estratégia empresarial estabelecida. Aparte do quesito da decomposição do modelo de dados corporativos, o modelo a ser adotado para a decomposição funcional não é definido de forma diferenciada das técnicas existentes, podendo optar-se por uma técnica como a análise estruturada, por exemplo. Nesse sentido, a engenharia da informação torna-se como que um arcabouço, dentro do qual os modelos de análise funcional encontram uma forma mais definida (por causa das restrições estratégicas) de aplicação, tornando os seus resultados menos aleatórios.

3.4.4 — ANÁLISE DE REQUISITOS BASEADA EM PROTÓTIPOS

Existem métodos de requisitos conhecidos como “métodos de protótipos”, que tentam cumprir o papel de antecipar a evidenciação de erros de requisitos, para antes da implementação do software numa linguagem alvo (conforme [Gane,1988]).

Segundo [Yourdon,1989], esses métodos podem ser aplicados a projetos considerados “bons candidatos”, tendo como característica principal o fato de resultarem em softwares essencialmente “on-line” (em-linha), quer-se dizer, softwares que não têm processamentos em lote, como cálculos de necessidades de materiais, atualizações integradas de recepção de materiais, etc.. Isso ocorre porque esses métodos conseguem representar apenas a parte do software com a qual o usuário terá contato direto e isso pode garantir a qualidade do ponto de vista da adequação (caso a hipótese do cliente venha a confirmar-se), mas não da funcionalidade.

A principal crítica aos métodos de protótipo, quando foram criados, era o fato de que os requisitos dos clientes não eram documentados e, pelo fato de as linguagens de prototipação serem apenas linguagens de simulação (não serem linguagens de programação propriamente), quando definitivamente chegava o momento do software ser programado, os requisitos haviam se degenerado, restando apenas o desenho de telas e relatórios. Esse aspecto desmotivou o uso de protótipos, no início. Entretanto, na primeira metade da década de 1990 começaram a ser implementadas ferramentas de desenvolvimento de software baseadas naquilo que comercialmente ficou conhecido como RAD (“Rapid Aplication Development” — Desenvolvimento Rápido de Aplicativos), buscando-se a redução dessa deficiência, uma vez que a linguagem de protótipos

passou a ser a própria linguagem de programação definitiva.

Atualmente a tecnologia RAD tem sido amplamente divulgada e aplicada (exemplos de linguagens que utilizam essa tecnologia são a “Delphi” e a “C++Builder”, ambas da empresa Inprise (antiga Borland)) e o que tem-se observado é que, para softwares com meia dúzia de funções, um modelo de análise de requisitos baseado em protótipos e em tecnologia RAD é útil, desde que o projeto seja, nas palavras de [Yourdon,1989], um “bom candidato”. Entretanto, se por um lado o modelo resultante desses métodos, suportados pela tecnologia RAD, elimina a necessidade de os requisitos serem formalmente documentados, foi criado um novo problema que é a dificuldade na manutenção do software. Conforme [Yourdon,1989,p.64]:

> Se o protótipo é na verdade jogado fora e substituído pelo sistema de produção, há um perigo real de que o protótipo possa acabar sem haver um registro permanente dos requisitos do usuário. Isso provavelmente tornará a manutenção e modificação cada vez mais difícil com o passar do tempo (dez anos após o sistema ser montado, será difícil para os programadores de manutenção incorporarem uma mudança, pois ninguém, incluindo os usuários de “segunda-geração” que agora trabalham com o sistema, se lembrará do que ele pretendia fazer a princípio).

3.4.5 — ANÁLISE ORIENTADA A OBJETOS

Um dos primeiros livros sobre análise orientada para objetos foi escrito por Sally Shlaer e Stephen J. Mellor, em 1988 (traduzido para o português em 1990, conforme [Shlaer,1990]) e trata-se de uma abordagem de análise “interessante” do ponto de vista científico, na medida em que por um lado não é uma técnica de análise, no sentido restrito da decomposição que busca aprender alguma coisa, e por outro lado vem transformando-se em fenômeno de um mercado ansioso por técnicas que lhe dêem uma solução na captura de requisitos, principalmente para o desenvolvimento de aplicações de gerenciamento e gestão. Sobre essas aplicações (como é o caso dos ERP’s) [Furlan,1998,p.4] fala que “*Esses softwares em si são um grande alvo da tecnologia de objetos uma vez que a modularização em nível de componentes em lugar da modularização em nível de subsistemas pode proporcionar um ajuste mais adequado na* ***implementação*** *do produto nas empresas clientes*”. O grifo em “implementação” foi

acrescentado para demonstrar que o texto deixa transparecer que esse autor concorda que as ferramentas da orientação a objetos aplicam-se às atividades de "implementação", ou seja, nas atividades que seguem o processo de análise de requisitos (projeto e produção do software), como na concepção original de [Booch,1986], apresentada anteriormente. Entretanto, a que deve-se a afirmação, na mesma publicação de [Furlan,1998,p.12], de que "*Um aspecto importante da teoria dos objetos é a sua característica intrínseca de* ***analisar*** *o mundo como ele é, permitindo organizar resultados de maneira mais fácil e natural*" (grifo acrescentado) e por que, para autores como Rumbaugh (apud [Furlan,1998,p.15]), a orientação a objetos é "*uma nova maneira de* ***pensar os problemas*** *utilizando modelos organizados a partir de conceitos do mundo real*"? (grifo acrescentado). Ao longo da presente pesquisa observou-se que o termo "análise de sistemas", em alguns casos (possivelmente aplicável a [Furlan,1998,p.12]) também representa as atividades relacionadas ao projeto do software. Entretanto, quando relacionado à orientação aos objetos, aparentemente, essa questão torna-se crônica. Observe-se o comentário de [Hay,1998]:

> As livrarias em todo o mundo estão cheias de livros concernentes à "análise orientada para objetos". Será possível que o mundo da "orientação para objetos" mudou completamente a natureza da análise de sistemas? Se dermos ouvidos aos seus aficcionados, isto deverá ser aceito como verdadeiro ... entretanto Grady Booch afirma que a análise deverá focar o ambiente, não a forma ... ele afirma que a análise orientada para objetos trata, na verdade, sobre projeto de sistemas.

E como explicar o comentário de [Furlan,1998,p.98]? Diz ele: "*A criação de um modelo de objetos é, freqüentemente, resultado da consolidação de diversos projetos de escopo funcional. Criar modelos é um trabalho* ***altamente criativo****, e* ***não há uma solução final ou uma resposta correta que possa ser verificada ao final do trabalho***" (os grifos foram acrescentados). Partindo-se do pressuposto desse ponto de vista, o presente trabalho não teria mais razão de ser. De qualquer forma, para que essa questão possa ser melhor compreendida, passemos a observá-la sob a luz de [Shlaer,1990,p.6]:

> Muitos projetos produzem, como seu primeiro produto importante, uma relação formal dos requerimentos do sistema. Os requerimentos são revistos então por

> um comitê de especialistas dos vários departamentos da organização. Todavia, devido ao fato de a relação dos requerimentos conter material de tantas áreas diferentes, cada membro do comitê se julga responsável pela qualidade de somente uma parte das especificações. A responsabilidade de entender a relação em seu todo e em profundidade, juntamente com todas as hipóteses e conceituação em que se baseia, tipicamente não cabe a ninguém. Como resultado, é quase inevitável que o documento dos requerimentos seja considerado mais tarde incompleto e/ou incongruente. É provável que, assim que estas falhas começarem a surgir, a credibilidade dos requerimentos fique tão prejudicada que o documento chegue a não ser levado em conta pelos que desenvolvem o sistema. O trabalho em que isto acontecer terá pouca serventia

e, mais adiante:

> Em nossa opinião, é conveniente iniciar o projeto de desenvolvimento do software com a construção de um **modelo de informação do problema da aplicação**. Depois disso, quando apropriado para o problema, podem ser aplicados os outros passos da análise (diagrama de fluxo de dados, diagramas de transição de estados, trabalhos de requerimentos formais e similares) *(grifo acrescentado).*

O que deve ser observado é que pressuposto fundamental da teoria de objetos, sobre a qual está fundamentado [Shlaer,1990] e [Furlan,1998], está em que, modeladas as informações dos objetos que fazem parte do ambiente do problema em estudo, todas as respostas, relacionadas ao problema, ficarão naturalmente disponíveis. Aparentemente isso faz sentido, entretanto não se pode deixar de dar atenção ao alerta de [Senge,1990,p.70]: "*... existe uma característica fundamental dos sistemas humanos complexos: causa e efeito não estão próximos no tempo e no espaço*". Diante disso, não é coerente presumir-se que é possível a modelagem das informações do ambiente do problema (onde normalmente os efeitos são evidentes), sem conhecer-se as causas do problema, causas essas que delimitam a verdadeira extensão do ambiente do problema, que podem ir além das fronteiras físicas da organização social sob análise. A questão das dificuldades com os documentos de requerimentos, apresentada em [Shlaer,1990] é bem real, entretanto a modelagem dos objetos do presumível ambiente do problema não

é suficiente. Para [Senge,1990,p.79] "*A verdadeira alavancagem na maioria dos problemas administrativos está em entender a complexidade de dinâmica, e não a complexidade de detalhes. Infelizmente as "análises de sistemas" enfocam quase sempre a complexidade de detalhes*", podendo essa complexidade de detalhes ser ilustrada, justamente, pelo resultado de um modelo de análise orientado para objetos: a decomposição da organização por objetos, ao invés da decomposição pelas relações dinâmicas, que ocorrem nos eventos entre os elementos dentro dessa organização e fora dela, com outras organizações com as quais ela relaciona-se.

Observando-se a comentário de [Furlan,1998, pag.7]: "A tarefa mais difícil é a de modelar os processos existentes sem a interferência das pessoas, organogramas e cultura estabelecida" tem-se uma noção do porquê que os analistas de sistemas têm tantas dificuldades para identificar o essencial no estudo para o desenvolvimento dos sistemas de informação: a ausência de foco no relacionamento, buscando estudar as organizações sociais "sem a interferência das pessoas ... e culturas estabelecidas", como se isso fosse possível, uma vez que os sistemas sociais não possuem uma anatomia, ou uma estrutura aparte de seu funcionamento, conforme foi visto anteriormente a partir de [Katz,1974], o que, naturalmente, pressupõe a presença das pessoas com suas culturas. O que não se pode fazer é esquecer-se que os modelos de objetos nasceram para a análise e implementação de sistemas embarcados, que controlam realmente objetos como máquinas de fotocópia, aviões, barbeadores, etc., e que possuem uma anatomia bem definida, independente da interferência de variáveis sociais.

3.5 — CONCLUSÃO DO CAPÍTULO

Nesse capítulo foram apresentados alguns modelos de análise de requisitos e os problemas que esses modelos têm enfrentado, tanto problemas objetivos (como as conseqüências de uma análise mal feita) quanto problemas inerentes aos modelos

propriamente. No próximo capítulo será apresentada uma proposta que busca atingir justamente esses dois focos (que complementam-se entre si): reduzir os problemas dos modelos atuais de análise e, por conseqüência, minimizar os problemas com sistemas de informações nas organizações sociais.

CAPÍTULO 4 — PROPOSTA DE PROCESSO PARA ANÁLISE DE REQUISITOS

Antes de encaminhar-se a explicitação de uma proposta de processo para análise de requisitos, será feita, a seguir, uma descrição mais detalhada dos problemas que a envolvem. Essa descrição será feita a partir da definição do consumidor objetivo, passando pela definição dos objetivos da empresa usuária de um sistema de informações gerenciais e culminando com o estudo daqueles que deveriam ser os objetivos do processo de análise. Identificadas algumas deficiências nesses processos, será apresentada a nova proposta.

4.1 — DEFINIÇÃO DO CONSUMIDOR OBJETIVO

Um aspecto particularmente complexo na indústria de software, principalmente em projetos corporativos onde os serviços de adaptação do software às necessidades específicas de cada cliente é comum, é o de que a busca pela universalização e massificação das vendas tem levado à interpretação dos problemas dos usuários segundo a conveniência dessa massificação, causando interferências no processo de análise. Conforme [Kaneko,1994], na indústria de serviços a variedade de requisitos dos diversos consumidores complica a padronização de cada tipo de serviço que deve-se proporcionar, uma vez que os resultados desses serviços raras vezes agradam a todos e, quando agradam a maioria, normalmente não proporcionam uma satisfação total. Diante dessa dificuldade, não é muito conveniente falar-se sobre qualidade sem que seja identificado o mercado alvo que deseja-se atingir e sem supor-se que o cliente potencial faça parte desse mercado. Para [Gause,1991], as pessoas assimilam particularmente o que lhes parece conhecido ou lhes interessa. Assim, é razoável supor-se que um projeto de adaptação de software seja influenciado pelas expectativas subjacentes do fornecedor que, na ausência de uma política adequada de qualidade, é levado a promover “arranjos” capazes de comprometer seriamente as hipóteses específicas de cada usuário, para que o software possa ser aplicado a uma gama maior

de clientes.

Qualquer processo de planejamento de qualidade de software torna-se temeroso diante desse cenário. Se não houverem acordos suficientes acerca das expectativas dos clientes e também dos fornecedores, aquilo que aparentemente seria uma coincidência de oportunidades acaba transformando-se em frustração. Mercados objetivos e acordos estabelecidos sobre as expectativas negociadas, permitem que sejam alcançadas vantagens consideradas suficientes, evitando que transformem-se em frustrações ao longo do processo de desenvolvimento ou na implantação do software. Por fim, isso reduz as possibilidades de que o documento de requisitos seja um simples registro de funções presumíveis que atendam às necessidades mais ou menos conhecidas e acordadas entre as partes. Esse é um pressuposto assumido como verdadeiro para que a metodologia que está sendo proposta não se torne inútil: que seja intenção do fornecedor a manutenção da voz do cliente.

Para que fique registrado, a norma ISO 9126 aborda a questão do contrato entre fornecedor e cliente de software, entretanto a qualidade sobre as questões contratuais ficam nos termos do acordo formalizado, não das expectativas de ambas as partes (vide, por exemplo, as alíneas “a.a”, “a.b”, e “b” da norma, sobre essa questão).

4.2 — OS REQUISITOS DO PONTO DE VISTA DA ORGANIZAÇÃO

Como visto anteriormente, as causas dos problemas num sistema social são geradas nos relacionamentos entre os elementos desse sistema social (tanto que o foco da teoria geral dos sistemas está no arranjo que conecta os componentes em um todo e não nos atributos constantes desses componentes, como já foi descrito). Entretanto, observando-se os modelos de análise de requisitos estudados no capítulo 3, constata-se que não são devidamente levados em conta os objetivos da organização sob análise, no que refere-se ao controle desses relacionamentos. Estudando-se os modelos propostos, eles aparentemente resumem-se:

a) à modelagem da organização, na forma como encontra-se ou na forma como espera-se que esteja no momento em que a implantação do software for concluída e

b) ao estabelecimento dos requisitos esperados do ***software*** em questão, como se esses requisitos fossem, necessariamente por si próprios, os requisitos da ***organização*** sob análise,

resultando, por fim, numa estatística em que, para aproximadamente 45% dos casos, os problemas que o software propõe-se a ajudar a resolver estão dissociados dos problemas reais dessa organização.

Assim, outro pressuposto básico para o modelo de análise a ser proposto é o de que não há informação útil se não houverem relacionamentos entre os elementos de um sistema. Elementos isolados podem até possuir atributos, mas esses atributos não têm importância fundamental, se os elementos estiverem dissociados uns dos outros. É sobre esses relacionamentos que a organização precisa de informações e esses relacionamentos ocorrem nos eventos sob seu domínio. Desse ponto de vista, a manutenção do controle sobre uma organização social depende da manutenção do controle sobre os eventos de seu domínio e a manutenção do controle sobre esses eventos depende de manter-se sob controle a intensidade do conteúdo das variáveis envolvidas (ao menos das mais influentes). As respostas desejadas pelos usuários de um sistema de informações gerenciais são sobre o conteúdo dessas variáveis, quais sejam: os valores previstos em determinadas variáveis (valores planejados) e os valores coletados após a ocorrência de um evento, que havia sido planejado. De posse desses valores será possível a dedução de um significado que pode servir ao propósito do controle e da tomada de decisão.

4.3 — O ESTUDO SOBRE EVENTOS NO MEIO INFORMÁTICO

A expressão “eventos” tem sido motivo de estudo no meio informático há, pelo menos, duas décadas. Para DeMarco, por exemplo, a completa modelagem de um software dependeria, também, da modelagem do seu comportamento. Essa modelagem poderia ser feita com um diagrama denominado “diagrama de transição de estados” e seria “... *formado pelos possíveis estados e pelos* ***eventos*** *que fazem com que o sistema passe de um estado para outro*” [DeMarco,1982,p.79] (grifo acrescentado).

Especificamente nessa referência, a mudança de estado do software seria disparada por um evento computacional, como o pressionamento de um tecla de função “[F1]” por exemplo, aplicando-se o conceito de *evento*, nesse caso, essencialmente à comunicação do usuário com o software (comunicação “homem-máquina”). Esse conceito foi estendido ao relacionamento entre o sistema sob análise e o meio onde ele se insere em [McMenamim,1984], para quem as atividades fundamentais de um sistema são um subconjunto de todas as suas respostas planejadas a eventos no mundo a sua volta. O que ocorre, entretanto, é que apesar do modelo de McMenamim ter o mérito de indicar o princípio adequado para uma análise produtiva, princípio esse que poderia ser traduzido pelo binômio “eventos” (ponto de partida, onde podem nascer os problemas de um sistema social) e “essência” (no sentido de “somente os que se justificam” ou “somente os que valem a pena”), a simples identificação dos eventos externos, que exigem da organização em questão algum tipo de reação, não garante que sejam esses os eventos que deverão ser colocados sob controle. A antítese seria o estudo de *todos* os eventos do domínio do sistema (tanto externos quanto internos), só que nesse caso perde-se o benefício do essencial, no sentido acima declarado.

