

QFD aplicado ao desenvolvimento de *software*: priorização de requisitos do cliente em uma matriz de funções

*QFD applied to software development: prioritization of customer requirements
in a matrix of functions*

Lisiane Scalvenzi

Graduada em Logística pelo Centro Universitário
Internacional.
Curitiba, PR [Brasil]

Paulo A. Cauchick Miguel

PhD pela Universidade de Birmingham, Inglaterra e Livre
Docente pela Escola Politécnica da Universidade de São
Paulo – USP, Professor Adjunto do Departamento de
Engenharia de Produção e Sistemas da Universidade Federal
de Santa Catarina – UFSC.; Professor credenciado no
Programa de Pós-graduação em Engenharia de Produção da
Escola Politécnica da USP.
Florianópolis, SC [Brasil]
cauchick@usp.br

Resumo

O Quality Function Deployment (QFD) é um método usado para desenvolvimento de produtos em diversos setores industriais. Uma de suas aplicações é no desenvolvimento de *software*. Neste trabalho, objetivou-se demonstrar uma aplicação do QFD e obter melhor entendimento dos requisitos dos clientes, facilitando o desenvolvimento do *software*. A aplicação foi baseada em dados e informações já existentes utilizadas no desenvolvimento de um *software* para captura de imagens advindas de equipamentos radiológicos, sem o uso do QFD. Comparando-se o desenvolvimento com e sem o QFD verificou-se melhoria na compreensão sobre o entendimento e priorização dos requisitos do cliente; porém, observaram-se pontos negativos, como baixa priorização de funções essenciais do programa. Este trabalho destaca-se por possibilitar a construção de uma matriz de funções. Com isso, concluiu-se que a utilização do QFD é adaptável ao desenvolvimento de *software*, proporcionando um suporte importante na identificação de prioridades de desenvolvimento, organização e documentação de requisitos.

Palavras-chave: Desenvolvimento de *software*. QFD. Matriz de funções.

Abstract

Quality Function Deployment (QFD) is a method extensively used in product development in a wide range of industries. One of the applications is for software development. In this paper, we seek to demonstrate a QFD application to assist a better understanding of customer requirements, thereby facilitating the development of software. The application was based on existing information used in the development of a software application for capturing images obtained from radiological equipment without the use of QFD. The comparison between the development with and without QFD not only had positive results, such as a better understanding of the customer's requirements, but also negative ones, such as the low priority given to essential functions of the software. A point of distinction of this paper in relation to other ones that apply QFD in software development is the construction of a function matrix. We conclude that the use of QFD is adaptable to software development by providing support for the identification of development priorities, organization, and documentation of requirements.

Keywords: Functional matrix. QFD. Software deployment.

1 Introdução

O crescimento do mercado de desenvolvimento de *software* e da concorrência entre as empresas desta área aumenta também a preocupação com a qualidade deste tipo de produto, visando a alcançar a satisfação dos usuários. Essa preocupação se reflete diretamente no desenvolvimento de *software*. Quando o planejamento é estruturado e alinhado com os requisitos dos clientes, o tempo de desenvolvimento e o retrabalho diminuem, bem como a qualidade aumenta. Nesse processo, o uso de ferramentas de suporte auxilia no desenvolvimento de *software* e um dos métodos que podem ser utilizados é o Desdobramento da Função Qualidade – ou Quality Function Deployment (QFD) – que tem como propósito traduzir a voz do cliente em requisitos técnicos (ou características da qualidade). Apesar de o QFD ser extensivamente utilizado em produtos (bens duráveis e não duráveis), sua aplicação no desenvolvimento de *software* tem-se também mostrado relevante na literatura (LIU, 2000; SONDA et al., 2000; SENNER; KARSACK, 2010; 2012).

Nesse contexto, o objetivo neste trabalho é demonstrar uma aplicação do QFD no desenvolvimento de *software*, visando a melhorar a compreensão sobre os requisitos dos clientes pelos desenvolvedores. A melhoria da qualidade de um *software* também é decorrente do entendimento sobre o que deve ser desenvolvido, resultando em uma solução que plenamente atenda as expectativas dos clientes. O mercado de *software* possui diversos pontos de vista para o desenvolvimento de produto e, geralmente, esse ponto de vista é o do desenvolvedor do código. Neste caso, o resultado para o cliente, muitas vezes não satisfatório, também é relevante para a organização desenvolvedora.

Além dessa breve introdução, este trabalho apresenta a fundamentação teórica na seção 2, estabelecendo alguns conceitos básicos sobre o

QFD, seguidos pelo desenvolvimento de *software* nesse contexto. Também é realizada uma análise de uma proposta relativamente muito citada na literatura, sobre o uso do Software Quality Function Deployment (SQFD). Em seguida, na seção 3, descrevem-se os métodos empregados no trabalho, visando à aplicação do QFD na remodelagem de um *software* de captura das imagens de equipamentos radiológicos da área médica. A seção 4 apresenta a aplicação do QFD nesta proposta de remodelagem de *software*, caracterizando os resultados do estudo. Por fim, a seção 5 encerra-se com as conclusões e sugestões para continuidade do estudo.

2 Fundamentação teórica

Esta seção trata dos conceitos básicos do Desdobramento da Função da Qualidade e da sua aplicação no desenvolvimento de *software*. São apresentadas as principais fontes bibliográficas utilizadas neste estudo, assim como uma análise de publicações em que se refere o uso do QFD para desenvolvimento de *software*.

2.1 Princípios do desdobramento da função qualidade

O QFD é definido como um método para o desenvolvimento de produtos que busca satisfazer os clientes pela tradução de suas demandas ou requisitos (Customer Requirements – CR) em especificações de projeto (DR) (AKAO, 1990). O método foi originado no final da década de 1960 no Japão, sendo aperfeiçoado nos anos 1970 por Yoji Akao (BÜYÜKÖZKAN et al., 2005), sendo inicialmente planejado para desenvolver um novo produto reunindo as necessidades dos clientes com tabelas de controle da qualidade (ROSSETTO et al., 2010), usadas na produção. O conceito básico do QFD é traduzir as necessidades e requisitos dos

clientes em atributos técnicos (características da qualidade ou especificações de engenharia) e, posteriormente, desdobrando-as em características de componentes, processo, produção, etc. (SENER; KARSAK, 2010).

