

# Priorização de Testes de Software: Uma Abordagem Orientada ao Cliente

Káthia Marçal de Oliveira  
Fabiano Mariath D'Oliveira  
Adelaide Figueiredo  
Universidade Católica de Brasília

## Resumo

A atividade de teste de software possui importância fundamental no ciclo de desenvolvimento de sistemas, podendo ter um custo de até cinco vezes mais que todas as outras atividades juntas. Vários fatores como a diversidade de técnicas, o tempo destinado a essa atividade e a necessidade de atender a requisitos do cliente; devem ser analisados em conjunto de forma a realizar testes de forma mais eficaz. Este artigo propõe uma abordagem de priorização de testes de software com foco nas exigências de qualidade do cliente para a realização de testes nos momentos que antecedem a entrega do produto final ao cliente. O modelo de referência utilizado para classificar as necessidades de qualidade do cliente é a norma ISO/IEC 9126 que, agregado ao método de Desdobramento da Função Qualidade estabelece os relacionamentos entre as necessidades de qualidade e as técnicas de teste, permitindo que o esforço da atividade de testes possa ser otimizado.

**Palavras-chave:** Testes de Software, Qualidade do Produto de Software.

## Abstract

Software Testing is an essential activity in the software development process, and it can spend till five times more than all the other activities. Several factors like the diversity of software testing techniques, the time allocated to this activity and the need to address the customer requirements, should be analyzed together to better define and execute the tests. This paper proposes an approach of defining software testing priority based on the customers' quality needs while testing a system before putting the system in use. To this purpose it is used the ISO/IEC 9126 and the Quality Function Deployment that establishes the relationship among software quality characteristics and software testing techniques.

**Key-words:** Software Testing, Product Software Quality.

## 1. Introdução

As organizações dependem de seus clientes devendo entender suas necessidades, atuais e futuras, e procurar superar suas expectativas. Dessa forma, existe cada vez mais a preocupação de garantir que o cliente receba o software solicitado com a qualidade requerida. A garantia de qualidade passou, portanto, a ter um papel importante na produção de software. Nesse contexto, a atividade de testes é considerada um elemento crítico para a garantia de qualidade de software ([1], p 68). A variedade de técnicas existentes é expressiva e o número de permutações de caminhos possíveis para se testar uma aplicação é excessivamente grande, tornando-se impossível realizar testes exaustivos no sistema [2].

O objetivo da atividade de teste é encontrar os erros antes do usuário[2]. Uma das formas de se encontrar estes erros é utilizar diversas técnicas no sentido de exercitar o software para que os

erros sejam revelados e corrigidos em tempo de desenvolvimento. As atividades de testes de software são definidas em diferentes estágios de forma a serem executadas em sintonia com as fases do desenvolvimento. Portanto, um estágio que merece atenção especial é o de aceitação, pois antecede à entrega do produto ao cliente e capta sua percepção sobre a qualidade do produto que será efetivamente entregue.

Para poder mensurar essa percepção da qualidade se faz necessário ter definido um conjunto de características de qualidade para o software a ser testado. A ISO/IEC 9126 [3] propõe esse conjunto de características identificando entre as suas aplicações o apoio na identificação de objetivos de testes e de critérios de aceitação para um produto de software. A última pesquisa do Ministério de Ciência e Tecnologia [4] mostra que no universo de empresas que estão adotando as práticas de teste de software, 57,2% afirmam que realizam testes de aceitação. No entanto, a pesquisa indica que mais de 50% das empresas não usam os critérios de qualidade da norma ISO/IEC 9126[3], deixando crer que esses testes não consideram os critérios de qualidade.

Outras dificuldades na realização dos testes de software podem ainda se apresentar, quais sejam: os custos dispensados com a atividade de testes[2]; a falta de cultura em não tratar a realização de testes como um processo [5]; e, o tempo destinado, que, geralmente, é sacrificado em função dos atrasos no desenvolvimento. A equipe responsável pela atividade de testes de software necessita, portanto, de uma estratégia para que os resultados obtidos sejam eficientes, ou seja, que utilize técnicas adequadas que possibilitem avaliar as características de qualidade desejadas pelo cliente e que permitam otimizar prazo e custos.

Esse artigo propõe uma abordagem de testes de software a serem aplicados na fase de aceitação, suportados por uma técnica de apoio à decisão que sustente um processo sistemático baseado na ISO/IEC 14598-5 [6] (norma que propõe um processo de avaliação de um produto). A técnica de apoio à decisão utilizada é a *Quality Function Deployment* [7] (QFD) ou Desdobramento da Função da Qualidade, originalmente definida para produção em fábrica e utilizado em software na definição de requisitos. O princípio básico dessa técnica é focar nos requisitos do cliente. O artigo não exclui as fases anteriores à aceitação.

Serão apresentadas, nas próximas seções, uma descrição do QFD (seção 2), uma discussão sobre testes de software (seção 3), a abordagem proposta nesse artigo (seção 4), uma aplicação prática dessa abordagem (seção 5) e conclusões obtidas nessa pesquisa (seção 6).