4.4 — A ANÁLISE DE REQUISITOS QUE PROJETA SOFTWARE

O que de fato ocorre, nos modelos analisados, é que existe um forte comprometimento entre o analista de sistemas e o software. Aparentemente, para toda análise de sistemas é imprescindível que a solução seja com um software. A análise não busca compreender os problemas e suas causas, mas procura identificar como uma solução de software poderia ser aplicada, para o problema da empresa. Assim, o problema da empresa deixa de ser o mérito da análise, para ceder lugar ao novo problema criado pelo analista: *como isso pode ser resolvido com um software?* Ora, se o objetivo da análise é a compreensão dos problemas e suas causas, o que vem sendo praticado não é análise de sistemas, mas tão somente a fase de projeto do software está sendo antecipada.

Esse aspecto não está presente apenas nos modelos tradicionais de análise de sistemas, como o modelo da análise estruturada ou o modelo da engenharia da

informação, mas também no modelo orientado aos objetos, como visto anteriormente.

4.5 — OS REQUISITOS DO PONTO DE VISTA DA ANÁLISE DE REQUISITOS

Paralelamente ao fato de os modelos de análise convencionais estarem mais orientados para o projeto do software do que para os problemas dos usuários, corre o objetivo que tem sido buscado com os sistemas de processamento de dados. Esses sistemas não deveriam ter o objetivo, em si, de produzir um resultado que não fosse a informação (ressalvado o conceito restrito de informação de [Warnier,1984], apresentado anteriormente, e levando-se em conta que o processador pode não ser um computador, quer dizer, mesmo quando o processamento seja feito manualmente). Logo, a não percepção de que os sistemas responsáveis pela geração de informações gerenciais não tem a intenção original de "fazer" alguma coisa, mas de fornecer informação (ou "saber" alguma coisa), acaba influenciando a análise dos requisitos, levando à busca de objetivos secundários de como os processos para gerar a informação ocorrem, ao invés de buscarem, primeiro, identificar quais as informações que de fato são relevantes para a tomada de decisão e para a manutenção do controle sobre a organização em direção à sua missão. Um exemplo de modelo que leva a esse desvio pode ser observado em [Yourdon,1990], que dissociou os sistemas de apoio à decisão e de planejamento estratégico dos chamados *sistemas operativos* (ou ainda, sistemas de processamento de ações: como emissão de notas fiscais, duplicatas, cálculo de custos, etc.), como se esses *sistemas operativos* tivessem uma função diferente da de geração de informação. [Katz,1974] já advertia para esse fato, em que o desenvolvimento de sistemas de informações ineficientes ocorriam por causa da concentração da análise em tentar compreender e modelar os princípios de funcionamento organizacional da empresa (ou, como acontecem os processos dentro da empresa), ao invés de tentar compreender as causas que podem fazer com que o sistema saia do controle.

4.6 — O MÉTODO DE ANÁLISE

A minimização dos atritos gerados pelo choque entre os elementos que deslocam-se para compor um sistema social, atraídos a partir da imposição de determinadas energias, é da responsabilidade do subsistema de gerenciamento dessa organização. O risco desse sistema social vir a desaparecer é função da reversão dessa capacidade de atrair os elementos, muitas vezes desencadeada sem que o subsistema gerencial perceba a tendência apresentada pelo conjunto histórico dos conteúdos das variáveis que representam esses atritos. Para que o conteúdo dessas variáveis sejam mantidos sob controle, é necessário que identifique-se:

a) quais são essas variáveis e, entre elas, quais são as mais representativas de cada evento (unidades de medida dos indicadores);

b) os valores dos conteúdos, admitidos para cada uma dessas variáveis (quantidades de unidades);

c) que deverá ser feito para que esses valores sejam respeitados (atividades do processo);

d) os pontos de coleta dos conteúdos dessas variáveis após a execução das atividades dos processos (registro de eventos);

e) os mecanismos que permitam a percepção de eventuais tendências (ferramentas de análise como relatórios e consultas) e

f) a forma de atuação, caso alguma tendência seja observada.

O método, que hora passa a ser proposto, cuja estrutura encontra-se representada na ***fig.4.6.a***, busca cumprir o papel de identificador do conjunto de parâmetros composto pelas alíneas acima. Trata-se, portanto, de um método para a identificação dos processos e dos requisitos de um sistema de informações gerenciais para uma organização social. Sobre a alínea “b”, cumpre salientar que o sistema deverá permitir que, de alguma forma, lhe sejam **alimentados** os valores admitidos para cada uma das variáveis. Não é, portanto, objetivo do método de análise, que está sendo proposto com o presente trabalho, a identificação de **qual** seria o conteúdo ideal para cada variável, sendo esse o mérito (obviamente) de cada caso sob análise.

Esse método faz uso das ferramentas da qualidade total. As que serão aplicadas aos estudos de caso, adiante, foram apresentadas no capítulo 2:

a) para apoio às atividades ou decisões identificadas com as chaves 1, 2, 6, 7, 8 e 9 da ***fig.4.6.a***: o diagrama de causa e efeito de Ishikawa, o gráfico de Pareto, a folha de verificação e

b) para apoio à atividade identificada com a chave 14, uma matriz derivada do QFD.

Além dessas ferramentas, foram desenvolvidas duas planilhas (planilhas de "respostas desejadas") que servirão de suporte às atividades e decisões identificadas com as chaves 1, 2, 4, 5, 10, 11, 12, 13 e 14 na ***fig.4.6.a*** e, adicionalmente, uma ficha que presta-se à documentação do conjunto de atividades e seus indicadores de qualidade, que compõe os processos cujos controles serão feitos imediatamente após o registro do evento de origem da variável a ser controlada, se esse controle não depender da mixagem de dados, ou seja, não depender da combinação de dados de mais de um evento (atividade da chave 16 na ***fig.4.6.a***). Essa ficha, identificada como "gabarito de processos", não faz parte do método de análise propriamente, uma vez que presta-se muito mais à implementação da solução, caracterizando-se portanto como parte das fases de projeto e produção da solução. Entretanto, por ter sido necessária a sua implementação para que os indicadores de qualidade (atributos) desses casos pudessem ser localizados, acrescentou-se o método para a definição desse gabarito ao presente trabalho no anexo 1 (as bases foram os trabalhos de [Pinto,1993] e [Vieira,1996]). No anexo 1a existem alguns exemplos reais da aplicação desse gabarito (a ficha nº 10 é a meta-ficha).

Como pode ser observado na ***fig.4.6.a***, o processo tem início com o levantamento de informações sobre a situação atual. Nesse levantamento de informações busca-se identificar os problemas que a organização social sob análise

está enfrentando.

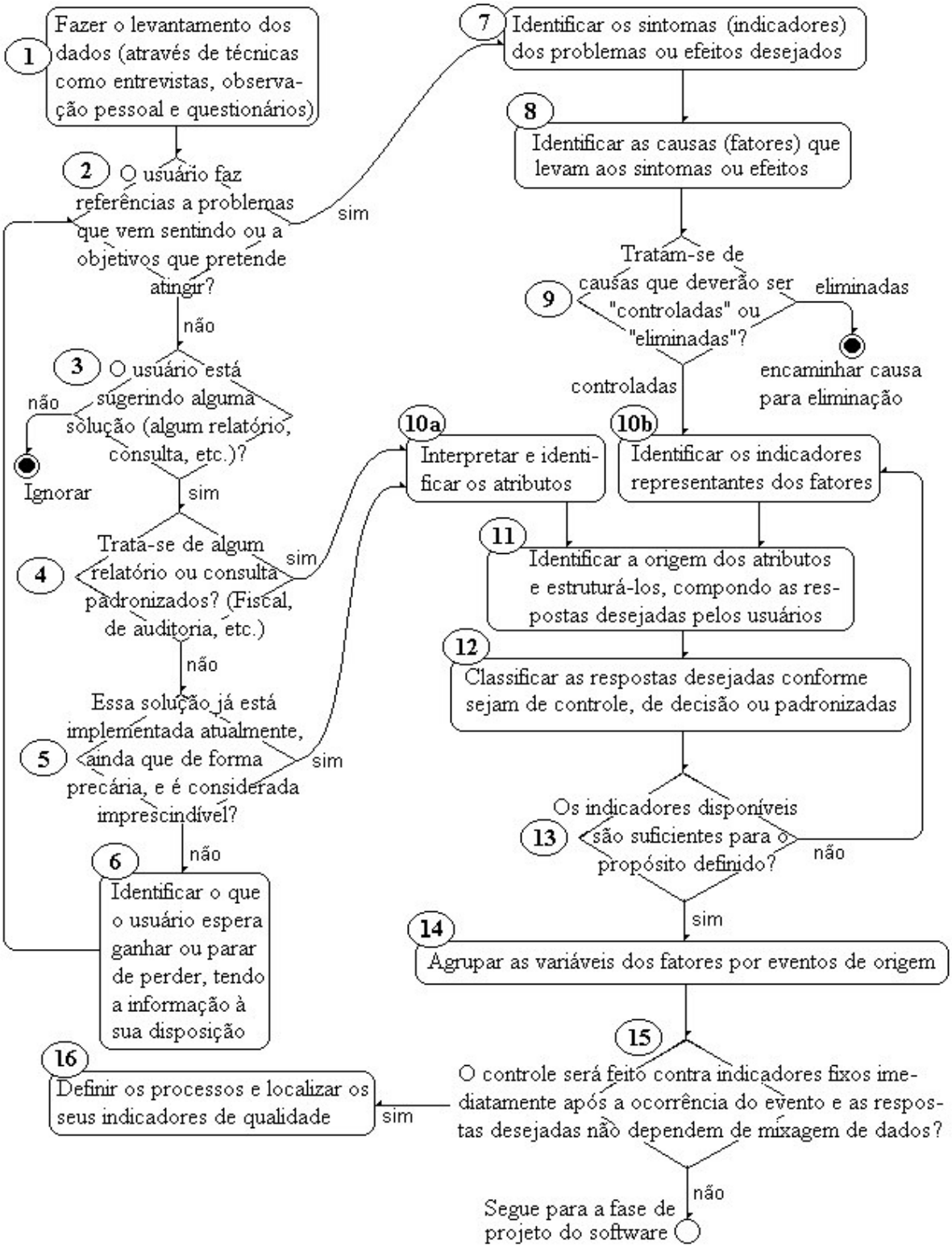


fig.4.6.a — seqüência no processo de análise

4.6.1 — FAZENDO O LEVANTAMENTO DAS INFORMAÇÕES

O modelo prevê dois momentos nos levantamentos das informações:

a) na primeira parte, os usuários têm a liberdade de apresentarem toda a sua causa, todos os seus problemas, todas as suas necessidades, enfim, têm a liberdade de falarem o que bem entenderem, sobre as questões relativas ao seu trabalho e sobre como acreditam que um sistema de informações possa, eventualmente, auxiliá-los. Nessa fase, o entrevistador deverá ficar atento para as expressões que:

   a.a) usuário utiliza e que possam transparecer que ele sente necessidade de informações;

   a.b) caracterizam problemas que ele vem sentindo e

   a.c) caracterizam objetivos de evolução da sua organização.

Esses aspectos precisam ser anotados. O que caracteriza a possibilidade de um sistema de informações vir a ser útil é a presença de certas palavras, nas sentenças formuladas pelo usuário, tais como:

   a.d) verbo “saber”, em expressões do tipo “eu nunca sei ...” ou “eu gostaria de saber ...”. Outra característica é a presença de sentenças como: “eles me perguntam ...” ou “o chefe me questiona sobre...”. Essas sentenças são pistas de que alguma informação é necessária;

   a.e) os verbos “planejar” e “controlar” em todas a suas variações possíveis (p.exe. “nossa meta é ...”, “os estoques estão fora de controle...”): o controle, como visto anteriormente, dá-se a partir da comparação dos dados resultantes do planejamento versus os que são coletados na execução dos processos, caracterizando, portanto, a necessidade de informações;

   a.f) calcular, montar relatório, montar gráfico, registrar, consultar.

Um aspecto importante é que o entrevistador não deve solicitar detalhes sobre qualquer informação que estiver sendo transmitida nessa fase do levantamento dos dados. Não se sabe ao certo os efeitos da interferência do

analista nessa fase, mas nos casos em que isso ocorreu o cliente entrou em detalhes que em nada alteraram o resultado final da análise, a não ser o fato de que o processo de levantamento de dados tornou-se mais demorado. Além disso, desconfia-se que o desvio da atenção do cliente sobre detalhes fazem com que ele perca a noção da sua linha central de pensamento, deixando de exprimir a completeza da natureza do problema como é por ele sentido. O principal aspecto que gerou essa desconfiança foi o fato de surgirem elementos totalmente estranhos à questão que estava sendo apresentada, caracterizando a falta de coesão em determinados momentos do raciocínio. Em hipótese nenhuma o entrevistado deverá ser estimulado, nessa fase, com recomendações de possíveis relatórios ou consultas que o analista presume que sejam importantes. A experiência indicou que o raciocínio dos clientes, sobre a possibilidade de alguma funcionalidade ser disponibilizada por recomendação do analista, é de que se a funcionalidade não vier a ajudar provavelmente também não vai atrapalhar, portanto, ele provavelmente vai aceitar a sugestão (isso ocorreu para todas as sugestões do estudo de caso "Vama", que será apresentado mais adiante), independente da sua real necessidade em relação à sugestão dada. A experiência indicou, também, que essas recomendações podem não ser úteis, mas provavelmente tornarão o projeto mais caro, mais demorado para ser implementado e podem dificultar a operação do sistema, obrigando os usuários à alimentação de determinados conjuntos de dados que não serviam ao propósito inicial do cliente;

b) a segunda fase do levantamento dos dados trata da análise propriamente dita, compreendendo as chaves 2 até 9 (inclusives) da ***fig.4.6.a*** e, para essa segunda fase, deverá ser feito um trabalho prévio de separação dos dados coletados na primeira fase. Nele será feito uso das "planilhas de respostas desejadas" (1 e 2, que serão apresentadas a seguir), do diagrama de Ishikawa e da folha de verificação (entre esses últimos dois, a opção será feita por aquele que melhor representar o caso em questão), que deverão ser preenchidos com os dados já disponíveis. A planilha número 1 e o diagrama de Ishikawa são os guias a partir dos quais será conduzida a segunda parte do levantamento de dados. Nessa segunda fase, os clientes

deverão ser estimulados a fornecerem detalhes sobre as respostas que eles esperam obter do sistema, bem como dos problemas para os quais esperam obter uma solução e suas causas. Os itens mais relevantes de serem estudados serão reconhecidos a partir dos seguintes critérios:

b.a) se a pessoa entrevistada for considerada de nível tático ou estratégico:

b.a.a) problemas que ela percebe e que ocorrem porque não há informações disponíveis que permitam a manutenção do controle sobre a sua organização ou setor;

b.a.b) problemas que ela percebe e que são causados por tomadas de decisão inadequadas, em função da ausência de informações suficientes sobre os recursos ou opções disponíveis;

b.a.c) dificuldade na composição dos relatórios solicitados por auditorias externas, controladoria do grupo ao qual a empresa pertence ou fiscalização do governo;

b.b) se a pessoa entrevistada for considerada de nível operacional, problemas que ela está sentindo nos afazeres do seu dia-a-dia para a geração, em tempo hábil e com a correção e completeza adequadas, dos dados que lhe são solicitados.

Em ambas as fases, serão desconsiderados todos os aspectos operacionais para a obtenção dos dados. O objetivo é que as dificuldades dos processos atuais da organização não exerçam influência sobre “como fazer” para produzir os dados, mas somente sobre “onde” os dados necessários têm suas origens (eventos de origem ou equações envolvendo mais de um atributo).

Um questionamento que levantou-se sobre todo esse processo e mais precisamente sobre as entrevistas com os eventuais usuários do futuro software, é como saber que o processo de entrevistas pode dar-se por encerrado? Por não haverem suficientes subsídios para uma resposta definitiva sobre essa questão, adotou-se que, para um primeiro momento até que hajam as condições para que isso possa ser mais exaustivamente testado, sempre que na segunda fase da entrevista (definida acima) não surjam novas “respostas desejadas” nem novos “problemas” ou “objetivos” a

serem atingidos (que caracterizariam-se como elementos da primeira fase da entrevista) o analista pode dar-se por satisfeito. Ou seja, sempre que na segunda fase da entrevista surjam elementos totalmente estranhos a primeira fase, o analista deverá tratá-los como fazendo parte de uma “nova primeira fase”, dispensando-lhes os tratamentos a partir da alínea “a” da presente seção.

4.6.2 — AS PLANILHAS DE RESPOSTAS DESEJADAS

Como foi dito anteriormente, a segunda fase do levantamento de dados será precedida de uma análise dos dados já obtidos na primeira fase desse levantamento. Essa análise deverá resultar em problemas que o cliente espera que sejam resolvidos (ou resultados que o cliente espera alcançar), que serão utilizados como ponto de partida para a identificação das suas causas (utilizando-se do diagrama de Ishikawa) e respostas que o cliente deseja obter às suas perguntas. Essas respostas podem representar as hipóteses de solução a problemas que os usuários têm, sendo necessário, nesse caso, que as próprias respostas desejadas sejam convertidas para a forma de problemas (ou resultados) e suas causas, para que possam ser testadas quanto à sua real aplicação (vide chaves 4, 5 e 6, da ***fig.4.6.a***). Para facilitar a coleta dos detalhes dessas respostas desejadas, foram desenvolvidas duas planilhas. Essas planilhas permitem identificar-se o que o cliente deseja “saber” do software, ou qual a “resposta desejada?”. O desenho dessas planilhas surgiu a partir da necessidade de um conjunto de atributos que não podem deixar de ser identificados na fase de análise, buscando-se minimizar o envolvimento dos projetistas do software com os futuros usuários do software, reduzindo-se, assim, os atritos gerados por problemas de comunicação.