O QFD usa matrizes para organizar e relacionar conjuntos de dados uns com os outros (LIU, 2000). A matriz mais conhecida é denominada de matriz da qualidade, de acordo com Cheng e Melo Filho (2007), conforme ilustra a Figura 1. Ainda segundo os autores citados, esta matriz é constituída pela relação entre uma tabela de desdobramento da qualidade exigida e uma de desdobramento das características da qualidade do produto final. No entanto, Cheng e Melo Filho (2007) acrescentam que o seu uso não pode ser generalizado para qualquer situação; em alguns casos, sua utilização não é apropriada, como, por exemplo, na situação de uma empresa fornecedora receber do cliente o desenho e as especificações do produto. Neste caso, a matriz da qualidade não seria necessária.

Cheng e Melo Filho (2007) colocam ainda que um dos processos de elaboração da matriz da qualidade ocorre por meio da “extração” das características da qualidade a partir das qualidades exigidas. Na Figura 1, essa extração é apresentada pela seta 1. Após a extração, é realizada a qualidade planejada, em que o cliente é novamente consultado para identificar o grau de importância das qualidades exigidas. A seta 2 da Figura 1 representa a “conversão”, que transmite o grau de importância dos elementos de uma tabela para outra, como, por exemplo, na transmissão da importância atribuída das qualidades exigidas para as características da qualidade.

Cheng e Melo Filho (2007) acrescentam ainda que essa etapa só pode ser realizada após um processo de “correlação” ter sido efetuado. Também colocam que a correlação – apresentada na Figura 1 pelo símbolo 3 – visa a identificar as

relações entre os elementos desdobrados de duas tabelas diferentes. O grau ou a intensidade da correlação recebe valores numéricos, que podem ser representados por símbolos equivalentes a uma relação: “forte”, “média” e “fraca” (CHENG; MELO FILHO, 2007). Os mesmos autores seguem descrevendo que, neste processo de correlação, é importante identificar se houve colunas ou linhas em que não há nenhuma correlação definida. Neste caso, deve ser identificado se algo foi omitido ou se está em excesso.

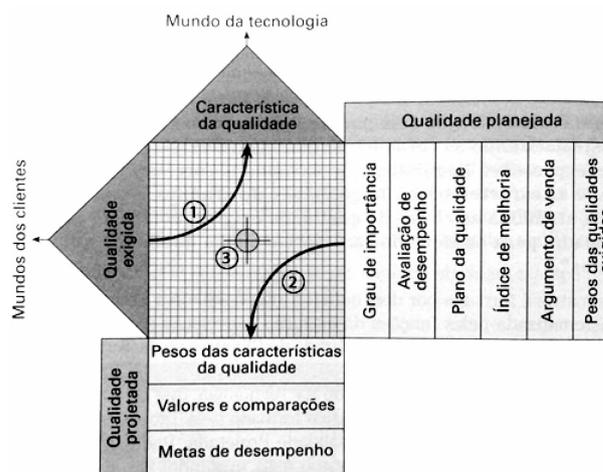


Figura 1: Matriz da qualidade

Fonte: Adaptada de Cheng e Melo Filho (2007, p. 136).

A matriz da qualidade mostrada na Figura 1 é um exemplo da matriz mais comum no QFD. Pode haver, entretanto, mais matrizes que demonstram os passos seguintes para completar o processo de desenvolvimento do produto. Esse conjunto de matrizes e tabelas de um determinado projeto de desenvolvimento é chamado de modelo conceitual (CHENG; MELO FILHO, 2007).

Rossetto et al. (2010) relatam que as aplicações do QFD apresentam uma variedade de benefícios para as organizações e citam, como exemplo, a redução do ciclo de desenvolvimento do produto em um terço, com um aumento correspondente na qualidade do produto. Isso pode ocorrer devido à diminuição do número de alterações de engenha-

ria. A aplicação do QFD pode ser utilizada em diversos setores industriais. Na sequência, o QFD é associado ao desenvolvimento de *software*, objeto de estudo neste trabalho.

2.2 Desdobramento da função da qualidade aplicado a software

O QFD é amplamente utilizado por empresas de manufatura, mas tem sido mais recentemente aplicado ao desenvolvimento de *software* (SENER; KARSAK, 2010). Segundo Sonda et al. (2000), o processo de desenvolvimento de *software* é criativo, não repetitivo. Por esta razão, este não é de fácil controle e, em geral, não possui controles formais. Em relação a esse aspecto, o QFD contribui efetivamente, pois exige a definição e priorização de indicadores que possibilitam o controle do processo, tanto durante o desenvolvimento do *software* como nas fases posteriores de manutenção e apoio ao usuário (SONDA et al., 2000).

Ao identificar claramente o que é mais importante para o cliente e quais as necessidades dos usuários, os esforços dos desenvolvedores se concentram no desenvolvimento do “sistema de *software*”, a fim de maximizar a satisfação do usuário final (KARLSSON, 1997). A análise de requisitos de um sistema do *software* é muitas vezes considerada como um dos passos

mais decisivos no processo do seu desenvolvimento (LIU, 2000), na qual os comandos que descrevem as funções e características do *software* devem ser desenvolvidos e acordados (KARLSSON, 1997).

O autor supracitado comenta também que, embora muitas vezes seja difícil de distinguir, requisitos de *software* são habitualmente divididos em duas categorias distintas: funcionais e não funcionais. Os requisitos funcionais têm como propósito descrever as funções do sistema de *software* (relações entre as entradas e as saídas do sistema de *software*), e os não funcionais visam a descrever os outros aspectos necessários relacionados ao sistema de *software*, o seu desenvolvimento e a sua utilização, como, por exemplo, relativo a aspectos de usabilidade e confiabilidade, descrevendo as características pertinentes para o uso real (KARLSSON, 1997).

Liu (2000) apresenta uma proposta para adaptação do QFD para desenvolvimento de *software*, em que o modelo conceitual proposto é composto pelas matrizes mostradas na Figura 2 (requisitos dos clientes x especificações técnicas, especificações técnicas x características do projeto, características do projeto x estratégia de implementação, e implementação estratégica x estratégia de teste).