## **2. Desdobramento da Função de Qualidade**

O *Quality Function Deployment* (QFD) [7] foi concebido no Japão no final dos anos 60, durante uma época onde as indústrias romperam com o modo de produção pós-guerra, baseado na cópia dos produtos estrangeiros e passaram a produzir baseadas na sua originalidade. O QFD nasceu nesse ambiente como um método de desenvolvimento de novos produtos como parte da abordagem da Qualidade Total.

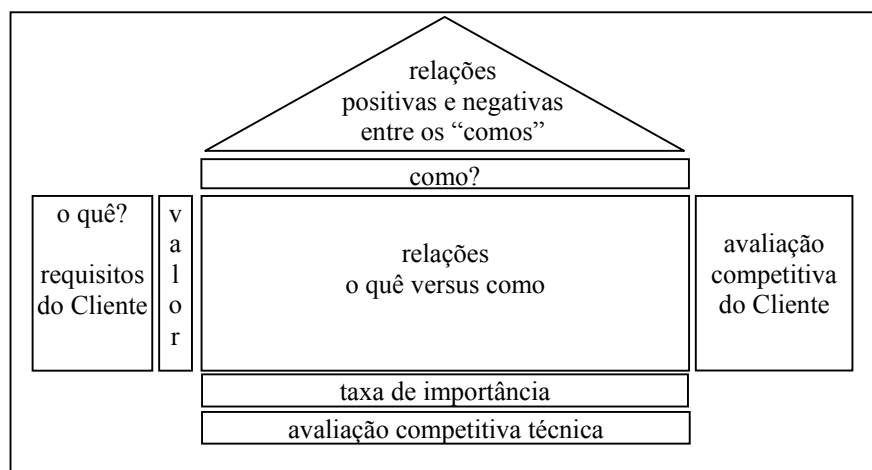
O QFD é uma forma sistemática de assegurar que o desenvolvimento de atributos, características e especificações do produto, assim como a seleção e o desenvolvimento de equipamentos, métodos e controles do processo sejam dirigidos para as demandas do cliente ou do mercado, ou seja, o QFD traduz as necessidades dos clientes em requisitos apropriados para a empresa, em cada ciclo do desenvolvimento do produto, desde a pesquisa e o desenvolvimento até a engenharia. Portanto, sua aplicação permite diminuir problemas no início da produção, minimizar mudanças no projeto, encurtar os ciclos de desenvolvimento, maximizar a produtividade e reduzir custos [8].

O QFD tem sido amplamente utilizado em diferentes áreas por ser uma ferramenta flexível e adaptada às mais diversas necessidades de planejamento de produtos. Alguns exemplos de uso do QFD são: a captura de requisitos e o auxílio ao desenvolvimento de sistemas [9],[10], aplicações na área médica [11] e planejamento educacional [12].

Segundo Haah *et al.*[10] o QFD utilizado na produção de software implica em diferentes benefícios, como: aumento da atenção para as perspectivas dos clientes, melhora na comunicação entre departamentos e usuários, fundamentação para justificar as decisões, quantificação qualitativa dos requisitos do cliente, representatividade dos dados facilitando o uso de métricas, definição mais rápida das características do produto e capacidade de adaptação a várias metodologias.

A metodologia do QFD envolve basicamente quatro fases que ocorrem através do processo de desenvolvimento do produto [8]: *fase de planejamento do produto*, para definir e priorizar as necessidades dos clientes, analisar as oportunidades oferecidas pela concorrência, planejar o produto para responder às necessidades e oportunidades e estabelecer os valores das características críticas; *fase de desdobramento de componentes*, que culmina com a identificação das características dos componentes que sejam críticos para a produção; *fase do planejamento do processo*, que representa a transição do projeto para as operações de fabricação onde é estabelecida a relação de cada processo a cada característica crítica do componente e listagens de parâmetros de controle dos processos; e, *fase de planejamento da produção* responsável por transferir as informações geradas para o chão de fábrica.

Durante cada fase, uma ou mais matrizes de relacionamentos, chamadas de “casas da qualidade”, são preparadas para ajudar o planejamento e definir quais processos e informações do projeto são críticas para a produção. A “casa da qualidade” é o produto principal do QFD sendo utilizada para apoio à decisão. Em linhas gerais, a casa da qualidade é composta dos seguintes elementos [8](figura 1):



**Figura 1 - A Casa da Qualidade do QFD**

- (i) os objetivos ditados pelo cliente chamados de requisitos do cliente ou simplesmente “o quê”;
- (ii) valores, ou peso, definidos pelo cliente para cada item “o quê”;

- (iii) requisitos do produto de como alcançar ou avaliar os requisitos que respondem as necessidades dos clientes, denominado na figura 1 de “como”;
- (iv) relações entre os requisitos do cliente (“o quê”) e os requisitos do produto (“como”) sendo usados símbolos e números para expressar as relações fracas, moderadas ou fortes;
- (v) relações positivas ou negativas entre os requisitos do produto, ou seja entre os itens “como”;
- (vi) uma comparação com a concorrência sobre sua competência em relação aos requisitos do produto, chamada de avaliação competitiva técnica;
- (vii) avaliações do cliente (ou avaliação competitiva do cliente) em relação a cada item “o quê” no que se refere aos competidores e a equipe interna; e,
- (viii) taxa de importância, obtida através da multiplicação do peso dos requisitos do cliente pelo fator de relacionamento.