Os gabaritos dessas planilhas são apresentados nas ***fig.4.6.2.a*** e ***4.6.2.b***. Originalmente havia sido desenvolvida uma única planilha, entretanto, por motivos de volumes de dados diferentes entre cabeçalho e atributos de uma resposta desejada, optou-se pela separação, o que levou ao seus formatos atuais. O significado dos campos da primeira planilha é o que segue, indexado pelas chaves na ***fig.4.6.2.a***:

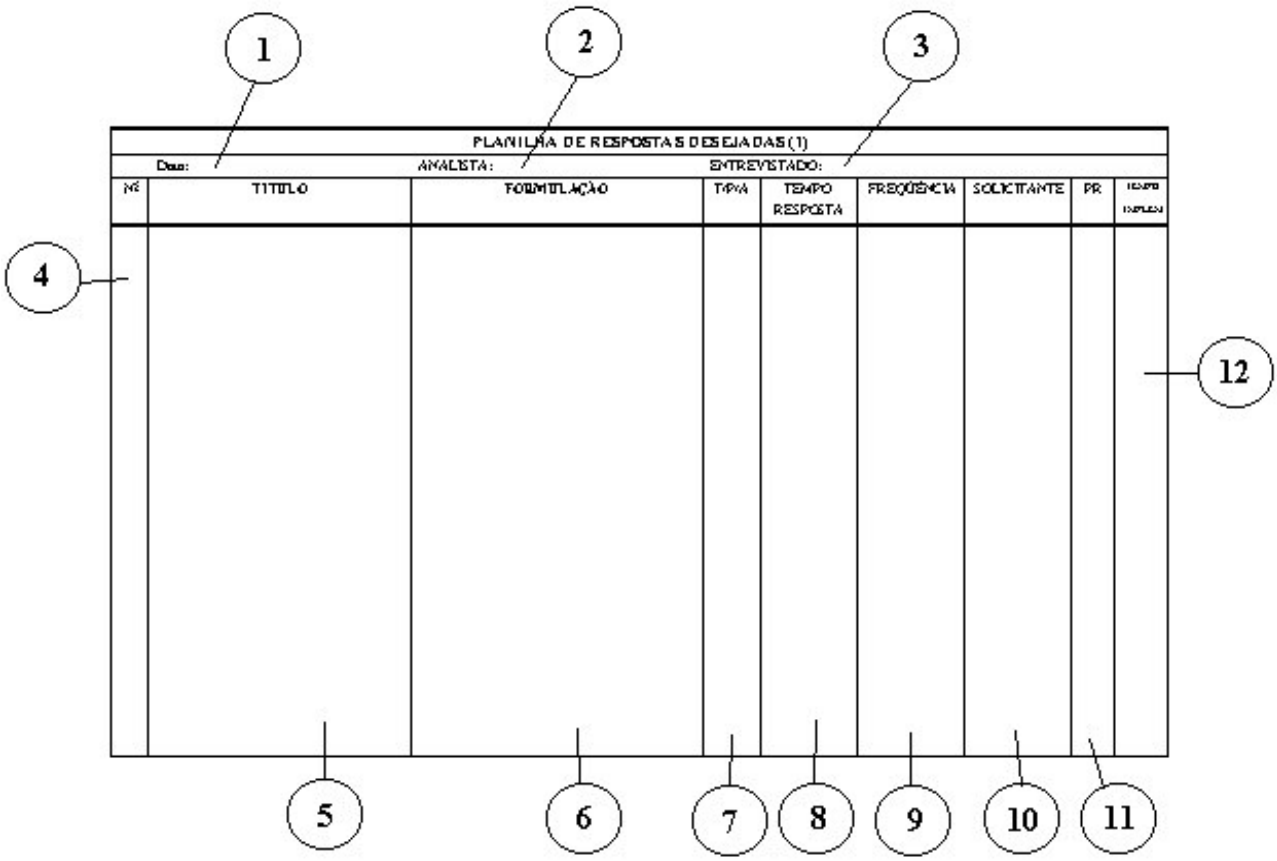


fig.4.6.2.a — planilha de respostas desejadas (1)

1) data: data em que foi feito o levantamento de dados;
2) analista: nome do analista responsável pelo processo;
3) entrevistado: nome da pessoa que foi entrevistada;
4) número: número seqüencial (ordinal) do título da resposta desejada. Esse número é imprescindível para que a segunda planilha possa ser indexada à primeira;
5) título: trata-se da forma como o entrevistado pretende referir-se ao conjunto de atributos que compõe o relatório, consulta ou arquivo em questão;
6) formulação: qual é a estrutura de relacionamento entre os componentes dessa resposta desejada. Tome-se por exemplo um relatório que deverá conter “as vendas efetuadas pelos representantes, em ordem inversa de

valores faturados (valores maiores primeiro), por região de atuação e por data de emissão das notas fiscais". A formulação será definida como segue:

((venda-do-representante / (1 / valor)) / região) / data-emissão.

As principais vantagens na utilização de um modelo matemático na formulação das estruturas de relacionamentos entre os elementos de uma resposta desejada são:

- espaço necessário para a descrição fica relativamente menor, quando comparado ao necessário com os modelos gráficos normalmente utilizados (baseados em desenhos de formulários);
- redução de ambigüidades no momento da definição do formulário, ainda que o tempo entre o levantamento de dados e o projeto do software seja extenso;
- através da incorporação das planilhas de respostas desejadas numa ferramenta CASE (Computer Aided Software Engeneering — Engenharia de Software Assistida por Computador), torna-se possível a derivação automática de parte das estruturas de dados do software;

7) TPA: meio no qual a resposta desejada será fornecida: T=Tela do computador, P=Papel (formulário de impressora) e A=Arquivo (em meio magnético ou ótico);

8) tempo-resposta: qual o tempo disponível para que o software, quando acionado, produza a resposta desejada. Essa informação, aliada à formulação, definida acima, será capaz de fornecer os subsídios para que a equipe de projeto possa definir a estrutura mais racional dos índices secundários da base de dados do software. Um aspecto curioso experimentado nesse item é que o usuário encontra dificuldades em definir esse tempo e, quando estimulado com uma sugestão, poderá reagir das seguintes formas:

   - se o tempo sugerido for muito baixo (10 segundos, para o caso apresentado na alínea 6, por exemplo) ele prontamente aceitará. Como

esse tempo poderá ser difícil de ser atingido, criou-se um problema para a equipe de projeto;

- se, por outro lado, o tempo sugerido for exagerado (10 horas, para o mesmo caso da alínea 6) ele reagirá com uma proposta de tempo bastante razoável. Observe-se a expressão num desses casos: “Tá louco! Quinze minutos, no máximo” (interessante que, antes desse estímulo, o entrevistado não fazia “nenhuma” idéia de tempo);

9) freqüência: com que freqüência essa resposta será solicitada. Essa informação será um complemento ao tempo-resposta, permitindo a sua consistência. Por exemplo, se um tempo-resposta de 10 horas for admitido para um relatório e a freqüência com que esse relatório será requisitado for de 4 vezes por dia, o tempo-resposta deverá ser, obviamente, redefinido;

10) solicitante: usuário ou entidade que solicita os dados constantes da resposta desejada em questão. Pode ser o próprio entrevistado (nesse caso o preenchimento é opcional), um órgão de fiscalização do governo, um gerente ou diretor da empresa, uma entidade de controle particular (auditoria, controladoria da matriz), etc.. Essa informação é importante porque permite que novas informações possam ser obtidas com os solicitantes das informações, caso surja alguma dúvida ou alguma inconsistência seja percebida;

11) prioridade: numa escala entre 1 e o número que representa o último título da entrevista, indica a prioridade relativa da resposta desejada em questão. Esse campo será útil somente quando já existe um software em funcionamento e o levantamento de dados seja para a criação de novos relatórios, consultas ou arquivos. O método a ser adotado para o estabelecimento da prioridade pode ser de livre escolha do analista. Nas aplicações efetuadas com o modelo foram adotados o método GUT (Gravidade, Urgência, Tendência) e o método de Mudge, conforme [Selig]. O preenchimento desse campo será feito somente após todas as respostas desejadas haverem sido configuradas;

12) tempo-implementação: tempo previsto para que a solução seja

implementada. A exemplo da alínea 11, trata-se de uma informação útil apenas quando o software já existir.

O significado dos campos da segunda planilha é o que segue, indexado pelas chaves na figura (***fig.4.6.2.b***):

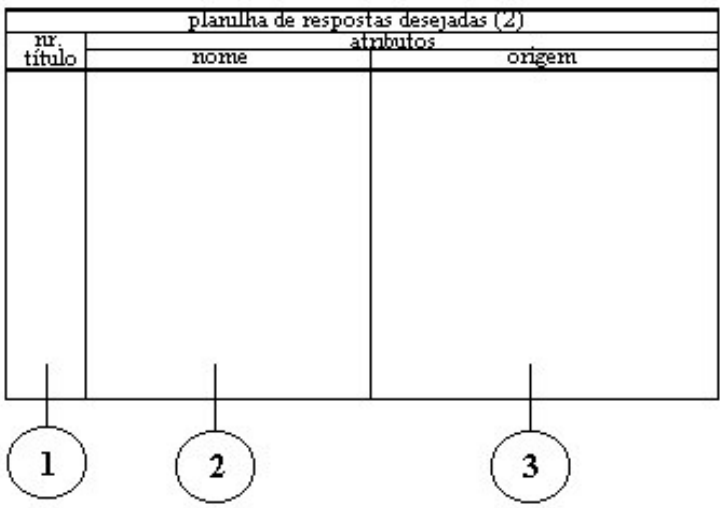


fig.4.6.2.b — planilha de respostas desejadas (2)

1) número do título: esse número precisa ser o mesmo utilizado para o título da resposta desejada na primeira planilha, estabelecendo o relacionamento entre elas (chave 4 na ***fig.4.6.2.a***);
2) nome do atribulo: nome de cada “campo” que compõe a resposta desejada;
3) origem do atributo: identifica a forma como será obtido o atributo em questão. Pode ser:
   - o evento de origem no qual o atributo é gerado e, conseqüentemente, alimentado no sistema;
   - um algoritmo ou equação, através do qual o atributo é obtido. Nesse caso deverão ser utilizados os mesmos nomes de dados já referenciados anteriormente na lista de atributos ou, caso não sejam dados constantes da lista de atributos, deverão ser definidos (entre parênteses) a origem desses dados;

- quando o software já existe: o local onde o usuário percebe o atributo (tela, relatório, etc.). Nesse caso deverá ser feito um trabalho de análise sobre o programa que apresenta o atributo em questão, para identificar-se a sua origem de fato.

Deverá ser tomado um cuidado especial para que os nomes dos atributos e dos eventos de origem sejam informados de forma estruturada, utilizando-se sempre o mesmo sistema de siglas. Isso facilitará a derivação (até mesmo automática) da matriz do QFD, que será vista adiante. No anexo 2 são apresentados princípios que podem ser adotados para a definição de siglas de nomes de dados (atributos e nomes de eventos).

4.6.3 — AS HIPÓTESES DOS USUÁRIOS: O PRINCÍPIO DO TESTE DE REQUISITOS

Juntamente com o levantamento dos dados, coletados e documentados nas planilhas de respostas desejadas e nos diagramas de Ishikawa, deverá ocorrer o processo de análise desses dados, de forma que os requisitos do sistema de informações possam ser obtidos. Para uma melhor compreensão desse processo, tome-se por exemplo o cliente de uma farmácia que apresenta-se diante do atendente e solicita um frasco de anti-ácido. Por não se tratar de medicamento que precise ser vendido com receita médica, o atendente, naturalmente, separa e vende o item solicitado, não tendo a mínima responsabilidade sobre se o remédio vai surtir o efeito desejado pelo cliente. De uma forma geral, a auto-medicação ocorre buscando-se um alívio ao desconforto causado por sintomas de problemas cuja causa pode ser desconhecida. O modelo de resolução de problemas adotado pelo cliente da farmácia segue o modelo mais convencional, e natural, dos modelos para busca de solução de problemas. Esse modelo está representado na ***fig.4.6.3.a*** e, para o caso da farmácia, poderia ser aplicado da seguinte forma: o sintoma de um problema é percebido (dor no estômago), uma hipótese de solução é estabelecida (se for ingerida uma porção de anti-ácido, o mal-estar vai desaparecer), a hipótese é testada (toma-se o anti-ácido). Caso essa atitude não reduza o sintoma do problema a níveis aceitáveis, uma nova hipótese de solução é estabelecida, recomeçando o ciclo. As experiências vividas por outras pessoas e as próprias experiências auxiliam na minimização do número de repetições

desse ciclo.

O que ocorre no desenvolvimento convencional de softwares para informações gerenciais é algo semelhante ao apresentado acima, onde as respostas desejadas pelos usuários às suas perguntas (anti-ácido), formuladas para o sistema de informações, podem ser fornecidas, de uma forma geral, independentemente do conhecimento do usuário (cliente da farmácia) ou do programador de computador (atendente da farmácia) sobre as verdadeiras causas do problema. A exceção está nas informações comprovadamente úteis que são geradas em conformidade com soluções já implementadas (conforme chave 5, da ***fig.4.6.a***) uma vez que, nesse caso, as hipóteses do cliente já foram testadas com essas implementações (ainda que de forma precária).

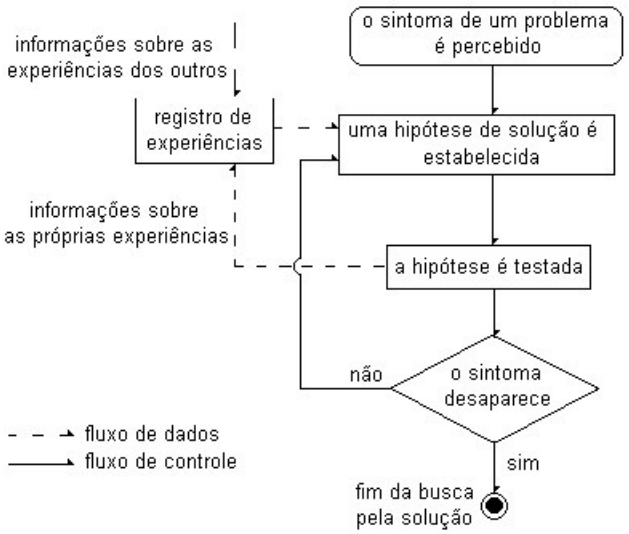


fig.4.6.3.a — fluxo de controle e dados na busca de soluções aos problemas

A partir desse enfoque, considere-se agora como exemplo uma outra situação, em que um determinado cliente convoca um analista de sistemas e solicita o desenvolvimento de um software para o controle dos estoques, a partir da seguinte hipótese:

- se: não faltar material em estoque
- então: aumento na produtividade da fábrica em 25% (porque, aparentemente, as pessoas ficam 25% do tempo paradas na produção, por falta de material).

Suponha-se que para a hipótese definida nesse exemplo resolva-se implementar o software para controle dos estoques e, após a implantação desse software, o requisito "não faltar material em estoque" tenha sido plenamente atendido, mas, mesmo atendendo-se plenamente esse requisito, obtenha-se somente 3% de aumento na produtividade. O efeito é que o software, ainda que atenda o ***requisito***, não atende à ***expectativa*** (ou hipótese subjacente) do usuário, logo, a utilidade do software será tida como praticamente nula. Para o caso apresentado, se o índice de adequação real é o inverso da razão entre o resultado esperado e o resultado atingido (tendo-se que para o fator "produtividade" a busca será de "quanto maior, melhor"), então:

- razão-real = 1 / (25/3) = 0,12 (para uma razão-ideal >= 1).

Analisando-se esse exemplo, percebe-se que as raízes dos problemas com a qualidade dos requisitos de softwares podem estar nas hipóteses estabelecidas pelo cliente. Sendo esse o caso, as expectativas frustradas dos clientes podem não ser mais do que a não confirmação das suas hipóteses subjacentes. Nesses termos, o ideal seria que ele tivesse a oportunidade de testá-las sobre modelos, antes do desenvolvimento do software. Uma tentativa de solução aos problemas de teste de requisitos, antes deles serem transformados em software, são os métodos baseados em protótipos, vistos anteriormente. Entretanto, o que observou-se com esses métodos foi que eles não são úteis quando o software a ser desenvolvido precisa atender dois ou mais grandes processos de uma empresa (como por exemplo os processos de suprimentos de materiais e o de controle da produção integrados). Se esses métodos não se aplicam, resta que as hipóteses sejam testadas sobre os resultados gerados pelo software depois de ter sido construído, implantado e já esteja em operação. O problema é que as hipóteses que não se confirmarem já influenciaram a estrutura desse software, e aí é tarde demais: os recursos já foram empregados, o tempo esgotou-se e as causas dos problemas permanecem fora de controle.