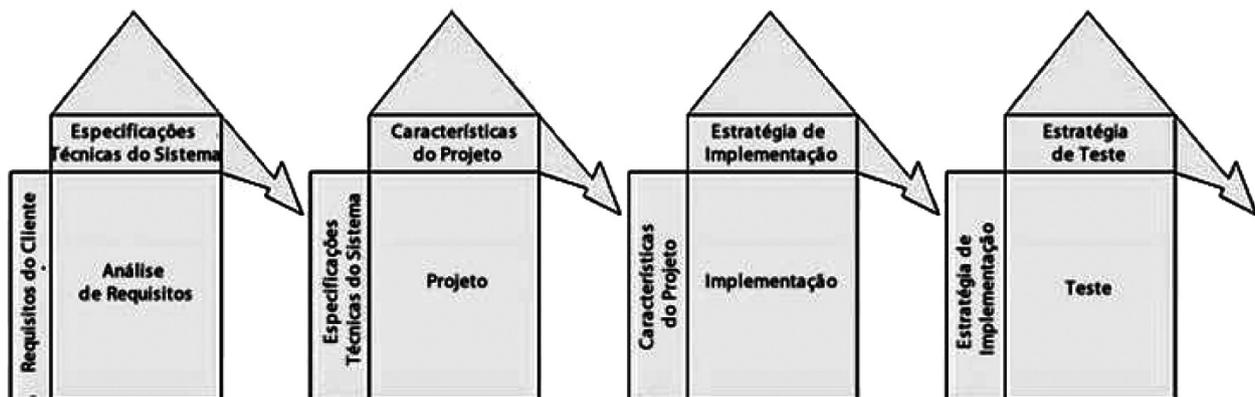


Figura 2: Software Quality Function Deployment (SQFD)
Fonte: Adaptada de Liu (2000).

Elboushi e Sherif (1997) indiretamente confirmam a proposta de modelo conceitual apresentada por Liu (2000) quando comentam que, com a utilização do QFD, foi possível capturar e trazer a voz dos clientes para a organização, a partir da definição funcional de requisitos, especificações, etc. Os autores anteriormente citados também apresentam algumas vantagens identificadas com o uso do QFD, tais como:

- A voz do cliente é utilizada para compreender as necessidades deste.
- A documentação de requisitos é aprimorada pelos registros em tabelas e matrizes.
- O planejamento se torna mais específico, a formação de consenso se torna mais fácil e o tempo de projeto é reduzido.
- Os pontos de controle são esclarecidos e a duplicação de esforços é significativamente reduzida.
- O número de gargalos é reduzido e há menos mudanças de projeto no final do desenvolvimento.

Cabe destacar que a adaptação realizada por Liu (2000) segue praticamente o mesmo padrão da utilização do QFD tradicional. Cheng e Melo Filho (2007) apresentam um modelo conceitual do QFD para processos digitais (Figura 3), em que a primeira matriz desenvolvida, estabelece a relação entre os requisitos dos clientes (qualidades exigidas) e funções do produto. Além das tabelas mostradas na Figura 3, Cheng e Melo Filho (2007) sugerem outras que podem ser utilizadas para gerar o modelo conceitual, tais como: (i) tabela de especificação de requisitos de usabilidade, (ii) tabela dos subsistemas; e (iii) tabelas de recursos.

Na busca bibliográfica realizada, não foram encontradas outras propostas de modelos conceituais ou adaptações com um maior detalhamento das matrizes para o uso do QFD no desenvolvi-

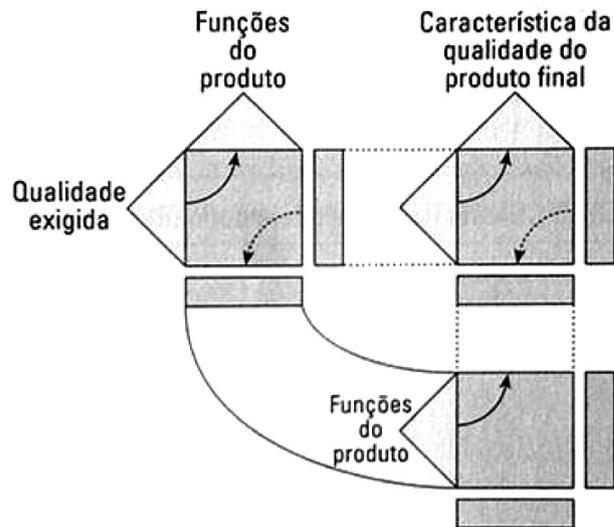


Figura 3: Exemplo de modelo conceitual para software

Fonte: Cheng e Melo Filho (2007, p. 203).

mento de *software*. Além das publicações já citadas, foi identificado o trabalho de Fernandes (2008), que apresenta um modelo conceitual para desenvolvimento de *software* composto por seis matrizes. A Figura 4 mostra essas matrizes e seus relacionamentos, em que o modelo conceitual é formado pelas seguintes matrizes:

- Desdobramento de necessidades do cliente, em funções do *software* e em requisitos do cliente.
- Desdobramento dos requisitos do cliente em requisitos do *software*.
- Desdobramento dos requisitos de *software* em requisitos dos subsistemas (ou definição da arquitetura).
- Desdobramento dos requisitos do *software* em possíveis riscos.
- Desdobramento dos requisitos do *software* em atividades dos processos.
- Desdobramento das atividades em requisitos dos recursos.

Nas adaptações encontradas na revisão da literatura para a utilização do QFD no desenvolvi-

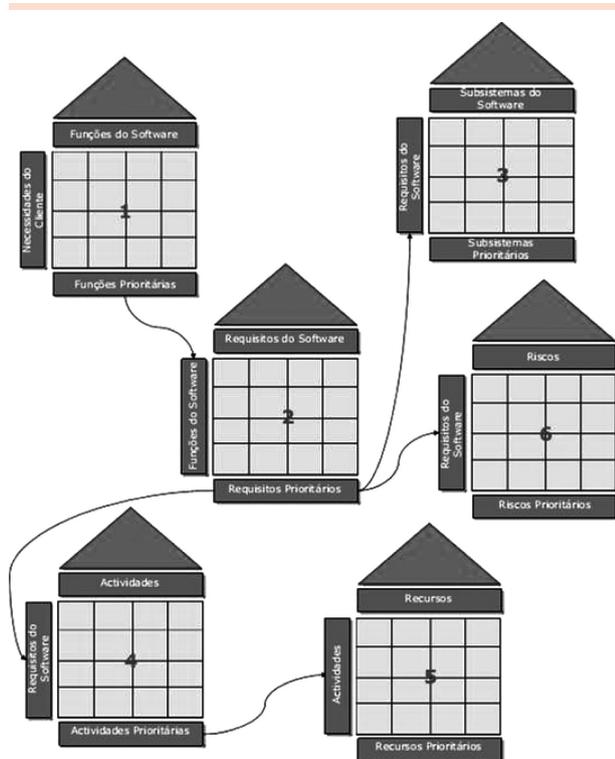


Figura 4: Modelo QFD para software
 Fonte: Fernandes (2008), p. 60.

vimento de *software*, utiliza-se a mesma base para as aplicações industriais tradicionais, conforme ilustrado nas Figuras 2, 3 e 4. Como já citado, existem adaptações que se adéquam ao processo de desenvolvimento de *software*, mas não foi identificada na literatura uma nova proposta.

