

# DESENVOLVIMENTO DE SISTEMAS E O MÉTODO QFD\*

Luís Alves da Silva

\*Artigo publicado em *Developers Magazine*, Ano2 N.14 Out./98

Analisar e definir o que um novo sistema deve fazer para tornar-se valioso para as pessoas que pagam por ele, é uma tarefa problemática em processamento de dados. [Gane, 1983]

Na raiz dessa problemática, estão as dificuldades técnicas e políticas, problemas de comunicação, o entendimento das necessidades dos usuários e o processo de desenvolvimento. Apesar da enorme quantidade de ferramentas e metodologias cujo objetivo é melhorar a qualidade dos aplicativos, ainda persiste a questão de traduzir, adequadamente, para a *linguagem informática* os processos de negócio, suas restrições e *regulações*, somente conhecidos, bem ou mal, pelos próprios interessados na automação.

Os diagramas preconizados pelas metodologias Têm se mostrado insuficientes para resolver a questão do entendimento e são, via de regra, de difícil compreensão pelos usuários e, mesmo quando eles participam da especificação de requisitos, o problema permanece.

Os métodos e técnicas propostos pelo TQC têm mostrado resultado nas mais diversas áreas em que foram aplicados, inclusive em serviços. Será que podem auxiliar a minorar, senão resolver a questão?

O Desdobramento da Função Qualidade (QFD)<sup>1</sup> revela o melhor de outras ferramentas e tecnologias. Foi desenvolvido por seus usuários, está em constante evolução e pode ser adaptado para ir de encontro a uma variedade de necessidades relacionadas com negócios, ajudando a focalizar seus detalhes e, principalmente, a concentrar esforços nas necessidades do cliente / usuário e em alguns pontos que muitas vezes ficam relegados a segundo plano.

O Método do QFD é extremamente útil na identificação e satisfação das necessidades dos clientes, auxilia o planejamento e, conseqüentemente, pode ajudar a melhor estimar prazos e orçamentos, se utilizado, principalmente, em conjunto com as sete ferramentas básicas da qualidade.

**QFD! O que é?** Erro! Indicador não definido.

O projeto do sistema deve ser tal que, quando da utilização, o usuário se sinta satisfeito e que esse sentimento perdure durante sua vida útil. Além de especificar os processos que serão automatizados, as restrições de negócio, procedimentos de controle e outros itens, existem dimensões da qualidade tais como,

“fácil de usar”, “amigável” e outras que apresentam dificuldades para serem traduzidas tecnicamente.

Traduzir os desejos e necessidades do usuário em características do produto e dos processos é o que se denomina **Desdobramento da Qualidade**. O desenvolvimento de software aplicativo é um projeto de engenharia como muitos outros e, assim, também pode ser tratado pelo método QFD que ajuda a traduzir as necessidades dos clientes em requisitos técnicos apropriados, em cada estágio do processo de desenvolvimento e a especificar as ferramentas e técnicas mais adequadas.

QFD é uma forma sistemática de assegurar que o desenvolvimento de funções, atributos, características e especificações do produto, assim como a seleção e o desenvolvimento e escolha de ferramentas, métodos e controles sejam dirigidos para as demandas do usuário ou do mercado [Eureka, 1993].

O objetivo perseguido com a utilização de QFD é assegurar a qualidade do produto através da perfeita identificação de objetivos, determinação de ações específicas, diminuição de problemas e da quantidade de mudanças durante o desenvolvimento, encurtamento da duração dos ciclos, redução de custos / preços e aumento da produtividade.

O QFD resulta numa mudança de enfoque do **Controle da Qualidade** na fase de testes para o controle da qualidade, permanente, em todas as etapas do projeto. Assim, acreditamos que será de grande ajuda na obtenção de um software com a qualidade esperada pelo usuário, pela clara definição dos objetivos e das tarefas necessárias para alcançá-las. Traduzir as necessidades do usuário em requisitos técnicos apropriados em cada estágio do ciclo de vida do projeto do aplicativo. O desafio é montar a equipe do projeto que deve envolver todas as áreas com responsabilidades no aplicativo.

**Erro! Indicador não definido. QFD! Se usar fica mais fácil?**

As ferramentas utilizadas são as *sete novas ferramentas da qualidade* ou ferramentas gerenciais, cujos benefícios estão listados no quadro abaixo.