É interessante observar-se que não há nada de errado no princípio dos protótipos integrados à tecnologia RAD, quando ele é enxergado do ponto de vista da

fig.4.6.3.a. Seria o caso de, a partir das hipóteses não confirmadas, estabelecerem-se novas hipóteses a serem testadas, repetindo-se esse ciclo até que uma solução considerada razoável fosse encontrada. O fato, entretanto, é que a razão entre os valores que representam os custos e os que representam os benefícios tendem a tornar-se desvantajosos para os benefícios, na medida em que o tempo passa, por causa da desagregação entre os elementos que compõem o sistema. A justificativa para o volume dos projetos considerados grandes e que são abandonados (vide ***fig.I.b***) pode ser essa: quanto maior o projeto, tanto mais difícil é encontrar-se uma solução razoável num primeiro ciclo completo de teste de hipóteses. Para [Furlan,1998, pag.67] "*a complexidade muitas vezes é mal compreendida e novos requerimentos são amiúde descobertos durante o processo, fazendo com que a equipe de trabalho amadureça com o tempo e visualize novas oportunidades de implementação*". Sob o ponto de vista do presente trabalho, nesse "amadurecimento" da equipe o que ocorre é que as hipóteses inicialmente formuladas pelos clientes não se confirmaram, levando ao surgimento de novas hipóteses.

Sabendo-se que os requisitos deverão ser testados, antes da sua implementação na forma de software, resta a pergunta: como fazer para que as soluções hipotéticas dos clientes sejam testadas antes do desenvolvimento e implantação desse software? Até nesse ponto, a argumentação do presente trabalho não permite uma resposta definitiva para essa pergunta, mas já é possível uma conclusão sobre o que **não** é suficiente fazer do ponto de vista **funcional:** testar-se a solução sugerida pelo cliente simplesmente pela sua **adequação**. Os testes sobre a adequação são válidos caso sejam confirmadas as hipóteses dos clientes, ou seja, teste de adequação é o mesmo que responder ao cliente da farmácia se "um anti-ácido vai diminuir a sua dor de estômago se essa dor de estômago for causada por acidez". Parece óbvio que sim, afinal, aparentemente foi para isso que inventaram anti-ácido! Mas, e se não for acidez a causa da dor no estômago do cliente? A questão toda parece estar no foco ideal que deve-se ter do problema: o efeito ou a causa? Se não há tempo nem recurso disponível para testar-se as hipótese de soluções sugeridas pelos clientes, que, de uma forma geral, são estabelecidas sobre os sintomas dos problemas, talvez deva-se testar as causas desses problemas. Esse é o principal argumento do presente trabalho: o teste de causas é preferível ao teste de adequação do software. Esse teste

de causas será feito da seguinte forma:

a) a partir das soluções hipotéticas imaginadas pelo cliente (relatórios e consultas a um sistema de informações são parte dos “remédios” que hipoteticamente vão auxiliar na solução aos problemas de controle da organização) que não tenham sido ainda testadas, conforme chaves 4, 5 e 6 da ***fig.4.6.a***, deverão ser identificados os sintomas (na forma de indicadores quantitativos ou qualitativos) do problema a ser resolvido ou efeito desejado pelo cliente (conforme chave 7 da ***fig.4.6.a***);

b) desenha-se o eixo central do diagrama de Ishikawa e na extremidade direita desse eixo é colocado o sintoma ou efeito com seus indicadores e a intensidade desses indicadores;

c) aplicando-se as técnicas previstas para a decomposição do efeito em suas causas, passa-se ao desenvolvimento do diagrama de Ishikawa, nunca esquecendo-se de que, para cada causa identificada, deverá ser identificada também a intensidade da influência dessa causa sobre o sintoma ou efeito, utilizando-se sempre o mesmo indicador quantitativo ou qualitativo adotado para o sintoma ou efeito (chave 8 da ***fig.4.6.a***);

d) identificadas as causas, poderá ser utilizado o gráfico de Pareto, visto anteriormente, para que seja identificada a importância relativa de cada causa sobre o sintoma ou efeito;

e) das causas mais representativas que não puderem ser eliminadas (conforme chave 9 da ***fig.4.6.a***), deverão ser identificadas as variáveis que as representariam num sistema de informações, bem como os eventos nos quais essas variáveis deverão ser coletadas e registradas (conforme chaves 10b e 11 da ***fig.4.6.a***);

f) a partir desse ponto, de posse das variáveis que representam os fatores que exercem influência sobre os sintomas ou efeitos a serem controlados, são definidas as estruturas das “respostas desejadas”, mais adequadas ao propósito de cada uma delas (conforme chave 11 da ***fig.4.6.a***). A adequação dessas respostas desejadas é no que refere-se a completeza das variáveis que as compõem (conforme chaves 12 e 13 da ***fig.4.6.a***):

f.a) se a resposta desejada for de controle, deverão ser identificadas as variáveis que representam os valores planejados e as que representam os valores reais dos fatores ou causas a serem controladas, sob pena de o controle não ser exercido por falta de informações, conforme foi visto anteriormente na seção 1.5;

f.b) se a resposta desejada for "padronizada", deverá ser comparado o "lay-out" padrão (extraído de publicações como o Diário Oficial ou manuais internos da companhia) com as estruturas definidas nas planilhas de respostas desejadas, para identificar-se eventuais atributos faltantes. Esse mesmo critério vale para as respostas desejadas que já estejam implementadas no momento da análise, das quais o cliente não abre mão (conforme chave 5 da ***fig.4.6.a***). Nesse caso o "lay-out" padrão básico é o da própria implementação atual;

f.c) se a resposta desejada for para o apoio aos processos decisórios, deverá ser verificado se os elementos de decisão estão presentes nas respostas desejadas. Os elementos de decisão são os diversos dados que representam as opções possíveis, dentre as quais o decisor vai fazer a seleção da mais adequada (por exemplo, um relatório que apresente uma única opção dentre as possíveis, pode não auxiliar em nada, num processo decisório).

4.6.4 — O AGRUPAMENTO DOS ATRIBUTOS POR EVENTOS — O USO DO QFD

O último passo no método refere-se ao agrupamento dos diversos atributos das respostas desejadas, por evento de origem desses atributos (tanto os atributos quanto os eventos constam da planilha de respostas desejadas número 2). O objetivo desse agrupamento antes da fase de projeto do software é o de garantir que não hajam interpretações variádas sobre quando deverão ser alimentados os dados no sistema (ou, quais as mensagens que deverão ser capturadas no sistema de informações). As atividades de normalização dos dados, na fase de projeto do software, poderão ser baseadas sobre o registro dos eventos aos quais o sistema estará sujeito, ao invés de

serem baseadas sobre documentos que posteriormente serão registrados em entidades de uma base de dados (conforme [Neto,1988]). A principal vantagem desse aspecto é a de que os atributos que deverão ser registrados nos eventos em questão já foram testados (sendo o seu registro imprescindível), já os documentos que eventualmente possam representar os eventos (como uma "nota fiscal de compras", no evento "recepção de materiais") possuem atributos que talvez nem sejam necessários à composição de alguma resposta desejada ou talvez nem contenham algum atributo que é necessário a alguma resposta desejada.

A ferramenta que adotou-se para o agrupamento dos atributos por eventos é uma matriz baseada nas matrizes do QFD (visto na seção 2.5.3) e o método de derivação dessa matriz a partir da planilha de respostas desejadas nº 2 é o que segue (a ***fig.4.6.4.a*** permite uma melhor compreensão do que será descrito):

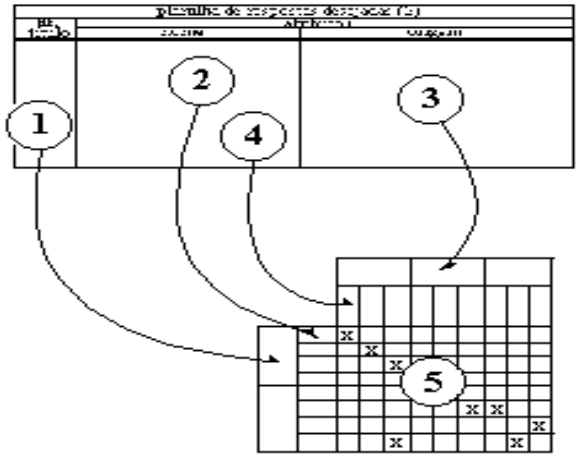


fig.4.6.4.a — derivação da matriz do QFD

a) os números dos títulos são registrados na dimensão mais abstrata do lado esquerdo da matriz (células mais externas à esquerda, conforme indicado pela chave 1 na ***fig.4.6.4.a***);

b) todos os atributos que compõem as respostas desejadas deverão ser registrados ao lado dos números dos títulos equivalentes, conforme indicado pela chave 2 na ***fig.4.6.4.a***. Até esse ponto, trata-se apenas de uma cópia dos títulos e dos atributos da planilha para a matriz;

c) os eventos serão registrados na dimensão mais abstrata do lado superior da matriz (células mais externas superiores, conforme indicado pela chave 3 na ***fig.4.6.4.a***). As origens que não sejam eventos (como equações envolvendo mais de um atributo, por exemplo), não deverão ser registradas nesse espaço;

d) os atributos que compõem as respostas desejadas e que não sejam compostos a partir de equações (que sejam elementares), deverão ser documentados sob a sua origem equivalente, conforme indicado pela chave 4 na ***fig.4.6.4.a***. Observe-se que os atributos comuns a mais de uma resposta desejada, aparecerão somente uma vez nesse espaço;

e) por último, deverão ser estabelecidos os relacionamentos no corpo da matriz resultante (conforme chave 5 na ***fig.4.6.4.a***). Cada atributo do lado esquerdo da matriz (referenciado pela chave 2 na ***fig.4.6.4.a***) deverá ter, no mínimo, 1 (um) relacionamento (indicado por um "x" na célula de cruzamento entre a linha e a coluna dos atributos envolvidos) com 1 (um) atributo equivalente na parte superior da matriz (referenciado pela chave 4 na ***fig.4.6.4.a***). Os atributos compostos a partir de equações (não tendo, portanto, origem no registro de um único atributo de algum evento) terão relacionamento com todos aqueles atributos que fazem parte da sua equação.

O agrupamento dos atributos por eventos, como foi explicado acima, permitirá as seguintes vantagens adicionais ao objetivo estabelecido no primeiro parágrafo da presente seção:

a) análise de impacto na manutenção do futuro software: permite identificarem-se as respostas desejadas que serão afetadas, caso um evento ou atributo de um evento, venha a ser alterado (seguindo-se a coluna do atributo ou evento afetado, todas as relações indicam as respostas desejadas que serão afetadas);

b) no caso de seleção de softwares de mercado, essa matriz permite uma análise de aderência: pelo lado esquerdo entra-se com o conjunto de respostas desejadas (e seus atributos) e pelo lado superior entra-se com

os registros de eventos (e seus atributos) suportados pelo software que está sendo oferecido. Os relacionamentos indicam os atributos necessários e que são suportados pelos registros de eventos do software. As linhas em branco indicam atributos necessários mas que não são suportados pelo software e as colunas em branco indicam atributos que deverão ser alimentados no software, mas que não teriam razão de ser, julgando-se as necessidades da empresa sob análise.

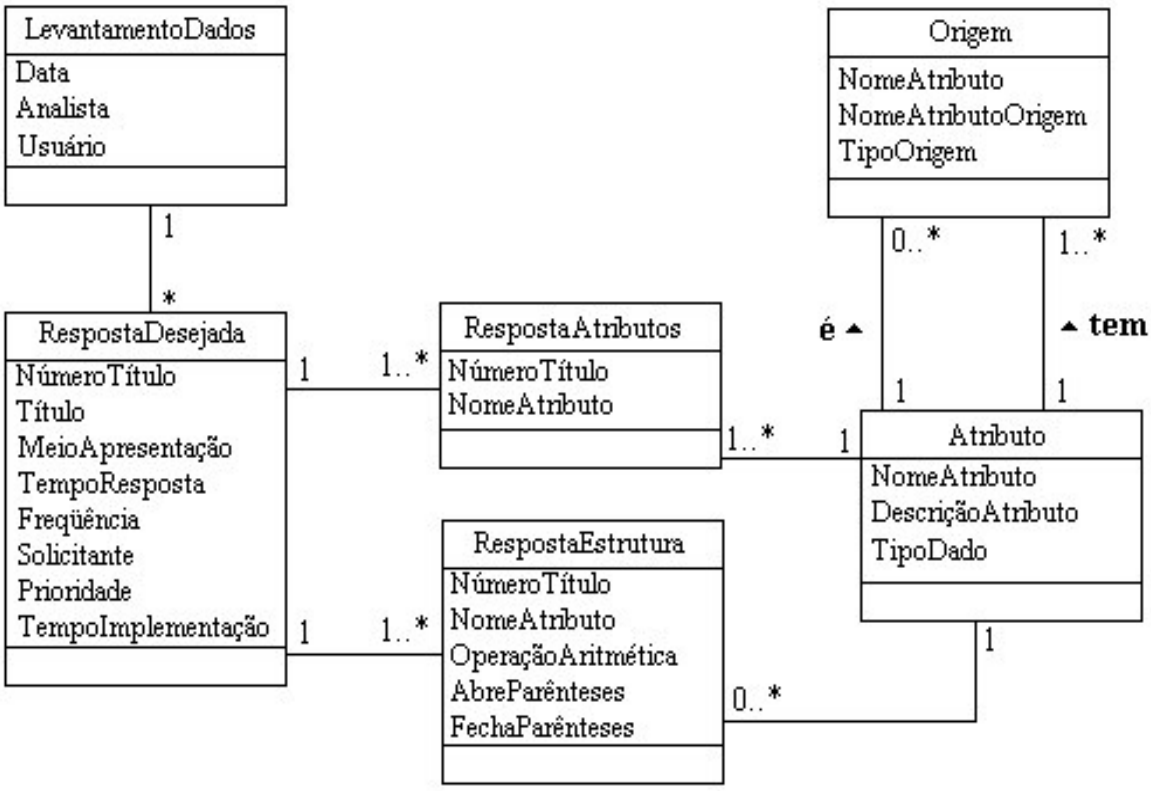


fig.4.6.4.b — estrutura normalizada de relacionamentos entre os dados das planilhas

Um aspecto interessante a ser observado é o relacionamento normalizado dos dados das planilhas de respostas desejadas entre si e a matriz de QFD, para a possível construção de um software que poderia vir a servir de apoio ao processo de modelagem (eventualmente um software de análise de requisitos auxiliado por computador — Computer Aided Requirement Analysis — CARA), integrável a uma ferramenta CASE. Esse relacionamento está demonstrado na ***fig.4.6.4.b***. Utilizou-se para o modelo um diagrama de classes, conforme a notação da UML.

4.7 — CONCLUSÃO DO CAPÍTULO 4

Tendo sido apresentada a proposta de método para análise de requisitos para sistemas de informações gerenciais, passa-se à apresentação de 3 (três) estudos de caso, sobre os quais o método foi aplicado.

CAPÍTULO 5 — APLICAÇÃO DO MÉTODO

Ao longo do desenvolvimento do presente trabalho foram implementados e documentados 56 (cinqüenta e seis) estudos de caso, aplicando-se as técnicas apresentadas no capítulo 4 de forma parcial ou total.

A empresa a partir da qual foram feitas as pesquisas e implementados os testes (diretamente, no caso das aplicações de interesse da própria empresa, ou indiretamente, no caso das aplicações de interesse didático) atua no mercado de sistemas integrados de informação, principalmente para indústrias. Trata-se da Kugel Soft Informática Ltda., situada na região norte do Estado de Santa Catarina, Brasil.

5.1 — HISTÓRICO DA EMPRESA

A Kugel atua no mercado de softwares para gestão empresarial desde 1991 e busca como diferencial nesse mercado o atendimento às especificidades de cada cliente, ou seja, trata-se de uma empresa de serviços de desenvolvimento de softwares “sob medida”, a partir da adaptação de um software que foi inicialmente implementado para duas indústrias do ramo metal-mecânico e que, atualmente, já está em uso por 11 (onze) fábricas, cujos faturamentos variam entre aproximadamente R$ 1.200.000,00 / ano e R$ 60.000.000,00 / ano.

5.1.1 — AS METAS DA KUGEL, SOBRE “QUALIDADE E PRODUTIVIDADE”

A grande meta da Kugel, atualmente, é a implementação de processos que lhe permitam reduzir as diferenças entre os prazos e custos orçados versus os prazos e custos efetivos após a execução dos serviços, para um nível aceitável, qual seja, um máximo de 10% (dez porcento). O presente projeto em muito contribuiu para esse fim. No seu início, as variações ultrapassavam a casa dos 300% e, atualmente, já são

possíveis contratos com variação máxima de 20%. O grande diferencial foi, sem dúvida, a possibilidade de conhecer-se de fato a real necessidade dos clientes antes das adaptações no software serem efetivadas.

5.1.2 — TECNOLOGIAS APLICADAS NA KUGEL

Atualmente a Kugel já trabalha com uma ferramenta CASE, que suporta os processos de desenvolvimento do software nas fases de projeto e produção (vide ***fig.3.2.a***). Essa ferramenta possui dispositivos tais como:

a) prototipadores para o desenho de formulários (telas e relatórios);

b) diagramas de “entidades e relacionamentos” e dicionário de dados, para o projeto dos bancos de dados;

c) diagramas de “estruturas”, para o projeto dos programas;

d) geradores automáticos de determinados programas dos sistemas, tais como os programas de atualização de tabelas do banco de dados;

e) geradores automáticos de manuais de operação do software (a partir do dicionário de dados);

f) referências cruzadas, para análise de impacto, de elementos de dados por tabelas ou por programas, que são muito úteis principalmente no controle das manutenções sobre o software (novas implementações, customizações específicas para um cliente, etc.)