2.3 Identificação e análise de trabalhos sobre a aplicação do QFD para software

Para Haag et al. (1996), a adaptação do QFD para desenvolvimento de *software* pode ser chamada de Software Quality Function Deployment (SQFD). Sener e Karsak (2010) afirmam que o SQFD é uma metodologia estruturada que se concentra na melhoria da qualidade do processo de desenvolvimento de *software* para criar produtos que atendam às expectativas dos consumidores por meio da implementação da “voz do cliente” ao longo do processo de desenvolvimento. Os autores

citados também comentam que o SQFD é quantificável e define os requisitos críticos dos clientes. O SQFD é uma adaptação da casa da qualidade, a matriz mais comumente utilizada no QFD tradicional (HAAG et al., 1996).

Neste estudo, foi identificado que a utilização do QFD para desenvolvimento de *software* tem sido citada em alguns trabalhos, tais como na proposta do SQFD. Das publicações verificadas, sete apresentam ou citam o SQFD como sendo uma nova metodologia ou adaptação de metodologia existente para *software*, advinda do QFD tradicional. O Quadro 1 lista algumas dessas publicações sobre este tema, incluindo uma síntese de seus objetivos.

Durante a análise da literatura apresentada no Quadro 1, foi verificado que o SQFD é colocado como sendo uma adaptação do QFD tradicional. Porém, não é claro quais modificações são realizadas a ponto de este tornar-se diferenciado. Os autores citados no Quadro 1 mostram que o SQFD é aplicado e adaptado ao desenvolvimento de *software*; contudo, essa aplicação corresponde ao QFD tradicional, não oferecendo uma nova proposta específica para este setor. No caso, o SQFD torna-se apenas um nome que difere do QFD e, não necessariamente, uma nova proposta, sob o ponto de vista metodológico e estrutural. Como não há uma caracterização clara sobre o SQFD, os conceitos se misturam nos textos, sendo citados como QFD ou como SQFD, principalmente pelo fato de não haver uma diferenciação específica entre ambos. Além da distinção dos conceitos não ser clara, verifica-se que não é apresentado nenhum modelo conceitual em que se argumente e defenda-se a diferenciação entre a proposta do SQFD e o QFD tradicional. Também não existem exemplos de aplicações que justifiquem ou demonstrem a diferenciação de QFD para SQFD como metodologias distintas. Assim, o SQFD é simplesmente uma variante do QFD típico, mas adequado sob o ponto de vista de aplicação.

Referências	Objetivos
Haag et al. (1996)	Relatar os resultados de uma pesquisa de principais empresas fornecedoras de <i>software</i> que estão usando SQFD como uma abordagem para a melhoria da qualidade de desenvolvimento de <i>software</i> .
Liu (2000)	Apresentar a matriz da qualidade e mostrar alguns passos em que o QFD pode auxiliar na melhoria da qualidade do <i>software</i> e na construção desses produtos com alta satisfação dos clientes.
Pai (2002)	Propor o aprimoramento do processo de integração entre SQFD e Goal-Question-Metrics e como este modelo de integração o processo de desenvolvimento de <i>software</i> pode aprimorar a qualidade do próprio <i>software</i> .
Liu et al. (2006)	Apresentar um método quantitativo da fixação de objetivos técnicos em SQFD para permitir a análise de impacto de valores-alvo não alcançados na satisfação do cliente. É baseado na avaliação do impacto dos atributos técnicos sobre a satisfação das necessidades dos clientes.
Rossetto et al. (2010)	Reforçar a questão que a qualidade assegurada deve ser tratada aproveitando-se de princípios e modelos disponibilizados pela teoria de esquemas de controle, juntamente com o QFD e as melhores práticas de modelagem de <i>software</i> baseado em Unified Modeling Language (UML).
Sener e Karsak (2010)	Demonstrar uma regressão <i>fuzzy</i> e abordagem de otimização para determinar os níveis-alvo em SQFD. É desenvolvido um modelo de programação matemática para determinar os níveis-alvo de atributos técnicos, com as relações funcionais obtidos por regressão <i>fuzzy</i> . Um problema de motor de busca visando à melhoria da qualidade é apresentado para ilustrar a aplicação da abordagem proposta.
Sener e Karsak (2012)	Propor uma regressão <i>fuzzy</i> estrutura de decisão em que considera as necessidades futuras para definir níveis-alvo de atributos técnicos do QFD para <i>software</i> . A proposta permite calcular o grau de importância integrado para cada necessidade dos clientes, combinando a voz atual com a voz futura destes.

Quadro 1: Publicações sobre SQFD

Fonte: Elaborado pelos autores com base na busca e análise da literatura.

Apesar de Haag et al. (1996) serem citados nas publicações sobre SQFD, seu trabalho não foi aprofundado nas obras pesquisadas, visando a obter um melhor detalhamento ou exemplificação da adaptação proposta. As sugestões de utilização do SQFD são para transformar a voz do cliente em requisitos de *software* e para uso da “casa da qualidade”. Ambas as propostas citadas são utilizadas no QFD tradicional para aplicações em produtos. Dessa forma, a adaptação sugerida pelos autores não difere do modelo padrão proposto pelo QFD para uso em produtos, e sim faz uma adequação para sua utilização no desenvolvimento de *software* que melhor atenda ao processo de desenvolvimento deste. Devido a tais fatores, neste trabalho, propõe-se o uso do QFD adaptado para desenvolvimento de *software*, aplicando-se os mesmos conceitos já expostos na literatura. Essa decisão também se deve ao fato de que o propósito deste trabalho é de uma aplicação empírica do QFD, baseada na literatura, e não o desenvolvimento de uma nova proposta conceitual.

2.4 Síntese das características e resultados encontrados a partir da análise da literatura

Por meio de uma análise da literatura disponível, foi verificado que os pesquisadores buscam aprimorar a utilização do QFD para o desenvolvimento de *software*, seja por meio de modelos de processos, melhores práticas ou análise matemática, como ilustra os trabalhos listados no Quadro 1. O foco de todas as análises é o cliente, sendo, portanto, estabelecido como o *software* pode ser melhorado para a satisfação deste. As contribuições do QFD para o desenvolvimento de *software* envolvem: a interação com a equipe de desenvolvimento e todos os envolvidos no projeto, o impedimento da perda de informações, o fornecimento de justificativas mensuráveis nas tomadas de decisões e a geração de uma melhor comunicação interdepartamental.