Na determinação nas necessidades ou requisitos dos clientes/usuários, a equipe de desenvolvimento é desafiada a olhar além do que o usuário está pedindo, para poder determinar a qualidade esperada. A participação do usuário na especificação dos requisitos, das características operacionais, da interface homem/aplicativo, dos controles internos, e de outros

<sup>1</sup> Quality Function Deployment.

atributos é fundamental e será facilitada com o QFD. Esta metodologia é empregada, em diversas áreas, para sintonizar a “voz do usuário” e tem ajudado muitas organizações a encantar o usuário. Vejamos como podemos utilizá-las, também, para encantar o usuário de software aplicativo.

Nossa proposta de aplicação das novas ferramentas da qualidade não está atrelada a qualquer metodologia, pois, o QFD ajuda a realçar o melhor das técnicas e métodos disponíveis em uma área de conhecimento.

Nas metodologias disponíveis, são utilizadas ferramentas de modelagem para focar características importantes do sistema, discutir correções e mudanças nos requisitos do usuário e assegurar o correto entendimento e documentação de forma a permitir que projetistas e programadores construam o sistema. Com tal propósito são propostas algumas ferramentas de modelagem:

*Diagrama de Fluxo de Dados, Diagrama Entidade-Relacionamento e Diagrama de Transição de Estado, Modelos Dinâmicos, Lógicos e Físicos, Encapsulamento, etc.*

Discutamos, então, a aplicação das ferramentas.

**Diagrama de Afinidades** Erro! Indicador não definido.

Tem como finalidade agrupar por afinidade ou relacionamento natural, conjuntos de dados verbais, parciais, fragmentados (retalhos) referentes a um problema complexo ou novo, estimulando a criatividade e proporcionando oportunidade para as pessoas liberarem suas idéias. De forma sistemática ajuda a entender a estrutura dos problemas a serem resolvidos.

O usuário de sistemas como os demais, é tímido, pois é, via de regra, incapaz de exprimir o que quer, embora o saiba, e de explicar o que faz a não ser fazendo. Como o DA deve ser construído a partir de dados verbais oferecidos pelos interessados no assunto via *brainstorm*, as sessões de trabalho que serão levadas a efeito contribuirão para eliminar ou reduzir essa timidez.

O analista trabalhará para descobrir: quais são as funções que o aplicativo deve executar, as

transformações necessárias, as entradas e produtos do sistema, onde as informações são geradas e para onde irão os resultados do sistema.

A técnica JAD foi elaborada para permitir a participação do usuário, mas carece de ferramentas adequadas. Se apoiada na técnica KJ<sup>2</sup>, utilizando a linguagem do usuário, permitirá encontrar aqueles elementos e consenso entre os participantes.

O método do DA usa palavras que expressam fatos, predições, idéias, opiniões e manifestações similares sobre situações confusas ou assuntos sobre os quais não se tem experiência prévia [Bouer, 1992], caso típico em desenvolvimento de software.

Alguns dos grupos de dados afins, necessários à modelagem do aplicativo constam do quadro abaixo.

ORIENTAÇÃO AO CLIENTE	Focaliza as necessidades do cliente Valoriza os critérios competitivos Prioriza recursos Identifica itens sobre os quais deve-se agir Estrutura a experiência anterior
REDUÇÃO DO TEMPO DE IMPLEMENTAÇÃO	Reduz alterações durante o desenvolvimento Limita a introdução de problemas posteriores Evita futuros desenvolvimentos redundantes Identifica oportunidades de futuras aplicações Faz emergir falsas premissas
PROMOVE FORMAÇÃO DA EQUIPE	Cria consenso na equipe Cria comunicação Identifica ações Cria visão global além dos detalhes
PROPICIA DOCUMENTAÇÃO ADEQUADA	Racional para o projeto Fácil assimilação Estrutura as informações Propicia estrutura para análise de sensibilidade

O **DIAGRAMA DE AFINIDADES** ajuda a desbloquear o

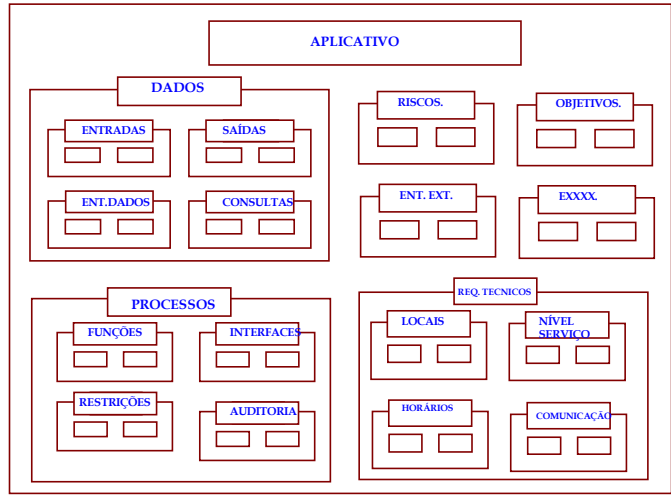
- Objetivo do aplicativo
- Requisitos para Interface Homem/Sistema
- Comportamento dependente de tempo
- Fluxos e Dados e de Controle
- Entidades de dados e seus relacionamentos
- Consultas
- Documentos e Relatórios
- Eventos
- Dados Históricos e prazos de retenção
- Entidades externas
- Fórmulas de cálculos
- Processos envolvidos
- Procedimentos de segurança
- Restrições legais e de negócios
- Características técnicas
- Decisões
- Interfaces com outros aplicativos