A maior dificuldade, entretanto, era a falta de formalismo nos processos aparte das fases de projeto e produção do software, a saber:

a) comercialização de serviços;

b) análise de requisitos;

c) instalação e implantação do software;

d) treinamentos;

e) faturamento e cobrança dos serviços;

f) manutenção de equipamentos e instalações, etc..

Esses processos foram beneficiados com essa pesquisa, na medida em que puderam ser uma aplicação direta das fichas de processos, cuja descrição encontra-se no anexo 1a, com exemplos reais no anexo 1b (ao todo, até o presente momento, foram desenvolvidas mais de 200 dessas fichas, na Kugel). Entretanto, o processo de análise de requisitos foi o mais beneficiado, hajam vistas às justificativas para a execução da presente pesquisa, já apresentadas anteriormente.

5.2 — APLICAÇÃO

Serão demonstrados, a seguir, três estudos de caso. Eles foram selecionados dentre os 56 implementados por serem os mais didáticos, nos aspectos que deseja-se demonstrar em cada um deles. Esses aspectos são tipificados quando o cliente, durante o levantamento de dados:

a) solicita relatórios ou consultas padronizados ou que já estão implementados e que são considerados imprescindíveis. Trata-se de casos em que o analista deve seguir para a chave 10a na ***fig.4.6.a***, a partir das chaves 4 ou 5. Para exemplificar esse aspecto será apresentado o caso “Folha de Pagamento”;

b) faz referência a problemas que deseja que sejam resolvidos, tendo como expectativa a eventual possibilidade de resolução com um software, mas sem sugerir uma solução com software (não há uma hipótese de solução). Trata-se de casos em que o analista deve seguir para a chave 7, a partir das chaves 1 $\rightarrow$ 2, na ***fig.4.6.a***. Para exemplificar esse aspecto será apresentado o caso “Netville”;

c) solicita relatórios ou consultas que não são padronizados e nem encontram-se implementados, por acreditar que tendo determinados dados

de controle, que constituem-se em suas hipóteses de solução, conseguirá levar a empresa a atingir um determinado objetivo (aumento das vendas, redução no prazo de entrega, etc.). Esses casos geram a necessidade da identificação do que o cliente espera ganhar ou deixar de perder tendo a informação à sua disposição, a partir do que serão testadas essas hipóteses. Trata-se de casos em que o analista deve seguir para a chave 7, a partir das chaves 6 → 2, na ***fig.4.6.a***. Para exemplificar esse aspecto será apresentado o caso "Vama".

5.2.1 — O CASO "FOLHA DE PAGAMENTO"

Um software para "Folha de Pagamento" tipifica os sistemas que têm por objetivo a geração dos relatórios necessários ao controle dos aspectos legais, dos clientes que dele fazem uso. Extensivamente, um software para "Recursos Humanos" seria (pelos conceitos do mercado) mais abrangente, atendendo, além dos aspectos legais, os aspectos de gerenciamento de pessoal, controle de benefícios, grêmio, etc., não sendo, entretanto, essa a preocupação de uma "folha de pagamento".

Pela própria natureza dos relatórios de ordem legal, tratam-se de respostas exigidas do software que não dependem de procedimentos complexos de análise, como seriam os necessários para a identificação das causas de problemas de algum cliente, uma vez que essas respostas são padronizadas e definidas em publicações emitidas pelo Governo.

5.2.1.1 — O levantamento dos dados

Inicialmente foi feita uma visita a uma empresa especializada em serviços de contabilidade, onde foram identificadas as principais fontes de dados necessários ao sistema em questão. Nessa visita identificou-se como principais fontes de informações (tendo sido, portanto, também visitadas) a Delegacia Regional do Trabalho e o Setor de Arrecadação do Ministério da Previdência e Assistência Social. Nesses locais obteve-se

o formulário “Notificação para Apresentação de Documentos”, onde encontram-se registrados os documentos que uma empresa deverá fornecer, caso venha a ser fiscalizada pelo Ministério do Trabalho. Entre esses documentos encontram-se alguns relatórios, que são precisamente as respostas desejadas pelos órgãos de fiscalização, no que refere-se às questões trabalhistas.

5.2.1.2 — A análise e documentação dos dados

Essa fase consistiu-se essencialmente de três etapas:

a) preenchimento das planilhas de respostas desejadas 1 e 2, até os atributos (documentadas no anexo 3);

b) identificação e preenchimento das planilhas de respostas desejadas 2, dos eventos de origem dos atributos, ou equações envolvendo mais de um atributo (documentadas no anexo 3) e

c) preenchimento da matriz do QFD, relacionando os atributos das respostas desejadas aos atributos equivalentes agrupados por eventos de origem desses atributos. Apenas uma parte dessa matriz está apresentada nesse trabalho (no anexo 3).

5.2.2 — O CASO “NETVILLE”

O caso “Netville” foi selecionado pelo fato de ser o mais didático dentre os casos testados em que o cliente não conversa com o analista imaginando uma solução de software, mas apenas apresenta os problemas causados pela ausência de controle. Especificamente no caso “Netville”, o cliente foi “levado” a falar de algum problema, porque foi essa a proposição do teste: que houvesse algum caso real de análise de sistemas, sem que o cliente falasse com o analista apenas sobre questões que presumivelmente ele estaria capacitado a resolver, quais sejam, questões que necessariamente envolvessem software. Dos 56 estudos de caso desenvolvidos, 28 deles foram feitos provocando-se os clientes nessa questão. Um fato interessante: em

100% dos casos em que os entrevistados representavam pessoas jurídicas (25 dos 28 casos), o analista responsável precisou estimular o entrevistado no sentido de que lhe fosse explicitado o problema a ser resolvido. Em nenhum dos casos essa questão surgiu espontaneamente.

5.2.2.1 — O levantamento dos dados

A Netville é uma das mais tradicionais empresas provedoras de Internet em Joinville. No levantamento de dados desse caso seguiu-se precisamente o "script" definido na ***fig.4.6.a***. Assim, foram feitos os levantamentos de dados em duas fases, como o previsto, sendo que na segunda fase o cliente foi *levado* a falar de algum problema que não necessariamente estivesse ligado a uma solução via software, sendo, então, apresentado pelo cliente o problema dos "cancelamentos de contas".

5.2.2.2 — A análise e documentação dos dados

O processo de análise e documentação dos dados consistiu-se essencialmente de quatro etapas (documentadas no anexo 4):

a) a principal fonte de dados nesse caso foi um conjunto de registros que a Netville mantém sobre os motivos que levam os clientes a cancelarem suas contas de internet. Esses dados foram planilhados numa folha de verificação;

b) da análise dos dados constantes na folha de verificação (pelo agrupamento das informações num gráfico de Pareto) foi possível a identificação das respostas necessárias de um sistema de informações, a partir do que passou-se ao preenchimento das planilhas de respostas desejadas 1 e 2 (até aos atributos). Nesse caso especificamente, definiu-se como requisito de informação dois relatórios indicando, em ordem inversa, as diferenças entre um tempo mínimo de acesso (planejado) e o tempo de acesso efetuado pelo usuário à internet (real), bem como um número mínimo de

acessos (planejado) e o número de acessos efetuados pelo usuário à internet (real), para que o principal fator de cancelamento de contas pudesse ser mantido sob controle: o fator “não tem utilidade” desencadeia, em média, 70% dos casos de cancelamento das contas e, desse percentual, foi constatado que 81,63% dos usuários (57,14% do total) não sabiam utilizar a internet, fazendo com que o pagamento da conta acabasse sendo um custo sem utilidade. A partir das respostas desejadas definidas acima, objetiva-se constatar quais os clientes que encontram-se nessa situação, o que representaria a eminência do cancelamento da conta, para proceder-se o treinamento devido;

c) identificação e preenchimento, na planilha de respostas desejadas 2, dos eventos de origem dos atributos e

d) preenchimento da matriz do QFD, relacionando os atributos das respostas desejadas aos atributos equivalentes agrupados por eventos de origem desses atributos.

5.2.3 — O CASO “VAMA”

O terceiro caso analisado foi o de uma empresa especializada em serviços de usinagem e que passa a ser denominado como o caso “Vama”. Esse caso foi dividido em duas etapas e na primeira delas foi desenvolvido um trabalho parcial de análise de requisitos, tendo sido identificadas 80 (oitenta) necessidades de respostas, entre estruturadas, de controle e para tomada de decisão. Nessa análise não foram feitos os testes das hipóteses do cliente. O tratamento dispensado às respostas desejadas nessa primeira etapa foi como se todas elas fossem do tipo padronizado ou já houvessem sido implementadas, já estivessem em operação e já houvessem sido testadas (seqüência das chaves 4 ou 5, alcançando a chave 10a na ***fig.4.6.a***). Nessa ocasião não havia uma perspectiva de implementação de um software novo para a empresa em questão; o objetivo era apenas o de identificar-se as respostas desejadas pelos futuros usuários, para, a partir disso, selecionar-se um software de mercado que pudesse ser aplicado à sua realidade. O resultado dessa primeira etapa está registrado nas planilhas de

respostas desejadas e na matriz de QFD do anexo 5. Observe-se que alguns títulos e formulações fazem referência a anexos que são cópias de documentos coletados por ocasião do levantamento de informações (esses documentos não serão apresentados). Constate-se também que os atributos das respostas desejadas e o seu respectivo reflexo nos eventos não foram registrados na matriz de QFD. Isso deve-se ao fato de que na ocasião da montagem dessa matriz havia uma restrição de tempo. Como no caso real tratava-se de uma análise diagnóstica, optou-se por simplificar o modelo, apresentando apenas os títulos e as suas relações com os eventos equivalentes.

O processo de seleção de software foi desencadeado enviando-se uma cópia dos documentos aos possíveis fornecedores de software, juntamente com as seguintes solicitações:

a) que fosse indicado com um asterisco (*) os títulos que poderiam ser atendidos pelo software, do jeito como encontrava-se na ocasião;

b) que fosse enviada uma cópia do relatório ou consulta em questão, emitido pelo software, como prova da sua capacidade em suprir à resposta desejada em questão;

c) que fosse enviada uma proposta comercial para atender essas necessidades indicadas com o asterisco (*);

d) caso não houvesse disponibilidade para atender a algum título e houvesse interesse em customizar uma solução, que fosse feita uma proposta de preço e prazo de entrega;

e) que fossem indicados os recursos humanos e de hardware, necessários à operacionalização do software.

A lista acima foi considerada como o mínimo necessário de informações (como uma versão simplificada dos atributos requeridos pela ISO 12119, para pacotes de software, cujos tópicos foram apresentados no capítulo 2), para que o fornecedor pudesse ser considerado apto a participar da segunda fase do processo de seleção, no qual o software viria a ser apresentado aos futuros usuários. Na ocasião foram selecionados dois fornecedores de software que iriam complementar-se entre si para o fornecimento de uma solução completa, mas o projeto acabou sendo descontinuado,

principalmente pela preocupação que surgiu de que essas soluções pudessem vir a apresentar problemas de integração, até que houvesse tempo por parte da Kugel, para o desenvolvimento de um projeto de software sob medida para esse cliente.

Decorrido 1 (um) ano dessa primeira etapa, o projeto foi retomado e um novo levantamento de informações foi feito, buscando-se identificar mudanças de requisitos em 51 das 80 respostas desejadas (somente aquelas que ainda não tinham uma solução implementada). Nesse ponto constatou-se um fato interessante: apesar das expectativas (tanto por parte da equipe da Kugel quanto por parte da direção da empresa cliente) de que as necessidades dos usuários mudariam, as mesmas respostas inicialmente previstas foram mantidas pelos futuros usuários. Decorridos 3 (três) meses dessa nova análise de requisitos, concluiu-se a implementação das funções do software que permitiam a emissão de 10 dessas 51 respostas desejadas e o resultado foi que 8 (oito) delas foram alteradas intensivamente antes que pudessem ser utilizadas. Esse fato configurou-se-se como o principal indício, para o presente trabalho, de que um usuário pode formular hipóteses de soluções para problemas que ele presume conhecer, mas na medida em que os detalhes do problema são melhor conhecidos, as soluções tendem a se alterar.

A partir dessa experiência, resolveu-se testar os requisitos dos usuários, aplicando-se a técnica descrita anteriormente, referenciada na ***fig.4.6.a*** pelas chaves 7, 8 e 9. Nessa ocasião foram testadas as hipóteses de 19 (dezenove) das 41 (quarenta e uma) respostas desejadas remanescentes. Na ocasião um determinado projeto do cliente encontrava-se com 8 (oito) meses de atraso em relação aos prazos fornecidos para o seu freguês. A direção da empresa desconfiava que, se tivesse à sua disposição um software para a programação e o controle da produção (que genericamente representava essas 19 respostas desejadas), esse prazo diminuiria substancialmente (na ocasião não foram definidos os indicadores que pudessem representar melhor esse “substancialmente”, apenas havia uma expectativa de que a maior parte do atraso devia-se à ausência do referido sistema). O teste procedeu-se pela aplicação do diagrama de causa e efeito de Ishikawa (apresentado no anexo 5) tendo como resultado geral que, dos 8 meses de atraso no prazo de entrega, apenas 1 mês (13%) poderia ser atribuído a eventuais problemas de programação e controle da produção, menos pelo fato de que uma atribuição objetiva pudesse ser feita e mais pela ausência de outros

fatores conhecidos, aos quais pudesse ser atribuído uma parcela dessa “culpa“ remanescente pelo atraso, o que diminuiria ainda mais a responsabilidade da ausência de um sistema de programação e controle da produção. Em outras palavras, o tempo de 1 mês foi deduzido. De fato, eram apenas conhecidos o tempo de atraso total e o tempo dos demais motivos apresentados. Diante desse fato, o cliente foi levado a refletir sobre a real necessidade desse sistema, concluindo, por fim, que apesar do resultado do teste, o software deveria ser implementado.

Um aspecto interessante observado a partir da aplicação do diagrama de Ishikawa, foi o de que quando o indicador de qualidade representante do resultado desejado for medido em “tempo” (horas, dias, meses, etc., como foi o caso dessa aplicação específica), uma de duas condições deverá ser adotada, para que o modelo não perca o sentido:

a) ou ignora-se o aspecto da dependência temporal entre os fatores que influenciam o resultado, somando-se os tempos de cada seta na composição dos tempos de cada seta central, sucessivamente até à seta central mestra (representante do efeito geral em estudo),

b) ou o modelo deverá ser construído formalizando-se a relação de dependência temporal entre as influências, buscando-se a linearidade temporal no modelo e, nesse caso, deverá ser eleito aquele de maior expressão entre os tempos que compõem uma determinada seta central (não somando-se os tempos dos fatores componentes). Esse princípio aplica-se às influências que, do ponto de vista do tempo, não possuem dependência entre si, podendo ser encaminhadas paralelamente umas com as outras. Nesse caso, obviamente, não é a soma dos tempos que representa o tempo total do efeito em estudo, mas o maior tempo, entre os fatores paralelos. Ainda, cada seta central poderá ser composta, além do maior tempo das suas influências, de um tempo adicional, específico dessa seta central.

CONCLUSÃO

Nesse trabalho foram apresentadas algumas reflexões sobre a importância da análise dos requisitos de sistemas, principalmente como base para o planejamento da qualidade no desenvolvimento de softwares de apoio aos processos de controle de uma organização social. Foram apresentados também alguns princípios teóricos e outros conceituais, adotados atualmente para a análise de requisitos, bem como conceitos relacionados à qualificação de softwares e à estrutura de algumas normas de padrões de qualidade, como a ISO 9000-3 e a ISO/IEC 9126. Por fim, foi apresentado e demonstrado um método de análise de requisitos para sistemas de informações gerenciais, que resume-se à aplicação conjunta das ferramentas da qualidade aliadas a um par de novas planilhas, denominadas de planilhas de respostas desejadas, e a um gabarito de processos, numa estrutura formalizada como um método, apresentada na ***fig.4.6.a***.

A estrutura lógica mestra desse método segue o seguinte raciocínio:

a) as causas dos problemas numa organização social são geradas pelo atrito resultante do processo de acomodação no relacionamento interno ou externo entre dois ou mais elementos de interesse dessa organização;

b) a manutenção do controle sobre os fenômenos gerados nesses processos depende de manter-se sob controle a intensidade do conteúdo de determinadas variáveis, representantes dos atributos dos elementos que estão movimentando-se para formar o sistema e os atributos específicos do próprio evento. As respostas desejadas, pelos usuários de um sistema de informações para controle, são sobre o conteúdo (previsto e real) dessas variáveis;

c) não há informação útil se os eventos não ocorrerem. Elementos isolados podem até possuir atributos, mas isoladamente esses atributos não têm importância alguma, do ponto de vista do controle. Disso deduz-se que a

análise de um sistema social não poderá ocorrer aparte do funcionamento desse sistema;

d) teste das hipóteses de soluções de controle, estabelecidas pelos clientes, deverão ser sobre as variáveis que representam as causas dos problemas (problemas esses que são representados por sintomas indesejáveis ou efeitos desejados), não sobre o resultado da aplicação de um sistema de informações já implementado.

Duas contribuições principais espera-se ter deixado com esse trabalho. A primeira delas está relacionada à construção de um arcabouço interdisciplinar capaz de atingir os interesses dos clientes de softwares de apoio aos subsistemas de informações gerenciais inseridos nas organizações sociais, a partir de um conjunto de bases, ao invés de bases individuais sectárias e, em alguns casos, até passionais, que têm uma orientação muito mais tecnológica do que disciplinar ou teórica, o que leva, conforme as reflexões apresentadas, ao favorecimento da forma em detrimento ao sentido e à utilidade da informação para o gerenciamento dessa organização social.