O desenvolvimento de um *software* carece da análise das necessidades dos clientes e, neste sentido, tal análise dá suporte a esse desenvolvimento. Segundo Haag (1996), uma das dificuldades no desenvolvimento de *software* é a escolha incorreta dos requisitos, ou seja, o programa é desenvolvido sem o entendimento das reais necessidades dos usuários, o que, posteriormente, gera custos altos para refazer os requisitos indispensáveis não avaliados na sua concepção. Para o desenvolvimento de um programa, é preciso que suas funções atendam as expectativas do cliente. Pai (2002) defende ainda que a qualidade do processo de desenvolvimento do *software* determina a qualidade deste. Um dos pontos importantes da utilização do QFD é, além de atender os requisitos do cliente, atingir os objetivos do projeto com a qualidade requerida, mesmo que a utilização do método seja apenas uma parte no ciclo de vida do desenvolvimento do produto.

Algumas lacunas a serem investigadas em trabalhos futuros compreendem as aplicações do QFD no ciclo de vida dos produtos buscando novas possibilidades de matrizes, bem como investigações sobre o uso da matriz de funções do produto para a concepção deste. As empresas de *software* trabalham com metodologias ágeis e uma linha de estudo também seria a de verificar o impacto da utilização do QFD nesse processo. Outro ponto a ser estudado é o uso do QFD para verificar novas tendências de mercado, por meio de levantamentos e, como decorrência, oferecer soluções inovadoras e desejadas pelos clientes. A partir de um dos princípios do QFD, de ouvir a voz do cliente e ter percepção de suas necessidades, a criação do produto pode tornar-se mais assertiva e, possivelmente, com mais inserção no mercado.

Neste trabalho, busca-se contribuir com a possibilidade da utilização do QFD no desenvolvimento de *software*, visando a uma adequação do produto às necessidades do cliente, associando-as às funções do produto/*software*. Objetiva-se tam-

bém destacar a utilização do método por uma equipe de desenvolvimento, enfatizando as funções do *software*. Como apresentado mais adiante, o QFD é usado em uma parte do ciclo de vida do desenvolvimento do *software*, o que auxilia na qualidade do produto final.

Na sequência, são apresentados os procedimentos de pesquisa utilizados para viabilizar metodologicamente o atual estudo.

3 Métodos de pesquisa

Este trabalho apresenta a aplicação do QFD na remodelagem de um *software* de captura de imagens. Seu propósito principal é mostrar ainda aspectos associados à priorização no desenvolvimento de um *software* com e sem a utilização do QFD. Foi verificado que a empresa desenvolvedora estava projetando um novo programa com a mesma função de um já existente. Na verdade, foi necessária uma remodelagem, pois o existente não atendia os requisitos de um cliente importante. O novo *software*, que neste trabalho é chamado simbolicamente de produto X-Capture, foi baseado em solicitações coletadas no cliente e no produto concorrente utilizado por ele.

Inicialmente, foi feita uma busca na literatura, nas seguintes bases de dados: ISI Web of Knowledge, Scopus, Scielo, Science Direct e IEEEExplore. As palavras-chave utilizadas para a busca foram: QFD, *software development*, *quality function deployment*, e *software*, com a busca em títulos, resumos e palavras-chave das publicações, resultando em um conjunto de artigos. Após a retirada das publicações duplicadas e a análise do conteúdo do resumo das restantes, foram identificados 34 artigos que possuíam relação com a utilização do QFD no desenvolvimento de *software*. Dentre esses textos, somente trabalhos que adotavam o QFD, em específico no desenvolvi-

mento de *software*, foram identificados, tal como o SQFD, anteriormente discutido.

Posteriormente, foram coletados dados e informações de como o *software* a ser desenvolvido funcionaria. Para isso, foi necessária a participação em reuniões da equipe da área de desenvolvimento. Em uma dessas reuniões, um dos autores do atual estudo propôs a utilização do QFD com o objetivo de comparar os resultados de priorização de desenvolvimento para a equipe de desenvolvimento e para a gerência de produto. Em termos de escopo, a proposta foi a de “[...] aplicar o QFD relacionando às tabelas de requisitos de clientes e funções do *software*, formando a matriz inicial [...]”. Uma das matrizes mais relevantes, definida pela equipe, foi a matriz de funções do *software*. Para o desenvolvimento dessa matriz do QFD, foram propostas quatro reuniões de trabalho. A equipe foi composta pelo coordenador da equipe de desenvolvimento de *software* (setor de pesquisa e desenvolvimento), gerente de produto (setor de *marketing*), coordenadora da equipe de teste (setor de pesquisa e desenvolvimento) e coordenadora da equipe de implantação do sistema (setor de serviços). Em cada uma das reuniões, foi desenvolvida uma parte da matriz, iniciando-se pela tabela de requisitos dos clientes, funções do *software*, qualidade planejada e qualidade projetada que são apresentadas a seguir no detalhamento da aplicação. Visando a descrever melhor a aplicação realizada, a seguir, um complemento dos procedimentos metodológicos é apresentado juntamente com os resultados.

4 Aplicação do QFD para remodelar um *software* de captura

Conforme mencionado anteriormente, o *software* X-Capture foi desenvolvido para atender uma demanda específica de um cliente potencial,

contudo, este também pode atender outros clientes. O desenvolvimento deste programa foi baseado no fluxo de trabalho atual do cliente e nas funcionalidades que ele já utiliza em um único *software*.

Inicialmente, foi realizada uma reunião para uma breve apresentação do QFD, obtendo-se boa receptividade da equipe, com interesse em participar e verificar os resultados que o método poderia trazer. Nesta reunião, foram solicitados os requisitos do cliente para o desenvolvimento do *software*, e o gerente de produto apresentou-os já especificados em itens desdobrados. Estes itens foram solicitados já definidos como a “voz do cliente”, ou seja, solicitações na forma expressada pelo cliente, que, porém, ainda não haviam sido documentadas. Devido a isso, a tabela inicial de requisitos do cliente foi composta pelos existentes. Estes foram revisados e verificados para confirmar se possuíam todas as solicitações principais acordadas pelo gerente de produto. Os requisitos foram reescritos para se adequar a linguagem do QFD, como, por exemplo: o requisito “anotação em imagens” foi alterado para “fazer anotações nas imagens: seta, texto, elipse” (ver Quadro 2). A partir dessa decisão, definiram-se os requisitos do cliente, mostrados no Quadro 2, ou seja, foi possível construir a primeira tabela. Ficou acordado que as funções deveriam ser revistas na reunião seguinte, momento em que seriam apresentadas as funções já existentes no projeto de desenvolvimento em andamento.