pensamento criativo e evitar que os especialistas

<sup>2</sup> O DA é derivado do método *KJ* (marca registrada) desenvolvido pelo DR. Kawakita Jiro.

esqueçam o que o usuário quer, permitindo obter quantidade abundante de dados e organizá-los em grupos baseados no relacionamento natural. É um processo criativo.

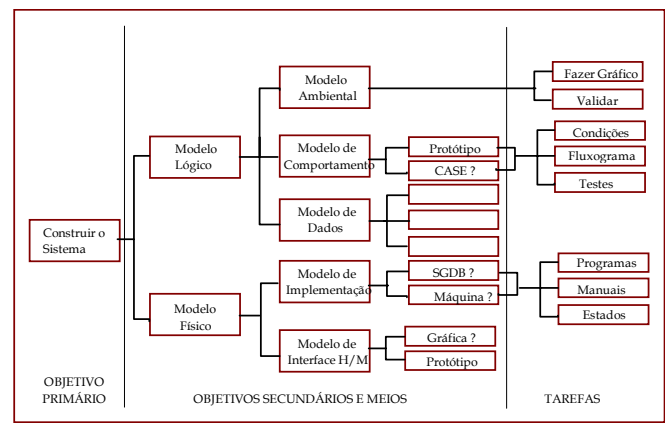
determinando os passos necessários à implementação. A falta de planejamento ou sua qualidade, é um ponto crítico em desenvolvimento de software e causa freqüente de atrasos e estouros de orçamento



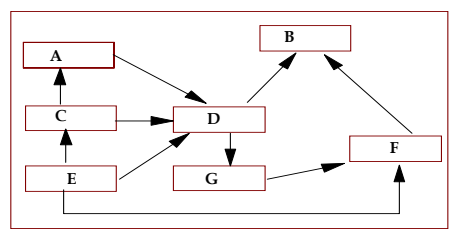
Definido o objetivo e escopo do aplicativo, este diagrama é poderosa ferramenta na determinação dos meios necessários e mais adequados à implementação do aplicativo. Quais modelos devem ser criados, quais técnicas serão empregadas, qual a plataforma de software básico e qual plataforma de hardware, são questões que se elucidarão com a análise proporcionada por este diagrama.

**Diagrama de Relações** Erro! Indicador não definido.

É empregado para facilitar o entendimento de como fatores ou itens relevantes em uma situação ou problema se relacionam logicamente, permitindo destacar os elementos necessários para alcançar um objetivo.



**Matriz de Priorização** Erro! Indicador não definido.



Sua construção, como a do diagrama anterior, é um processo de *lógica criativa*. Estimulando o raciocínio multidirecional

As ferramentas anteriores explodem o problema e levantam inúmeras possibilidades e alternativas para atuação da equipe.

		CRITÉRIOS				PRIORIDADE
		a	b	c	d	
A						4
B						5
C						3
D						2
E						7
F						8
						1

ele revela as conexões lógicas entre os diversos componentes do aplicativo possibilitando organizá-los de forma abrangente, identificando os pontos críticos e atingindo o consenso entre os participantes do grupo de trabalho.

Em que ordem as funções do aplicativo devem ser implementadas? Quais tarefas devem ser executadas prioritariamente? Quais relatórios serão os primeiros? Que consultas são mais importantes? As respostas nem sempre são fáceis.

Nas metodologias de desenvolvimento de sistemas há diagramas de natureza semelhante. O DFD e o DER retratarão mais fielmente a realidade, se construídos a partir do DR. A análise propiciada por este diagrama é similar à de causa e efeito do diagrama Ishikawa.

A matriz de priorização permite ordenar as possíveis soluções, funções e tarefas, a partir de critérios anteriormente estabelecidos, permitindo centrar esforços em pontos de maior impacto para o usuário. Mais uma vez, a necessidade da participação do usuário se torna óbvia.