A segunda contribuição está relacionada à construção de um modelo que pode ser aplicado de forma “produtiva” num ambiente industrial de desenvolvimento de softwares para informações gerenciais, considerando-se que uma das maiores barreiras na aplicação de determinadas técnicas atuais de análise de sistemas de informações refere-se justamente ao excesso de tempo necessário à construção de modelos que, objetivamente, podem não contribuir para a solução aos problemas dos clientes (as estatísticas apresentadas nesse trabalho atestam esse fato). A espera por soluções durante o processo de análise é angustiante para o cliente, que por um lado precisa submeter pacientemente a sua empresa a uma série de processos de levantamento de dados e modelagens e que, por outro lado, ao final do processo, podem não garantir o diagnóstico dos problemas. Com o tempo, os clientes passaram a ter dificuldades em admitir a eventualidade de o analista não ter certeza sobre se uma determinada solução viria a ser útil. Diante dessa eventual inutilidade dos resultados da análise de requisitos, ela é precipitada, cedendo espaço às atividades “mais úteis” do desenvolvimento de software, como o projeto e a programação. Parafraseando Glenford Myers ("*Nós tentamos resolver o problema precipitando o processo de projeto para que sobre tempo*

suficiente no final para descobrirmos os erros que surgiram devido à nossa precipitação no processo do projeto" (apud [Jones,1988, pag.30])), aparentemente nós tentamos resolver o problema precipitando o processo de análise de requisitos para que sobre tempo suficiente no final para descobrirmos os erros que surgiram devido à nossa precipitação no processo de análise de requisitos. Um paralelo semelhante não seria admitido na área médica, por exemplo. Isso seria algo como um paciente com dores diante do médico num pronto-socorro, sendo medicado apenas pela desconfiança de um diagnóstico de cólicas renais. Ainda que esse paciente não queira nem saber; ainda que ele queira apenas que lhe arranquem a dor e pronto, isso não seria admitido. O fato é que o médico não pode ir aplicando qualquer remédio sem ter certeza do diagnóstico, diagnóstico esse que precisa ser feito antes que o paciente comece a ter convulsões por causa da dor. Esse é o desafio a ser enfrentado e vencido pelos métodos de análise de requisitos: a precisão no diagnóstico sobre o problema, com o mínimo de ensaios sobre o ambiente do problema e sem nenhum teste com soluções hipotéticas.

Sendo assim, não é possível terminar a presente dissertação sem que ela seja arremetida sobre o principal aspecto que a motivou: a desconfiança de que muitos dos modelos para a análise de requisitos de software disponíveis e populares não são capazes de permitir a investigação e a compreensão dos sistemas de informações dentro dos sistemas sociais, uma vez que estão muito mais orientados aos aspectos físicos desses sistemas de informações, tais como: meios para transmitir as mensagens, como armazená-las, como recuperá-las, etc., do que com a interpretação do seu real sentido e da identificação da sua utilidade. Conforme [Katz,1974,p.47]:

> Não existe falácia mais universal, persistente, fútil e prejudicial às ciências sociais do que o uso do modelo físico para a compreensão de estruturas sociais. ... Enquanto os autores estiverem comprometidos com um arcabouço teórico baseado no modelo físico, eles não perceberão os fatos sociopsicológicos essenciais do caráter altamente variável e frouxamente articulado dos sistemas sociais.

RECOMENDAÇÕES

Foram identificados alguns aspectos intrigantes no presente trabalho. Entretanto eles não foram exaustivamente testados, de forma que pudessem ser representativos a ponto de serem traduzidos na forma de alguma teoria. Nesse sentido, os seguintes itens deverão ser analisados com mais dedicação, principalmente por causa do que parecem indicar os experimentos já efetuados:

a) os clientes, ao falarem com os analistas de sistemas, faziam pouca referência aos problemas que pretendiam resolver, procurando apresentar tão somente aquilo que julgavam ser a resposta ideal para os seus problemas de controle. Qual será a correlação entre esse fato e as estatísticas apresentadas na introdução desse trabalho, sobre a falência de projetos de software? Qual será a intensidade e o sentido dessa correlação (positiva, negativa, fortemente positiva, nula, etc.)?;

b) no caso “Vama” havia um ano entre o primeiro e o segundo levantamento de informações, mas, ainda assim, as necessidades de informações permaneceram as mesmas. Depois, imediatamente após a entrega de parte do software (apenas três meses depois do segundo levantamento de informações), 80% do que foi implementado precisou ser alterado, caso contrário não seria útil. O argumento do presente trabalho contempla o teste das hipóteses do cliente antes do projeto do software, mas o fato estranho é a questão temporal. Esse fato parece indicar que uma necessidade do usuário não muda necessariamente porque levou-se muito tempo para que ela fosse atendida, como muitos comentários fazem crer, mas ela pode mudar imediatamente após a sua primeira tentativa de aplicação, caso a hipótese inicialmente estabelecida por ele não se confirme, o que levaria a crer que, quanto mais rápido é implementado o software, tanto mais rápida será a mudança de requisitos, gerando expressões do tipo: o cliente não sabe o que quer, o cliente muda de idéia

a cada hora, etc., parecendo que as necessidades atuais de informações seriam muito mais voláteis que há 20 anos, mas, na realidade, as pessoas estariam apenas testando mais rapidamente as suas hipóteses, com softwares que são produzidos cada vez com maior velocidade e que, por esse motivo, cada vez mais rapidamente seriam descartados (aqueles das hipóteses que não se confirmaram). Se isso realmente for assim, há alguma esperança de que as estatísticas apresentadas na introdução do presente trabalho tenham uma alteração substancial, a menos que algum método de testes de requisitos orientado às causas dos problemas seja adotado?

c) Como apresentado no capítulo 4, não se sabe ao certo os efeitos da interferência (com sugestões, observações, etc.) do analista de sistemas na primeira fase dos levantamentos dos dados (aquela em que as pessoas poderão, livremente, apresentar a sua causa), junto ao cliente e futuros usuários do eventual sistema de informações. É conveniente que isso seja investigado, de forma que o método possa ser formalizado sobre bases teóricas, também nesse aspecto;

d) conforme alertado anteriormente, não foram devidamente desenvolvidos os critérios para identificar-se o momento adequado em que os levantamentos de informações devem ser concluídos. Apesar do método ter sido contemplado com um critério, essa questão deverá ser melhor implementada;

e) como já foi sugerido no capítulo 4, adiante deverá ser dada continuidade à implementação de um dispositivo de software que possa servir de apoio ao processo de modelagem das respostas desejadas e integração automática dos dados do QFD à uma ferramenta CASE.

ANEXO 1

O GABARITO DE PROCESSOS

AN.1 — O GABARITO DE PROCESSOS

O gabarito de processos, proposto a seguir, foi desenvolvido objetivando-se criar um suporte sobre o qual os processos de uma casa de software pudessem ser documentados, sem que algum item fosse ignorado, buscando-se a implementação da metodologia de estabilização e melhoria da qualidade de Shewhart (ciclos PDCA de aperfeiçoamento da qualidade), apresentada anteriormente. Como objetivos derivados, esse gabarito também serve como um suporte tecnológico, de apoio à implantação da norma ISO 9000-3, nas atividades do ciclo de vida (planejamento do desenvolvimento e planejamento da qualidade) e nas atividades de suporte (documentação de regras, práticas e convenções e apoio no treinamento). Presta-se, também, ao apoio na implantação das estruturas definidas pelo modelo SEI-CMM, no nível 2 (atividades relacionadas ao planejamento e controle de projetos e processos de engenharia de software, permitindo que tornem-se repetitíveis), no nível 3 (no que refere-se aos programas de treinamentos), no nível 4 (como apoio tecnológico ao planejamento da qualidade) e no nível 5 (estimulando o melhoramento contínuo das atividades dos processos, a partir da realimentação sistemática desses processos com os indicadores intermediários, colhidos nos controles após cada atividade do processo, e estimulando a melhoria dos resultados gerais do processo, pelo estabelecimento controlado de novos objetivos gerais). O gabarito é apresentado na ***fig.AN.1.a***, e o significado dos campos é o que segue, indexado pelas chaves na figura:

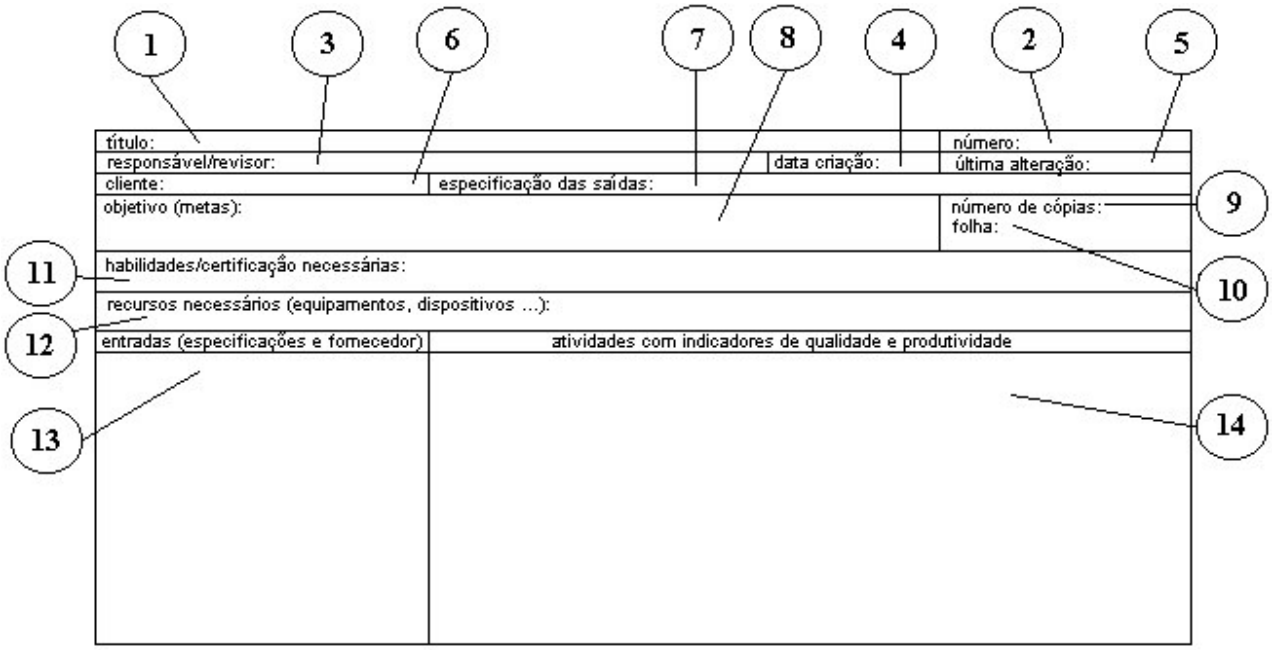


fig.AN.1.a —gabarito de processos

1) título do processo: deve prover uma identidade ao processo, não permitindo ambigüidades com outros processos nem dúvidas no momento da sua seleção, por parte do usuário. Inicialmente, quando a ficha foi criada, a estrutura do título seguia a estrutura sintática de uma oração na língua portuguesa, qual seja: sujeito + verbo + objeto (por exemplo: funcionário limpa a estação de trabalho, programador liga o computador, Regina prepara o café, etc.). Logo nos primeiros gabaritos o sujeito, as conjunções e os artigos foram suprimidos, e o verbo foi para a forma infinitiva (por exemplo: limpar estação de trabalho, ligar computador, preparar café). Após o preenchimento de pouco mais de duas centenas desses gabaritos, observou-se que as pessoas que deveriam valer-se deles para a execução das suas atividades nem sempre o faziam. A impressão inicial era a de que esses processos haviam sido institucionalizados (assimilados, de forma que as pessoas não precisavam mais recorrer às fichas). O fato estranho observado, entretanto, era que o modo como as atividades eram executadas não seguiam, necessariamente, à seqüência do que estava documentado e,

muitas vezes, o processo era totalmente diferente (executado como se ainda não existisse um processo definido). Observando-se esses fatos, foi feita uma entrevista com alguns profissionais que deveriam fazer uso desses processos, para identificar-se os seus motivos. A pergunta feita foi: — temos observado que, em muitos casos, os processos não têm sido usados. Quais são esses casos e quais os motivos? — A síntese do resultado foi a seguinte:

- 100% respondeu que já conhece a maioria dos processos que são executados no dia-a-dia de cor, por isso não fazem uso dos documentos;
- para os casos em que o processo não é conhecido e mesmo assim não é utilizado, 75% respondeu que é muito difícil localizar a ficha entre as existentes, sendo assim, recorrem aos processos somente quando ele é crítico (processos cujo resultado não possa ser garantido sem o uso dos documentos). 25% respondeu que não têm passado por situações em que precisa executar processos desconhecidos.

A partir desse levantamento, para reduzir-se o problema de localização dos processos, uma nova estrutura de agrupamento derivada dos modelos orientados para objetos está sendo adotada. Com essa medida, espera-se que a localização dos gabaritos fique mais eficiente. Isso pode acontecer, em primeiro lugar pela redução no número de documentos (por causa dos aspectos de herança) e em segundo lugar pela facilidade que pretende-se alcançar com o novo tratamento dos títulos desses processos. A estrutura dos títulos passa a seguir o seguinte formato: objeto (que vai sofrer o processo) + verbo no infinitivo (indicando a ação que será aplicada sobre o objeto especificado) + complemento (opcional). Na ***fig.AN.1.b*** está representado um exemplo de como fica a estrutura de relacionamento entre os processos, utilizando-se a notação UML (Unified Modeling Language — Linguagem de Modelagem Unificada) conforme [Booch,1997], em dois diagramas sobrepostos: o diagrama de classes (em preto) e o diagrama de colaboração (em cinza);

2) número do processo: seqüencial, subseqüente ao último existente numa tabela de índices;

3) responsável/revisor: pessoa a ser acionada em caso de manutenção do processo (via PDCA) ou em caso de dúvidas na sua utilização. O revisor funciona como um “fiador” do processo. É a pessoa que será acionada em primeiro lugar, caso algum problema seja identificado;

4) data de criação do processo: permite identificar a idade de um processo;

5) data da última alteração: permite identificar se os processos que foram distribuídos estão atualizados;

6) cliente: entidade que será favorecida com o resultado processo (pessoa, setor, empresa, outro processo, etc.);

7) especificações das saídas: deve indicar o resultado do processo, em termos genéricos. Deve representar o que as pessoas podem esperar ver, após a execução do processo. Por exemplo: nota fiscal preenchida, estação de trabalho limpa, contrato assinado, programa de consulta, etc.;

8) objetivos ou metas do processo: precisa indicar o(s) objetivo(s) específicos do processo em questão. É necessário tomar cuidado para não confundir os objetivos com o título nem com a saída (especificados acima). De uma forma geral, conforme forem estabelecidos, os objetivos estimulam ou não a evolução do processo. Alguns aspectos que podem ser levados em conta, na definição dos objetivos, são:

   - qual a “utilidade” da saída especificada acima, após a conclusão do processo. Para alguns exemplos de saída especificados acima, poderiam ser: permitir a identificação das mercadorias que estão circulando, diminuir o stress no ambiente de trabalho, etc.;
   - os objetivos funcionam como um indicador de qualidade geral do processo, permitindo o teste do processo, como por exemplo: obter contratos com lucros de 7% sobre o custo, permitir a localização de um cliente pelo seu nome fantasia em 10 segundos;
   - o objetivo pode justificar a existência do processo; pode indicar qual o

problema que deseja-se resolver. Por exemplo “confeccionar um programa de consulta aos dados de um cliente” não justifica a existência do processo, uma vez que esse objetivo pode ser alcançado sem um processo formal. Já, “confeccionar um programa de consulta que permita a obtenção dos dados de um cliente em menos de 10 segundos” poderia ser um exemplo de objetivo que justifica o processo;