Na segunda reunião, foram apresentadas e discutidas as funções existentes. Inicialmente, foram apresentadas as funções principais, tais como: (i) cadastros de dados do conjunto de imagens, (ii) captura de imagens e vídeos, (iii) análise de imagens e emissão de texto, (iv) impressão e, por último, (v) finalização. Em seguida, definiu-se a função principal do *software* e, posteriormente, foram adicionadas as funções secundárias, necessárias para este. A função principal do X-Capture ficou

RC: Requisitos dos Clientes	Ser integrado ao servidor por meio de controle de usuários
	Ser multilíngue: português, inglês e espanhol
	Enviar conjunto de imagens capturadas e texto para o servidor
	Fazer <i>download</i> de um conjunto de imagens
	Continuar com a ação executante quando reaberto, caso o <i>software</i> feche forçosamente
	Integrar automática ou manualmente com a lista de tarefas atual ou pré-existente
	Cadastrar manualmente caso não exista uma lista de tarefas
	Cadastrar convênios
	Capturar imagens a partir de sinal de vídeo
	Converter as imagens capturadas no formato X em imagens no formato Y
	Converter os vídeos capturados em formato Y <i>multiframe</i>
	Fazer anotações nas imagens: seta, texto e elipse
	Ajustar cor, brilho, gama e contraste nas imagens capturadas
	Possuir um editor de texto com <i>templates</i> pré-definidos
	Realizar cálculos específicos dentro do editor de texto
	Ter impressão do texto e imagens
	Poder configurar <i>layouts</i> de impressão

Quadro 2: Requisitos dos clientes

Fonte: Elaborado pelos autores no desenvolvimento realizado.

definida como: “capturar e converter imagens do formato da fonte de entrada”, denominado neste estudo em imagens no formato de destino, aqui chamado de Y. No Quadro 3, são listadas todas as funções discutidas e definidas para o X-Capture.

A discussão nessa segunda reunião foi a mais demorada, pois nela debateu-se sobre quais seriam as funções secundárias que geraram opiniões contraditórias entre os participantes da equipe. No entanto, foi possível chegar a um consenso sem que o mediador precisasse intervir mais fortemente. Em seguida, na terceira reunião, foi apresentada e distribuída aos participantes a matriz das funções com as tabelas dos requisitos do cliente, tabela de funções do *software* e a qualidade planejada. Nessa etapa, o objetivo foi definir o grau de importância dos requisitos do cliente para ele

Funções do Software	Capturar e converter imagens no formato X em imagens no formato Y
	Capturar vídeo
	Capturar vídeo com múltiplas entradas
	Visualizar e editar imagens em formato Y
	Permitir edição dos vídeos capturados
	Permitir texto referente a um conjunto de imagens formato Y
	Permitir a impressão de imagens formato Y e textos
	Permitir a gravação de CD das imagens formato Y e textos
	Cadastrar dados manualmente ou usar os dados existentes no servidor
	Disponibilizar tabelas de cálculos específicos dentro dos textos
	Possibilitar busca e recuperação das imagens no servidor
	Enviar para o servidor
	Buscar dados anteriores no servidor
	Buscar lista de tarefas

Quadro 3: Funções do software

Fonte: Elaborado pelos autores no desenvolvimento realizado.

mesmo, realizar a análise comparativa do produto atual (que não o atende) e do produto concorrente em relação aos requisitos, realizar o planejamento da qualidade e fazer a análise do argumento de vendas também em relação aos requisitos do cliente. A Figura 5 apresenta os dados anteriormente citados, no que se refere à qualidade planejada. A identificação do grau de importância atribuída pelo cliente aos requisitos já havia sido realizada em visita a ele efetuada pelo gerente de produto. Como na aplicação do QFD, os requisitos não foram alterados, mas mudou-se somente a forma de escrevê-los, foi então utilizado o mesmo grau de importância, disponibilizado pelo cliente, tendo sido coletados pelo gerente de produto.

Conforme apresentado na Figura 5, nessa aplicação, foi definido o grau de importância numa escala de 1 a 5, sendo 1 atribuído a requisitos “pouco importante”, aumentando até 5, definido para os “muito importante”. Como prevê o QFD (CHENG; MELO FILHO, 2007), nessa etapa, foi também realizada uma análise compa-

rativa, em que o produto atual da empresa e o do concorrente são avaliados para verificar sua *performance* diante dos requisitos do cliente. Para esta avaliação também foram atribuídos valores numa escala de 1 a 5, sendo 1 atribuído ao menor desempenho do produto, aumentando até chegar a 5, em que este valor é definido para o melhor desempenho do produto.

Baseado nas atribuições comparativas foi possível então planejar o grau de melhoria que se desejava alcançar, levando em consideração a capacidade produtiva e o planejamento estratégico da empresa. A qualidade planejada também utiliza uma escala de 1 a 5, da mesma forma que no grau de importância e na análise comparativa. Outro item avaliado foi relativo aos argumentos de venda, que são avaliados pelos requisitos identificados como mais interessantes sob o ponto de vista mercadológico e também considerados estrategicamente relevantes pela empresa. A Tabela 1 mostra a classificação do argumento de vendas, em que os valores são representados por símbolos.

Tabela 1: Símbolos e valores dos argumentos de venda

Argumento de vendas		
⊙	Especial	1,5
⊕	Comum	1,2
Em branco	Óbvio	1

Fonte: Adaptado de Cheng e Melo Filho (2007, p. 127).

Os valores definidos foram transferidos para uma planilha eletrônica, sendo calculados a taxa de melhoria, o peso absoluto e o peso relativo. A taxa de melhoria pode ser obtida por meio de avaliação da qualidade planejada dividida pelo nível atual do produto já existente. O peso absoluto é calculado pela multiplicação da taxa de melhoria pelo grau de importância e pelo argumento de vendas. Já o peso relativo calcula-se pelo peso absoluto dividido pela soma de todos os pesos ab-

solutos e depois multiplica-se por 100 para obter o valor percentual. A Figura 5 mostra a qualidade planejada e o resultado dos cálculos já gerados na planilha eletrônica. Com isso a qualidade planejada, apresentada na Figura 5, foi concluída.

Por fim, a quarta e última reunião para a aplicação do QFD foi a mais longa e com uma quantidade maior de discussão. Nessa etapa, são definidos os relacionamentos ou correlações das tabelas de requisitos dos clientes com as funções do *software*. Antes de realizar as correlações foram apresentadas para a equipe as intenções a serem utilizadas, conforme é mostrado na Tabela 2.

Tabela 2: Símbolos e valores de correlação

Correlação		
⊙	Forte	9
○	Moderado	3
△	Fraco	1
Em branco	Inexistente	-

Fonte: Adaptado de Cheng e Melo Filho (2007, p. 150).