**Diagrama em Árvore** Erro! Indicador não definido.

Dado um objetivo principal, é preciso um plano e a eliminação de fatores que impeçam a sua consecução. O diagrama em árvore permite desdobrá-lo em objetivos secundários e identificar o que deve ser realizado, mapeando tarefas, métodos, técnicas e ferramentas e

A matriz de priorização evita o risco de manutenção do *status quo*, da mesmice e permite focalizar a atenção sobre as soluções e opções mais importantes antes de se partir para o planejamento detalhado das atividades.

O importante não é a construção da matriz em si, mas o que se consegue em termos de atingir a satisfação do usuário e da adequada aplicação dos recursos.

**Matriz de Relações** Erro! Indicador não definido.

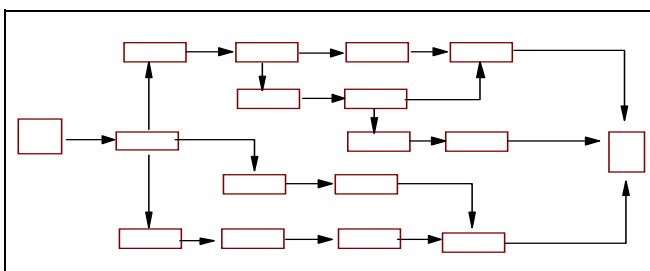
As prioridades estabelecidas anteriormente podem ser validadas através da matriz de relações que é uma ferramenta útil para análise multidimensional. Assim será possível verificar como se relacionam os itens priorizados. Poderia ser aplicada para avaliar a adequação das plataformas de hardware e software disponíveis, na solução para o aplicativo e para vincular os objetivos às necessidades de tre-inamento, proces-sos às pessoas a seus proprietários, serviços a objetos de dados específicos, etc.

	a	b	c	d	
A					4
B	●	△		●	5
C			○		3
D	●	○	●		2
E				△	7
F		○			8
					1

Existem diversos tipos de matrizes de relações e uma indicação dos mais adequados ao desenvolvimento de software aplicativo depende do caso em questão e do bom senso. A análise decorrente do uso desta ferramenta se mostra detalhada e sistemática, podendo estimular o surgimento de novas idéias, novas concepções e abordagens. É possível que, nesta análise, se mostre a necessidade de voltar aos passos iniciais do trabalho e rever os paradigmas pré estabelecidos.

**PDP** Erro! Indicador não definido.

O PDP é um método que permite a identificação de eventos incertos (contingências) durante a implementação de um plano e determina contramedidas (respostas) para esses eventos [Bouer, 1992]. Eventos incertos ocorrem com grande frequência durante o



desenvolvimento de sistemas de informação, o que é suficiente para propormos o emprego desta ferramenta para o planejamento dos trabalhos. A análise do processo de desenvolvimento, apoiada pelo PDP permite explorar caminhos alternativos a partir de um ponto de partida até o final desejado e identificar as melhores opções em termos de definição de medidas preventivas.

O PDP é uma ferramenta que permite antever e gerenciar ocorrências inesperadas. O princípio subjacente ao PDP é que em qualquer percurso até

um objetivo sempre nos deparamos com incertezas e riscos. O inesperado precisa ser antecipado e precisamos dispor de um elenco de medidas para superar os desvios decorrentes. Bouer [op. citada] recomenda seu uso em casos onde a atividade sob planejamento é uma novidade, o plano é complexo, os riscos de desvios são altos, o prazo é crítico e as contingências são razoáveis. Tais recomendações caem como luva no processo de desenvolvimento de software.

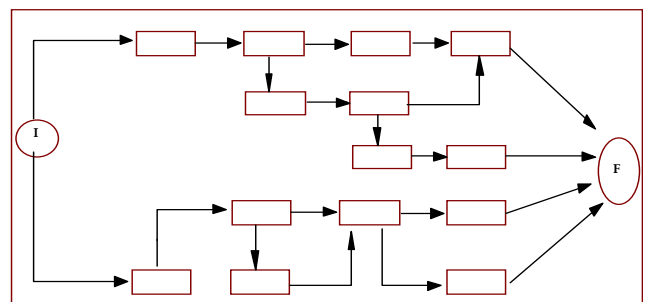
Além disso esta ferramenta nos parece útil no detalhamento dos processos automatizados de modo a identificar adequadamente as opções lógicas que são os casos IF-THEN-ELSE, assim como o estudo dos procedimentos de controle e segurança.