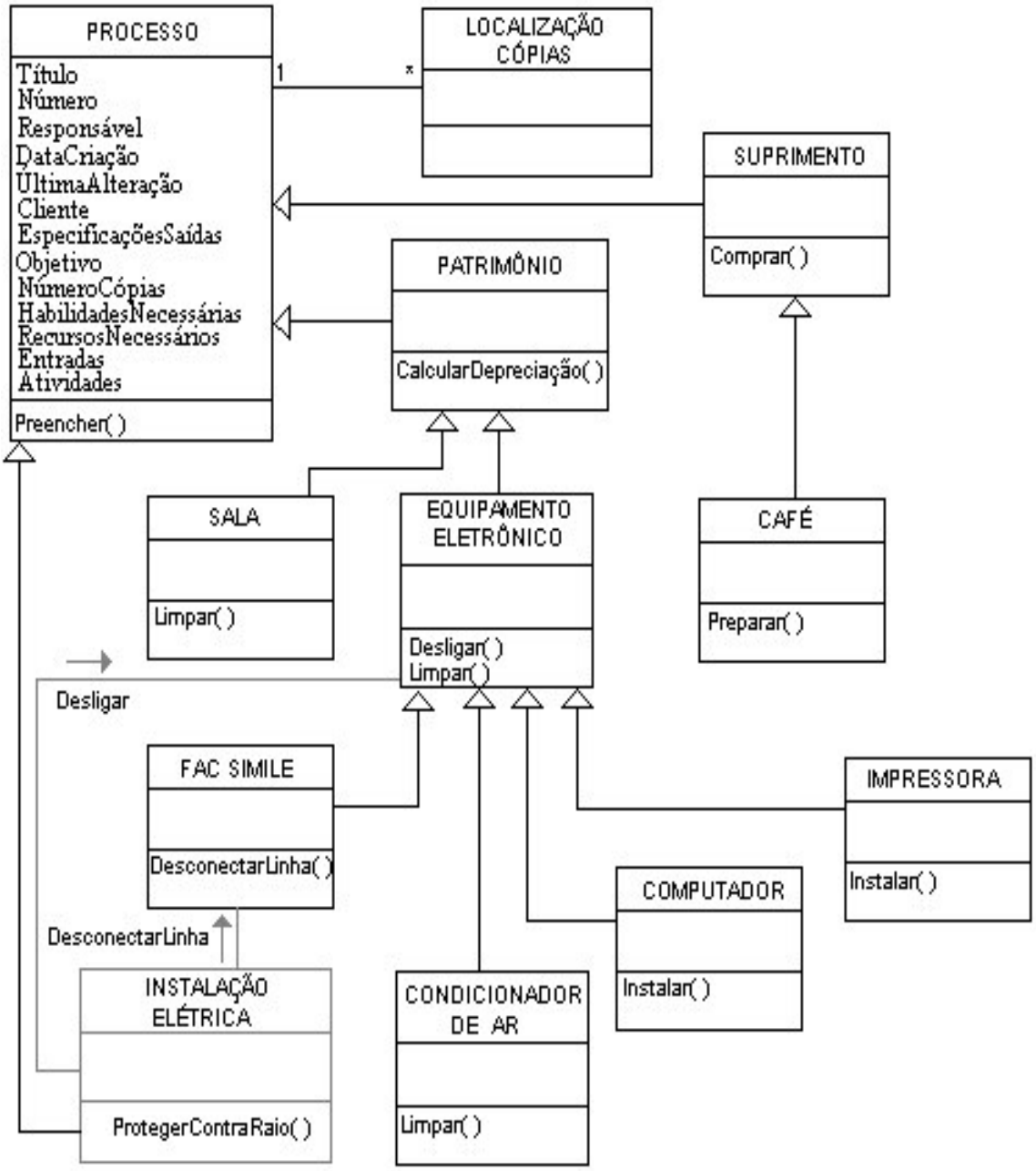


fig.AN.1.b — exemplo de estrutura de relacionamentos entre os processos

9) número de cópias: quantidade de cópias distribuídas, para que se saiba quando da melhoria do processo, as folhas a serem substituídas. A localização das cópias (com o nome das pessoas que receberão o processo) encontra-se na tabela de índices já indicada no item “número do processo”;

10) folhas: quantidade de páginas do processo;

11) habilidades ou certificações necessárias: refere-se aos atributos de formação ou de certificação das pessoas que vão executar o processo. Esse aspecto evita que as atividades transformem-se em algoritmos, apresentando uma sintaxe apenas com declarações de “o que precisa ser feito” sem muitos detalhes de “como fazer”. A partir da definição das habilidades, espera-se que as pessoas que forem executar o processo dominem a essência dos termos empregados. Nos casos em que as habilidades disponíveis não forem suficientes, a expressão “requer treinamento específico” é preenchida nessa célula. Em todos os casos, é sempre interessante que as primeiras execuções sejam feitas com o acompanhamento do responsável pelo processo;

12) recursos necessários: refere-se aos equipamentos, ferramentas, utensílios, ambientes, etc. que são necessários à execução das atividades. É importante que não se confundam os recursos necessários com suprimentos nem com habilidades. Um indicador de qualidade dessa célula é que nela não são registrados itens que se consomem no processo;

13) entradas: aqui sim, são registrados os suprimentos necessários à execução do processo, e até outros processos que sejam executados como pré-requisitos. É importante que não sejam esquecidas as indicações sobre fornecedores ou o local onde a entrada está disponível, bem como marcas, modelos, referências, quantidades, lembrando sempre que alterações nos atributos de um suprimento podem exigir alterações nas atividades do processo, para que o mesmo resultado final seja mantido;

14) atividades: são os passos (ou instruções) do processo. É recomendável

que as atividades venham seguidas de indicadores de qualidade e produtividade, para que o processo possa ser testado ao longo da sua execução, não ficando na dependência exclusivamente do objetivo geral (que, como foi apresentado, pode ser um indicador de qualidade geral do processo). Ainda sobre os indicadores de qualidade e produtividade, observou-se para alguns casos que eles se confundem, não ficando claro se são uma coisa ou outra. Um exemplo que pode ser citado é o caso de um processo de vendas. Para uma determinada atividade reconheceu-se como indicador de qualidade (ou produtividade?) que "o contrato seja conquistado com menos de quatro visitas ao cliente", o que indicaria que os melhores argumentos haviam sido empregados (pelo menos melhores do que os argumentos da concorrência) e a um custo de vendas razoável. A partir da dificuldade em descrever-se atributos que pudessem definitivamente qualificar os dois tipos de indicadores (dificuldade comum na área de serviços), de uma forma geral recomenda-se o seguinte:

- como indicador de qualidade: algo que indique que a atividade foi cumprida com sucesso em termos de objetivos específicos, como por exemplo: "aparece uma tela com o símbolo de uma lixeira" ou "acende-se o "led" da esquerda no painel frontal do modem";
- como indicador de produtividade: algo que indique que a quantidade de recursos usados não é excessiva, ou que leve a crer que o processo está ficando fora de controle em termos de cronograma. O principal recurso a ser controlado é o tempo. Por exemplo, se uma das atividades depende da execução de um programa de computador, e esse programa está levando mais tempo que o normalmente observado (sendo esse tempo definido como o indicador de produtividade), pode ser que esteja em "looping". Se isso vier a ocorrer, deverá ser referenciado o processo adequado (um processo de teste "anti-looping", por exemplo) para que o controle da situação seja recuperado.

AN.1.1 — ALGUMAS CONSTATAÇÕES, SOBRE APLICAÇÕES DO GABARITO

As experiências com o gabarito de processos, aplicado conforme o modelo de processo sugerido por Shewhart, têm indicado que é possível a formalização dos processos, tendo como resultado mais evidente a facilidade no treinamento de pessoal. Num segundo momento esses processos tendem à estabilização, na medida em que os indicadores são alcançados, tanto em nível de objetivo final quanto em nível de atividades. Num terceiro momento, após a estabilização dos indicadores, é possível a identificação de novos objetivos a serem alcançados, recomeçando o processo de estabilização. Nos experimentos, algumas observações puderam ser feitas:

a) nem sempre ocorreu a institucionalização completa de um processo específico, ficando os profissionais que dele fazem uso na sua dependência. Tratam-se de processos cujos resultados são críticos (processos para manipulação de programas de cálculos complexos como custos e necessidades de materiais) ou processos muito pouco utilizados (como impressão de certificados ou de cartões de visita);

b) em alguns casos as fichas passaram a funcionar como uma lista de acompanhamento, para garantir que nenhum passo fosse ignorado, apesar da sua institucionalização (como, por exemplo, os processos de liberação de programas ou de inclusão de um novo usuário na rede de computadores);

c) os profissionais que mais fazem uso dos processos são os que sentem-se inseguros, principalmente os funcionários mais recentes. Entretanto, uma série de erros em processos já definidos foi observada entre os profissionais que já os haviam “incorporado”, ou seja, profissionais que não recorrem aos processos com tanta freqüência. Resta saber até que ponto a “institucionalização” deve ser estimulada, como fator de eficiência, uma vez que nesses casos o tempo perdido para refazer o serviço justificaria a

consulta. Um novo atributo para o gabarito está em estudo. Esse atributo, com base nos riscos presumíveis de eventuais erros, qualificaria um determinado processo como não passível de execução sem a sua consulta.

ANEXO 2

TÉCNICAS PARA DEFINIÇÃO DE SIGLAS DE NOME DE DADOS

AN.2 — TÉCNICAS PARA DEFINIÇÃO DE SIGLAS DE NOME DE DADOS

Há determinadas situações em que a identidade de um atributo (nome do dado) é maior do que o espaço disponível, para as quais são necessárias abreviações (siglas), de forma a obter-se uma identidade completa e significativa do atributo em questão. Isso gera uma nova necessidade: a da padronização de siglas, que permite que uma mesma linguagem seja adotada por toda a equipe responsável pelos processos do ciclo de vida do software. Para esse fim seguem duas sugestões de métodos:

a) um mnemônio de duas letras, correspondentes respectivamente a primeira letra e a primeira consoante da palavra, conforme ilustrado na ***fig.AN.2.a***:

SIGLA	DESCRIÇÃO
CD	CODIGO
PR	PARAFUSO
NR	NUMERO

fig.AN.2.a — siglas a partir das primeiras consoantes

b) serão utilizadas as primeiras letras da palavra, que fazem algum sentido. Se a sigla deve representar mais de uma palavra ela será uma composição da primeira letra de cada palavra (ignorando-se as preposições, conjunções, pronomes, etc.). Caso haja duplicação de siglas serão utilizadas as letras subseqüentes das palavras (uma de cada palavra para as compostas) até que obtenha-se uma sigla exclusiva. Observe-se o exemplo da ***fig.AN.2.b***:

SIGLA	DESCRIÇÃO
COD	CODIGO
PARAF	PARAFUSO
NUM	NUMERO
OP	ORDEM DE PRODUCAO

fig.AN.2.b— siglas a partir das primeiras letras

Para as aplicações feitas na presente pesquisa, utilizou-se o segundo método por apresentar menor probabilidade de duplo sentido. De qualquer forma devem-se tomar alguns cuidados:

a) evitar a utilização dos dois métodos em conjunto;

b) evitar a definição de uma sigla que vier a gerar um sentido pejorativo;

c) nunca utilizar menos de duas letras para a sigla (o recomendável é que sejam entre três e cinco letras).

ANEXO 4

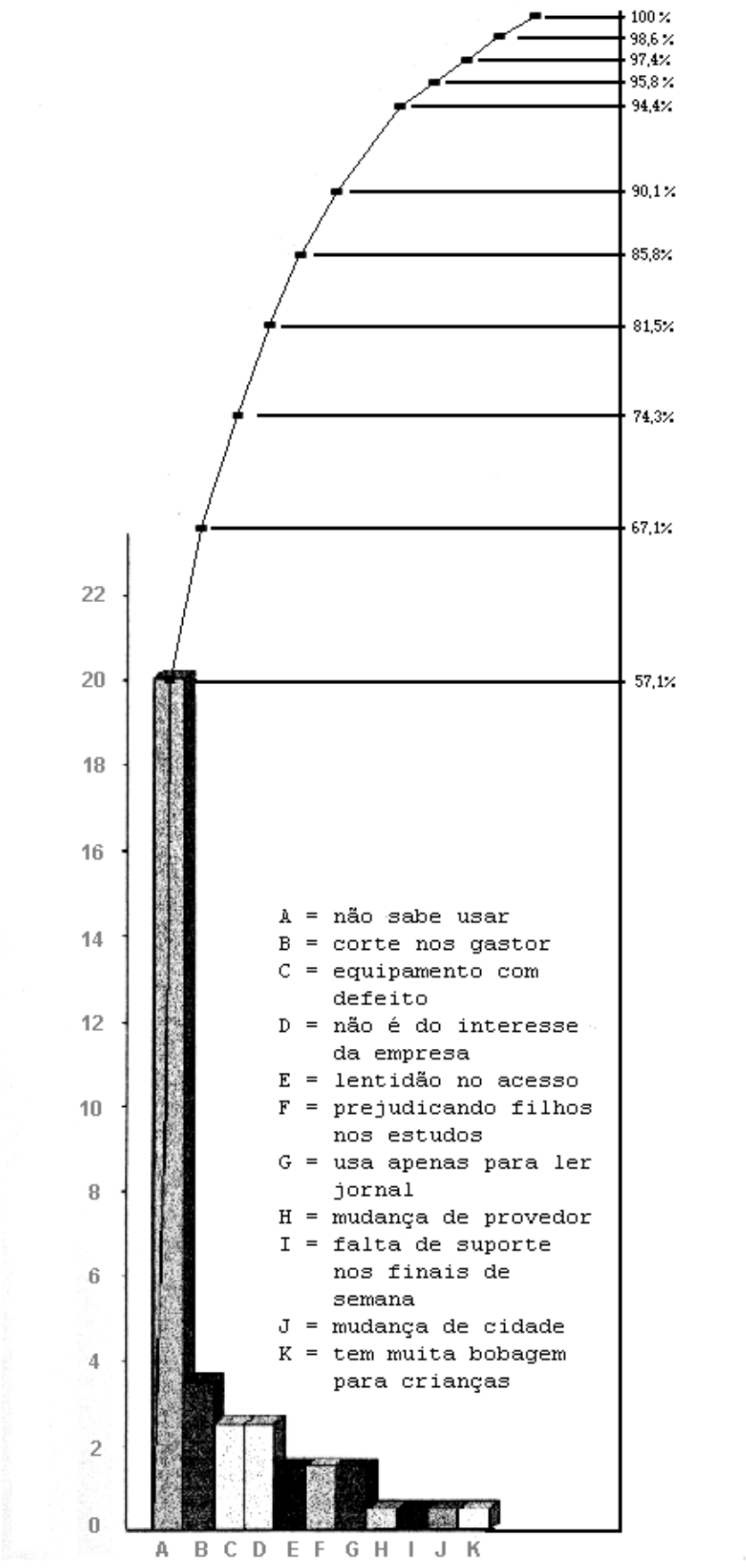
DOCUMENTAÇÃO DO CASO “NETVILLE”

AN.4 — FOLHA DE VERIFICAÇÃO

MOTIVOS DOS CANCELAMENTOS	ABRIL	MAIO	MÉDIA	% / MÉDIA
lentidão no acesso	2	1	1,50	4,29
equipamento com defeito	3	2	2,50	7,14
não tem utilidade	**22**	**27**	**24,50**	**70,00**
corte nos gastos	4	3	3,50	10,00
prejudicando os filhos nos estudos	2	1	1,50	4,29
mudança de provedor	1	0	0,50	1,43
falta de suporte nos finais de semana	1	0	0,50	1,43
mudança de cidade	1	0	0,50	1,43
total	**36**	**34**	**35,00**	**100,01**

Dos que afirmam que “não tem utilidade”:

MOTIVOS DOS CANCELAMENTOS	**ABRIL**	**MAIO**	**MÉDIA**	**% / MÉDIA**
não sabe usar	**18**	**22**	**20,00**	**81,63**
usa apenas para ler jornal	1	2	1,50	6,12
não é do interesse da empresa	3	2	2,50	10,20
tem muita bobagem p/ crianças	0	1	0,50	2,04
total	**22**	**27**	**24,50**	**99,99**


100 %
98,6 %
97,4%
95,8 %
94,4%
90,1 %
85,8%
81,5%
74,3%
67,1%
57,1%
22
20
18
16
14
12
10
8
6
4
2
0
A = não sabe usar
B = corte nos gastor
C = equipamento com defeito
D = não é do interesse da empresa
E = lentidão no acesso
F = prejudicando filhos nos estudos
G = usa apenas para ler jornal
H = mudança de provedor
I = falta de suporte nos finais de semana
J = mudança de cidade
K = tem muita bobagem para crianças
A B C D E F G H I J K

PLANILHA DE RESPOSTAS DESEJADAS (1)								
Data : 26/05/1999		ANALISTA : Márcio Antônio Scopel		ENTREVISTADO : Marcelo Leandro Borba				
Nº	**TÍTULO**	**FORMULAÇÃO**	T/P/A	Tempo resposta	Freqüência	solicitante	PR	Tempo implem.
1	Tempo de Acesso	(Dados-Usuário / (1 / Diferença-Tempo)) / Período	T+P+A	1 Hora	1 / mês	Área Técnica		
2	Número de Acessos	(Dados-Usuário / (1 / Diferença-Acesso)) / Período	T+P+A	1 Hora	1 / mês	Área Técnica		

PLANILHA DE RESPOSTAS DESEJADAS (2)		
Nº TÍTULO	**ATRIBUTOS**	
	NOME	**ORIGEM**
1	Nome-Usuário	Venda Assinatura
	Endereço-Usuário	Venda Assinatura
	Telefone-Usuário	Venda Assinatura
	Tempo-Mínimo	Planejamento
	Tempo-Acessado	Acesso Internet
	Diferença-Tempo	Tempo-Mínimo - Tempo-Acessado
2	Nome-Usuário	Venda Assinatura
	Endereço-Usuário	Venda Assinatura
	Telefone-Usuário	Venda Assinatura
	Quantidade-Mínima-Acessos	Planejamento
	Quantidade-Acessos-Efetuados	Acesso Internet
	Diferença-Acessos	Quantidade-Mínima-Acessos – Quantidade-Acessos-Efetuados

		Acesso Internet		Planejamento		Venda Assinatura		
		Quantidade-Acessos-Efetuados	Tempo-Acessado	Quantidade-Minima-Acessos	Tempo-Mínimo	Nome-Usuário	Endereço-Usuário	Telefone-Usuário
Tempo de Acesso	Nome-Usuário					**X**		
	Endereço-Usuário						**X**	
	Telefone-Usuário							X
	Tempo-Mínimo				**X**			
	Tempo-Acessado		X					
	Diferença-Tempo		X		X			
Número de Acessos	Nome-Usuário					X		
	Endereço-Usuário						X	
	Telefone-Usuário							X
	Qtde-Mínima-Acessos			X				
	Qtde-Acessos-Efetuados	X						
	Diferença-Acessos	X		X				

REFERÊNCIAS BIBLIOGRÁFICAS

[ABNT,1996] Tecnologia de informação — Avaliação de produtos de software — Características de qualidade e diretrizes para o seu uso. Associação Brasileira de Normas Técnicas,1996.