Também foi explicado para a equipe que as correlações deveriam ser realizadas bem como analisado o grau de influência de cada uma das funções em relação aos requisitos do cliente, ou seja, como as funções afetam os requisitos. A ordem de análise das correlações foi realizada preenchendo-se as colunas de cima para baixo e da esquerda para a direita. Nesse aspecto, o início do preenchimento das correlações na matriz trouxe dificuldades de entendimento de como deveria ser feita a pergunta para avaliar este relacionamento. Houve uma forte tendência de inversão, pois se perguntava com que intensidade o requisito do cliente afetava a função, quando na verdade deveria ser o inverso. Quando essa situação ocorria era necessário interromper a discussão gerada e retomar a explicação de que a análise de intensidade da correlação deveria ser

		Qualidade Planejada								
		Grau de Importância	Análise Comparativa		Planejamento da Qualidade			Peso		
			Nosso Produto	Concorrente	Qualidade Planejada	Taxa de Melhoria	Argumento de Vendas	Peso Absoluto das Qualidades	Peso Relativo das Qualidades (%)	
RC: Requisitos dos Clientes	Ser integrado ao servidor através de controle de usuários	5	1	1	5	5	⊙	37,5	13,98	
	Ser multilíngue: português, inglês e espanhol	2	1	1	2	2	⊙	6	2,24	
	Enviar exames e textos para o servidor	5	2	1	5	2,5	⊕	15	5,59	
	Fazer <i>download</i> de um conjunto de imagens	3	1	5	3	3		9	3,36	
	Continuar com a ação executante quando reaberto, caso o <i>software</i> feche forçosamente	5	1	3	5	5		25	9,32	
	Integrar automaticamente ou manualmente com a lista de tarefas atual ou pré-existente	3	3	1	3	1	⊙	4,5	1,68	
	Cadastrar manualmente caso não tenha lista de tarefas	5	5	5	5	1	⊕	6	2,24	
	Cadastrar convênios	4	1	4	4	4		16	5,97	
	Capturar imagens a partir de sinal de vídeo	5	5	5	5	1		5	1,86	
	Converter as imagens capturadas no formato X em imagens no formato Y	5	5	2	5	1	⊕	6	2,24	
	Converter os vídeos capturados em formato Y <i>multiframe</i>	5	1	3	5	5	⊙	37,5	13,98	
	Fazer anotações nas imagens: seta, texto e elipse	2	1	4	2	2	⊕	4,8	1,79	
	Ajustar cor, brilho, gama e contraste nas imagens capturadas	2	1	3	2	2	⊕	4,8	1,79	
	Possuir um editor de textos com templates pré-definidos	5	1	4	5	5	⊙	37,5	13,98	
	Realizar cálculos específicos dentro do editor de texto	4	1	4	4	4	⊙	24	8,95	
	Ter impressão de textos e imagens	5	2	4	5	2,5	⊙	18,8	6,99	
	Poder configurar <i>layouts</i> de impressão	3	1	4	3	3	⊕	10,8	4,03	

Figura 5: Qualidade planejada

Fonte: Desenvolvida pela equipe de desenvolvimento com base na proposta de trabalho.

determinada indagando-se como a função deveria afetar o requisito do cliente, em uma relação de causa e efeito. Em outras palavras, questiona-se se a função (causa) “mede” determinado requisito (efeito), e com que intensidade. Em suma, as funções do *software* deveriam “atender” os requisitos dos clientes.

Outro ponto importante na definição das correlações foi a tendência da equipe de atribuir uma forte intensidade de correlação para vários requisitos. Em um primeiro momento, as correlações foram realizadas de forma mais livre, ou seja, a equipe atribuiu as intensidades de forma rápida, com algumas discussões em alguns itens, porém

sem maiores problemas. Isso resultou numa matriz com muitos valores de intensidade “forte” e “moderada”. Num segundo momento, foram feitos novos questionamentos sobre as intensidades, instigando a equipe a refletir e responder novamente sobre as intensidades, como, por exemplo: “Realmente esta função afeta este requisito com intensidade forte?”. Assim, foram feitas alterações de intensidades, sendo algumas até eliminadas. Este exercício também foi importante para ressaltar a importância do mediador e do conhecimento necessário sobre os conceitos sobre o QFD. Quando a matriz foi reavaliada houve novas discussões e argumentações sobre as intensidades. Isso trouxe questionamentos sobre as próprias funções do *software*. No entanto, nada, além das intensidades, foi feito.

A Figura 6 mostra as correlações na matriz e os resultados dos cálculos que possibilitam o término da sua construção, ou seja, os cálculos referentes à qualidade projetada. Na figura também pode ser vista a ordem de prioridade do desenvolvimento de funções.

4.1 Discussão sobre a matriz de funções e priorização

A matriz de funções mostrada na Figura 6 corresponde ao primeiro passo realizado para o desenvolvimento do *software* mencionado. Para o desenvolvimento de qualquer produto, é necessário fazer uma análise funcional, isto é, é preciso definir a função principal e as funções secundárias que o produto deve cumprir para atender os requisitos desejados pelo consumidor. Esta análise é ainda mais importante no caso de desenvolvimento de *software*. A construção da matriz mostrada nessa figura possibilita ter uma espécie de “mapeamento” de quais requisitos dos clientes estão associados às funções previstas pelo *software*.

Em outras palavras, a definição das funções é imprescindível para que cada um dos requisitos

seja atendido, em maior ou menor intensidade. Esta avaliação é mostrada nas células centrais da matriz que mostram o grau de intensidade dessa relação (“forte”, “médio”, “fraco”, ou inexistente, quando em branco). A partir dos cálculos mostrados na “qualidade projetada” (linha horizontal com os números dispostos nas células abaixo das correlações), é possível identificar qual a função mais importante e, assim, centrar esforços no desenvolvimento desta e de outras funções mais relevantes. No caso da matriz na Figura 6, a função do *software* “permitir texto referente a um conjunto de imagens formato Y” é a que resultou em maior peso relativo (15,98%) e, assim, é esta uma das funções que deve ser priorizada e receber investimentos para seu desenvolvimento. A fase seguinte (não mostrada no atual trabalho) seria desenvolver as especificações do *software* (linhas de código, etc.) que atendam esta e outras funções que devem ser prioritárias.

A Tabela 3 apresenta a comparação entre a priorização do desenvolvimento das funções com a aplicação do QFD e a priorização sem o QFD, que já havia sido realizada antes desta aplicação do QFD. Na priorização com o QFD, a apresentação é sequencial, já a dada sem o QFD é definida por grupos de funções que possuem a prioridade de desenvolvimento, ou seja, no QFD é mais detalhada e, principalmente, vinculada aos requisitos dos clientes. Desta forma, a aplicação do QFD também foi relevante, pois, além de priorizar a partir da posição do cliente, ainda organiza de forma sequencial o desenvolvimento do produto.