Acreditamos que o PDP pré-planejado seja o mais indicado no nosso caso, antecipando os problemas, na sua maioria bastante conhecidos dos especialistas, antes do início da implementação. Cada etapa do ciclo de vida deve ser cuidadosamente estudada, identificando-se os possíveis resultados com perguntas do tipo "o que deve ser feito se X ocorrer? e no caso de ocorrer B ou C, o que fazer?"

**Rede de Atividades** Erro! Indicador não definido.

O diagrama da rede de atividades nada mais é que o velho conhecido PERT/CPM e é empregada para planejar o cronograma mais adequado para a implementação do software, isto é para transformá-lo em realidade concreta, detalhando-se o encadeamento de todas as atividades necessárias. Muitos projetos se tornaram grandes fracassos devido à falta de um cronograma realista.

O atendimento dos prazos é um ponto crítico no desenvolvimento de sistemas, exigindo que o tempo de execução das tarefas seja bem gerenciado. As recomendações de Bouer sobre sua aplicação, também, se enquadram, perfeitamente, no nosso caso: os recursos são os mesmos para quase todas as atividades, as sub-tarefas são familiares e os tempos relativamente conhecidos, muitas atividades são críticas para o sucesso, existem caminhos alternativos a considerar e quando eventuais folgas podem ser vir



como válvula para minimizar gargalos que podem comprometer o prazo final..

A questão do prazo e do orçamento fazem parte da *crise do software*. Assim, estabelecer um plano adequado para o projeto e acompanhar seu andamento e conhecer a duração de cada tarefa é fundamental, inclusive para elevar o nível de maturidade da organização que se dedica a tal empreitada.

## Pré-requisitos

O uso conjugado dessas ferramentas é que traz garantia de sucesso. Entretanto, sua aplicação pressupõe alguns pré-requisitos necessários para aumentar a eficiência no uso e evitar o fracasso [Moura, 1.994]:

- **Reunir as pessoas certas** - os componentes da equipe devem contribuir com todo o conhecimento disponível e dispor de autoridade para definir soluções.
- **Acreditar que planejamento economiza tempo** - O tempo total, será menor. Retrabalhos e desvios serão evitados, aumentando a probabilidade de cumprimento dos prazos e a equipe poderá se contrapor a prazos determinados apenas por interesses comerciais.
- **Conhecimento das ferramentas** - qualquer ferramenta exige treinamento prévio prático e não apenas teórico. Ishikawa há muito tempo atrás concluiu que “*qualidade começa e termina com treinamento*”.
- **Valorizar opiniões adversas** - É preciso saber valorizar aquelas que nos são antagônicas e não apenas tolerá-las. A discussão construtiva leva ao sucesso e exige novas formas de organização do trabalho, tal como a organização celular.
- **Bom senso** - é necessário para saber quando cada ferramenta, técnica e método deve ser utilizado e até onde prosseguir com o destacamento, evitando perder tempo, ganhar frustrações e úlceras.

## Conclusão

Sem dúvida uma pesquisa mais profunda se mostra necessária para montar um esquema adequado de aplicação do QFD. Entretanto, acreditamos ser uma forma de, pelo menos, reduzir as causas de insatisfação dos usuários, elevar o nível de maturidade do processo de desenvolvimento, aumentar a qualidade, a produtividade e manter os custos em patamares aceitáveis.

E então? Que tal tentarmos integrar o QFD e as sete ferramentas básicas da qualidade à metodologia que estamos praticando? Pode ser passo valioso.

Luís Alves da Silva

Tel. 3050-2344 E-mail: luisilva@usp.br

## Referências Bibliográficas Erro! Indicador não definido.

Moura, Eduardo *As Sete Ferramentas Gerenciais da Qualidade* - Makron Books Brasil, 1.994

Bouer, Gregório Apostilas utilizadas nos cursos da EPUSP em 1.993.

Pressman, Roger S. *Software Engineering - A Practitioner's Approach* 3d Ed. McGraw Hill International Editions 1.992

August, Judy H. *JAD Joint Application Desing* - Makron Books Brasil, 1.993

Eureka, W. E. & Ryan, N. E. *QFD - Perspectivas Gerenciais do Desdobramento da Função Qualidade* - Qualitymark Editora RJ, 1.993.

Bossert, James L. *Quality Function Deployment. A Practitioner's Approach* ASQC Quality Press - Milwaukee, Wisconsin, 1.991

Gane, Chris & Sarson, Trish. *Análise Estruturada de Sistemas* - LTC Editora, 1.988

McMenamim, S. M. & Palmer, J. F. *Essential Systems Analysis* - Prentice Hall, Inc Englewood Cliffs, New Jersey , 1.984