[Akao,1993] Yoji, Michiteru Ono. Recent Developments in Cost Deployment and QFD in Japan. EOQ '93, Helsinki, Finland, pp. 250-256, 1993.

[Andrade,1996] Ana Luísa P., e outros. Aplicação da Norma ISO/IEC 12119 na Avaliação da Qualidade de Produtos de Software. Anais da VII Conferência Internacional de Tecnologia de Software — Qualidade de Software. Curitiba, pgs. 75-89, 1996.

[Bender,1997] Howard J. System and Software Requirements Analysis and Specifications. University of Maryland, 1997.

[Belchior,1996] Arnaldo D., Geraldo Xexéo. e Ana Regina C. da Rocha. Evaluating Software Quality Requirements using Fuzzy Theory. COPPE/UFRJ, 1996, http://www.cos.ufrj.br/~xexeo/artigos/isas96/isas96.htm.

[Bergman,1995] Bo. Quality in Academic Leadership: a Contribution to the Discussion. Total Quality Management, vol. 6, Linköping University, Linköping, Sweden, 1995.

[Booch,1986] Grady. Object-Oriented Development. IEEE Transactions on Software Engineering,1986.

[Booch,1997] Grady, James Rumbaugh e Ivar Jacobson. The Unified Modeling Language for Object-oriented Development, Versão 1.1 http://www.rational.com, September 1997.

[Breitman,1998] Karin Koogan e Julio C.S.P. Leite. Suporte Automatizado à Gerência da Evolução de Cenários, PUC-Rio, 1998.

[Cheng,1995] Lin Chih e outros. QFD — Planejamento da Qualidade. Belo Horizonte, Fundação Christiano Ottoni, UFMG, 1995. 261p.

[Coad,1992] Peter e Edward Y. Análise Baseada em Objetos. Rio de Janeiro, Campus, 1992. 225p.

[Dean,1994] Edwin B.. Requirements Engineering — from de Perspective of Competitive Advantage. INCOSE Conference, 1994.

[DeMarco,1989] Tom. Controle de Projetos de Software. Rio de Janeiro, Campus, 1989.

303p.

[Drucker,1998] Peter. A Quarta Revolução da Informação. Revista Exame, pgs. 56-58, 26/agosto/1998.

[Duailibi,1995] Roberto. Phrase Book. Banco de Frases para Estimular sua Criatividade. São Paulo, Cultura Editores Associados, 1995, 208 p.

[El Boushi,1994] M.e outros. Towards Better Object Oriented Software Desings With Quality Function Deployment. The Sixth Symposium on QFD, pp. 301-321, 1994.

[Fiates,1995] Gabriela G. S. Fiates. A Utilização do QFD como Suporte à Implementação do TQC em Empresas do Setor de Serviços. Florianópolis, UFSC, 1995.

[Furlan,1998] José Davi. Modelagem de Objetos através da UML — *the Unified Modeling Language*. São Paulo, Makron Books, 1998.

[Gane,1988] Chris. Desenvolvimento Rápido de Sistemas. Rio de Janeiro, LTC, 1988. 170p.

[Gause,1991] Donald C. Gause, Gerald M. Weinberg. Explorando Requisitos de sistemas. São Paulo, Makron Books, 1991.

[GDBCP,1979] Grande Dicionário Brasileiro de Consultas e Pesquisas. São Paulo, Novo Brasil Editora Ltda, 1979.

[Hay,1998] David C.. UML Misses the Boat. Essential Strategies, Inc., http://www.essentialstrategies.com/publications/modeling/uml.htm, 1998.

[Hampton,1983] David R.. Administração Contemporânea. São Paulo, McGraw-Hill, 1983. 494p.

[Heylighen,1992] F., C. Joslyn. Introduction to Principia Cybernetica: Cybernetics and Systems Theory. University of Pennsylvania — Penn Engineering, http://pespmc1.vub.ac.be/SYSTHEOR.html, 1992.

[IEES,1997] Instituto de Estudos Econômicos em Software. Programa “Pacote OK”. Campinas, 1997.

[Ishikawa,1993] Kaoru. Controle de Qualidade Total à Maneira Japonesa. Rio de Janeiro, Campus, 1993. 221p.

[Jones,1988] Meilir Page-. Projeto estruturado de sistemas. São Paulo, McGraw-Hill, 1988.

[Kaneko,1994] Noriharu. QFD. Quality Function Deployment JUSE, Japan,1994.

[Katz,1974] Daniel, Robert L. Kahn. Psicologia Social das Organizações. Atlas S.A., São Paulo, 1974, 551p.

[Magnani,1998] Giuseppe. Software Process Improvement. 3ª Semana de Engenharia de Software, CTI, São Paulo, 12 a 14 de agosto de 1998.

[McMenamim,1984] Stephen M., John F. Palmer. Essential Systems Analysis. Prentice-Hall, Inc., 1984.

[MCT,1996] Qualidade no Setor de Software Brasileiro. Brasília, Ministério da Ciência e Tecnologia - Secretaria de Política de Informática e Automação, 1996.

[MCT,1998] Qualidade no Setor de Software Brasileiro. Brasília, Ministério da Ciência e Tecnologia - Secretaria de Política de Informática e Automação, 1998.

[Neto,1988] Acácio Feliciano, José Davi Furlan e Wilson Higa. Engenharia da informação: metodologia, técnicas e ferramentas. São Paulo, McGraw-Hill, 1988. 262p.

[Normandeau,1998] Josette D.. Iso Standards. Ideacom International Inc.,1998. http://www.tradewinds-tv.com/program11/en1100.html

[Nussenzweig,1981] Herch Moisés. Curso de Física Básica — Fluídos — Oscilações e Ondas — Calor. São Paulo, Edgard Blücher, 1981. 500 p.

[Paladini,1994] E. Pacheco. Qualidade Total na Prática. Implantação e Avaliação de Sistemas de Qualidade Total. São Paulo, Atlas, 1994. 214p.

[Pinto,1993] Jane L. G. C.. Gerenciamento de Processos na Indústria de Móveis. Florianópolis, UFSC, 1993.

[Poulin,1998] Louis-Andre. Software Process Assessments & Process-Related Risk Management. 3ª Semana de Engenharia de Software, CTI, São Paulo, 12 a 14 de agosto de 1998.

[Rocha,1998] Ana Regina C.. Qualidade de Software. Joinville, UDESC/UFRGS — Programa de Mestrado em Ciências da Computação, 1998.

[Selig] Paulo Maurício. Apostila de Análise de Valor. Florianópolis, GAV — UFSC.

[Senge,1990] Peter M.. A Quinta Disciplina. São Paulo, Editora Nova Cultural Ltda., 1990. 352p.

[Shlaer,1990] Sally e Stephen J. Mellor. Análise de Sistemas Orientada para Objetos. São Paulo, McGraw-Hill, 1990. 178p.

[SPCI,1996] Some Data on Software Development, Software Productivity Consortium Service Corporation, Herdnon, Virgínia,1996.

[Stange,1995] Plínio. Decomposição da Função Qualidade DFQ. Florianópolis, UFSC, 1995.

[Stange,1996] Plínio. Sobre o Gerenciamento da Produção Orientado para a Qualidade

Total da Empresa com base na Função Perda de Taguchi. Florianópolis, UFSC, 1996.

[Stevens,1988] Wayne P.. Projeto Estruturado de Sistemas. Rio de Janeiro, Campus, 1988. 228p.

[Stuttgart,1997] Fields of Research of the Department of Software Engineering. Software-Labor Stuttgart, 1997. angela@informatik.uni-stuttgart.de.

[Tayer,1996] Richard H.. Software Requirements Engineering: a Tutorial. Second IEEE International Conference on Requirements Engineering. California State University, 1996.

[Tsukumo,1996] Alfredo N. e outros. Modelos de Processo de Software: Visão Global e Análise Comparativa. Anais da VII Conferência Internacional de Tecnologia de Software — Qualidade de Software. Curitiba, pgs. 219-234, 1996.

[Vieira,1996] Sandro R. B. Um Sistema de Gerenciamento da Qualidade para Fábricas Montadoras com Ênfase no Método Tagushi e QFD. Florianópolis, 1996. Dissertação (mestrado em Engenharia). Universidade Federal de Santa Catarina.

[Vieira,1994] Sônia; Ronaldo Wada. As 7 Ferramentas Estatísticas para o Controle da Qualidade. Brasília. QA&T Consultores Associados Ltda., 1994. 133p.

[Ward,1987] Paul T.. Desenvolvendo Sistemas Sem Complicação. Rio de Janeiro. LTC, 1987. 288p.

[Warnier,1984] Jean-Dominique. Lógica de Construção de Sistemas. Rio de Janeiro. Campus, 1984. 194p.

[Wille,1986] Sílvio Aurélio C.. A Natureza de Modelos de Simulação. Curitiba, FURJ-FAE — Texto de Apoio para Educação Continuada — Programa de Pós-graduação em Administração de Materiais, 1986.

[Wilson,1998] Marcos Guerrero. Gestion de Riesgos y sus Aplicaciones, los Metodos S:PRIME y S:PLAN. Anais da IX Conferência Internacional de Tecnologia de Software — Qualidade de Software. Curitiba, pgs. 99-116, 1998.

[Yourdon,1989] Edward. Administrando o Ciclo de Vida do Sistema. Rio de Janeiro, Campus, 1989. 159p.

[Yourdon,1990] Edward. Análise Estruturada Moderna. Rio de Janeiro, Campus, 1990. 836p.

[Zirbes,1996] Sérgio Felipe e José Palazzo M. O., Reutilização de Modelos de Requisitos de Sistemas. UFRG, 1996.

BIBLIOGRAFIA

AKAO, Yoji. Quality Function Deployment. Integrating Customer Requirements into Product Design. Productivity Press, Massachussetts, USA, pp. 331-353, 1990.

ALEXANDER, James A., Michael C. Lyons. Quality Function Deployment: Successful Applications in Sales & Service. EOQ '93, Helsinki, Finland, pp. 257-262, 1993.

AMARAL Filho, Antônio Rubens Anciães. Projeto Estruturado — Fundamentos e Técnicas. Rio de Janeiro, LTC, 1988. 159p.

AZEVEDO, Gláucia D. Franco, Eduardo P. de Souza. Aplicação de Modelos de Qualidade à Especificação de Requisitos. Apostila dos Tutoriais da IX Conferência Internacional de Tecnologia de Software: Qualidade de Software. Curitiba, 1998.

BASILI, Victor R., e outros. Measuring the Impact of Reuse on Quality and Productivity in Object-oriented Systems. Anais da VII Conferência Internacional de Tecnologia de Software — Qualidade de Software. Curitiba, pgs. 17-41, 1996.

BERTALANFFY, Ludwig Von. Teoria Geral dos Sistemas. Petrópolis — RJ, Editora Vozes Ltda., 1975, 351p.

CAUTELA, Alciney L., e Enrico GFP. Sistemas de Informação — Técnicas Avançadas de Computação. São Paulo, McGraw-Hill, 1986. 217p.

CHEN, Peter. The Entity-relationship Model - Toward a Unified View of Data. Massahusetts, ACM - MIT, 1976.

CHEN, Peter. The Entity-relationship Model - A basis for the enterprise view of data. Massahusetts, ACM - MIT, 1977.

DATE, CJ. Introdução a Sistemas de Banco de Dados. Rio de Janeiro, Campus, 1981. 513p.

DAVIS, William S.. Análise e Projeto de Sistemas — Uma Abordagem Estruturada. Rio de Janeiro, LTC, 1987. 378p.

DeMARCO, Tom. Análise estruturada e especificação de sistema. Rio de Janeiro, Campus, 1989. 333p.

DIJKSTRA, Edsger W.. Go To Statement Considered Harmful. MIT, Communications of the ACM, 1968.

El BOUSHI, M.e outros. Towards Better Object Oriented Software Desings With Quality Function Deployment. The Sixth Symposium on QFD, pp. 301-321, 1994.

EMAM, Khaled El, Dennis R. Goldenson. Some Results from the SPICE Phase One Trials Assessments. Anais da VII Conferência Internacional de Tecnologia de Software — Qualidade de Software. Curitiba, pgs. 51-71, 1996.

FISHER, Alan S.. CASE — Utilização de Ferramentas para Desenvolvimento de Software. Rio de Janeiro, Campus, 1990. 264p.

GANE, Chris e T. Sarson. Análise Estruturada de Sistemas. Rio de Janeiro, LTC, 1983. 257p.

GARCIA, Francilene Procópio e outros. Gerência de processos para o desenvolvimento, disponibilização e evolução de produtos de software baseada no molde R-CYCLE. Anais da VII Conferência Internacional de Tecnologia de Software — Qualidade de Software. Curitiba, pgs. 203-217, 1996.

HAMMER, Michael e James C.. Reengenharia — Revolucionando a Empresa. Rio de Janeiro, Campus, 1994. 189p.

HELMAN, Horácio e outros. Análise de Falhas (aplicação dos métodos de FMEA — FTA). Belo Horizonte, Fundação Christiano Ottoni, UFMG, 1995. 156p.

HETZEL, William. Guia Completo ao Teste de Software. Rio de Janeiro, Campus, 1987. 206p.

ISHIZU, Syohei. Roles of QFD in Quality Information Systems. The Eighth Symposium on QFD, pp. 125-129, 1996.

KIHARA, Takami, Charles E. Hutchinson. QFD as a Sturctured Design Tool for Software Development. The Fourth Symposium on QFD, pp. 370-383, 1992.

KUZAWINSKI, Karla M.. How to Apply the Power of Computing to QFD. The Eighth Symposium on QFD, pp. 381-390, 1996.

LAMIA, Walter M. Using the QFD A-1 Matrix to Identify Software Development Risks. The Fourth Symposium on QFD, pp. 555-572, 1992.

LINSTON, Barbara. TQM and Software Engineering: a Personal Perspective. The Fourth Symposium on QFD, pp. 343-360, 1992.

MATHUR, Aditya P.. New Trends in Software Quality Monitoring and Achievement During Software Development. Anais da VII Conferência Internacional de Tecnologia de Software — Qualidade de Software. Curitiba, pgs. 43-49, 1996.

McCLURE, Carma e James Martin. Técnicas Estruturadas e CASE. São Paulo, Makron Books, 1991. 854p.

McMENAMIM, Stephen M., John F. Palmer. Análise Essencial de Sistemas. São Paulo, McGraw-Hill, 1991. 567p.

MOURA, Carlos A. T. e outros. Integração de Técnicas para Análise de Segurança de Software. Anais da VII Conferência Internacional de Tecnologia de Software — Qualidade de Software. Curitiba, pgs. 187-201, 1996.

PAGE-JONES, Meilir. Gerenciamento de Projetos. São Paulo, McGraw-Hill, 1990. 327p.

PARNAS, DL. On the Criteria to Be Used in Decomposing Systems into Modules. Massachussets, MIT, Communications of the ACM, 1972.

RIBEIRO, H. C. S. Introdução aos Sistemas Especialistas. Rio de Janeiro, LTC, 1987. 142p.

ROCHA, Ana Regina C. da, e outros. Requisitos de Ambientes de Desenvolvimento para Suportar Processos de Software. Anais da VII Conferência Internacional de Tecnologia de Software — Qualidade de Software. Curitiba, pgs. 107-121, 1996.

ROCHA, Ana Regina C. da, e outros. Processo para Desenvolvimento de Software baseado em Reutilização. Anais da VII Conferência Internacional de Tecnologia de Software — Qualidade de Software. Curitiba, pgs. 123-137, 1996.

ROCHA, Ana Regina C. da, e outros. Guias de Qualidade em Ferramentas de Desenvolvimento Baseada em Reutilização. Anais da VII Conferência Internacional de Tecnologia de Software — Qualidade de Software. Curitiba, pgs. 139-153, 1996.

SELNER, Claudiomir. Capturando e Entendendo as necessidades dos Clientes. Um Desafio ao QFD Para a Qualidade em Software. Florianópolis, UFSC, 1996.

SELNER, Claudiomir. Sistemas para a Qualidade Total. Um Modelo para o Desenvolvimento de Programas para a Custódia de Dados em Sistemas de Informação. Florianópolis, UFSC, 1996.

SEVERINO, Antônio Joaquim. Metodologia do Trabalho Científico. São Paulo, Cortez Editora — Autores Associados, 1989. 237p.

SHLAER, Sally e Stephen JM. Análise de Sistemas Orientada para Objetos. São Paulo, McGraw-Hill, 1990. 178p.

SMITH, J. A., K. Baker, S. Higgins. The Assessment Of Customer Satisfaction in Higher Education — A Quality Function Deployment Approach, EOQ '93, Helsinki, Finland, pp. 263-268, 1993.

STAA, Arndt von, Hélio Bruck Rotenberg. Adequando Metodologias de Análise Estruturada a Orientação a Objetos. Rio de Janeiro, PUC/RJ, 1988.

STANGE, Plínio. Decomposição da Função Qualidade — DFQ. Florianópolis, UFSC, 1995.

WARNIER, Jean-Dominique. Lógica de Construção de Programas. Rio de Janeiro, Campus, 1984. 185p.

WEINBERG, Gerald M.. Redefinindo a Análise e o Projeto de Sistemas. São Paulo, McGraw-Hill, 1990. 213p.

WEINBERG, Gerald M.. Explorando Requerimentos de sistemas. São Paulo, Makron Books, 1991. 368p.

YORDON, Edward. Administrando Técnicas Estruturadas. Rio de Janeiro, Campus, 1988. 244p.

YORDON, Edward. Revisões Estruturadas. Rio de Janeiro, Campus, 1989. 180p.