Após a finalização da matriz mostrada na Figura 6, foram levantados alguns pontos positivos e negativos sobre o processo de aplicação. A equipe envolvida com esta aplicação do QFD citou como pontos negativos: (i) a falta de conhecimento sobre o tema (QFD), (ii) a percepção de que a avaliação de prioridades baseadas no cliente afeta a prioridade de itens importantes

Tabela 3: Comparação de prioridades com e sem o uso do QFD

Funções	Priorização sem QFD	Priorização com QFD
Capturar e converter imagens no formato X em imagens no formato Y	1	4
Capturar vídeo	1	3
Capturar vídeo com múltiplas entradas	6	2
Visualizar e editar imagens em formato Y	2	10
Permitir edição dos vídeos capturados	4	11
Permitir texto referente a um conjunto de imagens formato Y	1	1
Permitir a impressão de imagens formato Y e textos	1	6
Permitir a gravação de CD das imagens formato Y e textos	5	14
Cadastrar dados manualmente ou usar os dados existentes no servidor	1	8
Disponibilizar tabelas de cálculos específicos dentro dos textos	3	5
Possibilitar busca e recuperação das imagens no servidor	2	9
Enviar para o servidor	1	7
Buscar dados anteriores no servidor	3	12
Buscar lista de tarefas	1	13

Fonte: Elaborada pelos autores com base no desenvolvimento realizado.

ções envolvendo diversos pontos de vista, e (iii) a obtenção de uma visão efetiva do que o cliente realmente está solicitando. Por fim, as conclusões extraídas a partir do estudo desenvolvido são apresentadas na seção seguinte.

5 Conclusões

Neste trabalho, teve-se como objetivo demonstrar uma aplicação do QFD no desenvolvimento de *software*, com o intuito de melhorar

a compreensão dos requisitos dos clientes pelos engenheiros e desenvolvedores envolvidos nesse processo. Com esta aplicação do QFD no desenvolvimento do *software* é possível concluir que seu uso é adaptável ao programa, como um método de suporte ao seu desenvolvimento. Assim, foi possível constatar a necessidade da utilização do QFD e a importância de gerar um modelo conceitual (não explorado neste trabalho) que auxilie no desenvolvimento de um *software*. O QFD apresenta, entretanto, pontos negativos que foram levantados nesta aplicação, como, por exemplo, o tempo necessário que o método demanda para as discussões, o qual remete a necessidade de um mediador que deve estar atuante nas reuniões.

Após análise dos resultados, concluiu-se também que os pontos positivos se sobressaíram em relação aos negativos. Em relação aos pontos positivos, destaca-se que é relevante construir e manter uma documentação e um histórico do trabalho realizado com o QFD, o auxílio que o método proporciona na tomada de decisão e a comunicação explícita sobre os requisitos do cliente. Vale destacar que trabalhar com equipes multidisciplinares, possibilita a apresentação de pontos de vista diferentes que contribuem para alcançar os objetivos de um projeto de *software*, tornando-o mais focado e eficaz. Para o desenvolvimento do *software* em questão, o QFD proporcionou um suporte significativo, estabelecendo funções extritamente vinculadas aos requisitos do cliente, gerando uma contribuição prática para a equipe de desenvolvimento e para a empresa. Como continuidade deste trabalho, sugere-se o desenvolvimento das outras matrizes que poderão compor o modelo conceitual (conjunto de matrizes), para conclusão do desenvolvimento do *software*. Salienta-se, entretanto, a contribuição deste trabalho ao apresentar o desenvolvimento da matriz de funções, que é pouco comum na literatura.

Referências

AKAO, Y. *Quality function deployment: integrating customer requirements into product design*. Portland: Productivity Press, 1990.

BÜYÜKÖZKAN, G.; FEYZIOĞLU, O.; FEYZIOĞLU, O. Group decision making to better respond customer needs in software development. *Computers and Industrial Engineering*, v. 48, n. 2, p. 427-441, 2005.

CHENG, L. C.; MELO FILHO, L. D. R. *QFD: desdobramento da função qualidade na gestão de desenvolvimento de produtos*. São Paulo, Blucher, 2007.

ELBOUSHI, M. I.; SHERIF, J. S. Object-oriented software design utilizing Quality Function Deployment. *Journal of Systems and Software*, v. 38, n. 2, p. 133-143, 1997.

FERNANDES, M. F. *Gestão de projectos de software com QFD e ABC*. 2008. 137 f. Dissertação (Mestrado em Sistemas de Informação)–Escola de Engenharia, Universidade do Minho, Minho, 2008.

HAAG, S.; RAJA, M. K.; SCHKADE, L. L. Quality function deployment usage in software development. *Communications of the ACM*, v. 39, n. 1, p. 41-49, 1996.

KARLSSON, J. Managing software requirements using quality function deployment. *Software Quality Journal*, v. 6, n. 4, p. 311-325, 1997.

LIU, F. et al. A quantitative approach for setting technical targets based on impact analysis in software quality function deployment (SQFD). *Software Quality Journal*, v. 14, n. 2, p. 113-134, 2006.

LIU, X.F. Software quality function deployment. *IEEE Potentials*, v. 19, n. 5, p. 14-16, 2000.

PAI, W.C. A quality-enhancing software function deployment model. *Information Systems Management*, v. 19, n. 3, p. 20-24, 2002.

ROSSETTO, S.; FRANCESCHINI, F.; DEMARTINI, C. A QFD framework for quality, innovation and high-tech product development dynamics. *Journal of Beijing Institute of Technology (English Edition)*, v. 19, n. 1, p. 74-82, 2010.

SENER, Z.; KARSAK, E. E. A decision model for setting target levels in software quality function deployment to respond to rapidly changing customer needs. *Concurrent Engineering*, v. 20, n. 1, p. 19-29, 2012.

SENER, Z.; KARSAK, E. E. A fuzzy regression and optimization approach for setting target levels in software quality function deployment. *Software Quality Journal*, v. 18, n. 3, p. 323-339, 2010.

SONDA, F. A.; RIBEIRO, J. L. D.; ECHEVESTE, M. E. A aplicação do QFD no desenvolvimento de software: um estudo de caso. *Produção*, v. 10, n. 1, p. 51-75, 2000.

Recebido em 7 jun. 2016 / aprovado em 8 set. 2016

Para referenciar este texto

SCALVENZI, L.; CAUCHICK MIGUEL, P. A. QFD aplicado ao desenvolvimento de software: priorização de requisitos do cliente em uma matriz de funções. *Exacta – EP*, São Paulo, v. 14, n. 4, p. 677-692, 2016.