Outros elementos podem ser adicionados como, por exemplo, o grau de dificuldade de realizar cada “como” ou o custo do mesmo.

A partir da definição da “casa da qualidade” pode-se analisar a matriz para tomar decisões sobre as áreas que merecem foco e as ações necessárias no planejamento.

### **3. A Atividade de Teste na Qualidade de Software**

A atividade de teste desempenha um importante papel nas atividades de garantia de qualidade de software. No ciclo de vida de um sistema, especificamente no processo de desenvolvimento regulamentado pela NBR ISO/IEC 12207 [13] existem treze atividades, das quais cinco estão diretamente ligadas às atividades de teste. Dado este fato, não é incomum que uma organização gaste cerca de 40% do esforço de projeto total com testes e em projetos dos quais dependem vidas humanas, indústrias aeronáuticas, geradoras de energia e petroquímicas. Este esforço pode chegar a cinco vezes mais do que todo esforço gasto no desenvolvimento[2].

Todos os esforços na atividade de teste concentram-se em projetar casos de teste que tenham probabilidades de encontrar erros no software, ou seja, situações que exercitem o sistema de acordo com os requisitos de qualidade para verificar se estes estão sendo realizados. Os casos de teste refletem os requisitos que devem ser verificados para que o cliente seja satisfeito, são mecanismos que tentam colocar o sistema à prova em uma situação específica de um requisito de qualidade particular [14].

Existem várias técnicas para projetar casos de teste. Apenas para se ter uma idéia, a partir da literatura conseguimos identificar mais de 20 técnicas disponíveis somente para testes de software. Estas técnicas estão divididas em dois grupos: as que conhecem as peculiaridades internas do código, denominadas “caixa-branca” como o teste de caminho básico e as que visam apenas sua interface ou “caixa-preta” como o método de particionamento de equivalência, que divide o domínio de entrada de dados em classes, através das quais os casos de teste possam ser derivados[14].

O planejamento de testes pressupõe que as técnicas estejam intimamente ligadas às características que devem ser avaliadas e como quantificar esta avaliação. Para alcançar um planejamento exequível e justificável faz-se necessário uma estratégia de testes, onde as técnicas de projeto de casos de teste se integram em uma série bem definida de passos resultando na construção de um bom software[2]. Uma boa estratégia de testes deve ser específica, prática e justificada para esclarecer quais as principais tarefas e desafios do projeto de testes.

Como o esforço de teste deve ser concentrado, alguns requisitos devem ser observados para maximizar a capacidade da equipe de testes. A atividade de testes deve garantir as seguintes necessidades fundamentais [16]: sem requisitos estabelecidos, a atividade de teste torna-se impossível; um produto de software deve satisfazer os seus requisitos estabelecidos; todos os casos de teste devem ser rastreados para um ou mais requisitos; e, os requisitos devem ser estabelecidos em termos que possibilitem a sua testabilidade.

O desenvolvimento de software possui características limitantes como qualquer projeto. Recursos limitados (prazo, financeiros, humanos), inovação tecnológica e complexidade do problema tornam a atividade de desenvolvimento uma tarefa difícil de se gerenciar [17]. Esse conjunto de restrições e limitações no desenvolvimento pode afetar a atividade de testes e levar a um caminho que muitos achariam indevido; liberar produtos mesmo que apresentem alguns erros, desde que estes sejam controlados. Bach (1997) [18] afirma, por exemplo, que é reconhecido que alguns sistemas são liberados contendo erros de software, indicando que a própria Microsoft começa cada projeto com a certeza que irá entregar o produto com erros conhecidos.

A atividade de testes pode ocorrer em vários momentos (estágios ou fases) dentro do ciclo de desenvolvimento do produto. Esses estágios implicam nos seguintes testes: *teste de unidade*, onde os testes são aplicados nos menores componentes de código criados; *teste de integração*, que visa validar se a combinação de componentes funciona corretamente quando colocados juntos; *teste de sistema*, que exercita o sistema como um todo ou um subsistema dentro do ambiente operacional controlado; e, *teste de aceitação*, onde são realizados os testes pelos usuários finais com foco principalmente em funcionalidade e usabilidade. Entretanto, alguns autores divergem quanto à nomenclatura e aos estágios referentes a teste de sistema e aceitação. Pressman (2001) [2] e Inthurn (2001) [19] definem os quatro estágios como sendo: unidade, integração, validação/aceitação e sistema. Maldonado e Fabri (2001) [1] definem três estágios: unidade, integração e sistema. Binder (2002, p.749) [15] define quatro estágios: unidade, integração, sistema e aceitação. O estágio que este trabalho se refere como aceitação pode ser definido como o momento que antecede a entrega do produto ao usuário, onde devem ser realizados os testes de sistema e de validação/aceitação com a participação efetiva do usuário quando necessário. Este momento acarreta pressões na equipe de desenvolvimento pois a presença do usuário, seja no apoio à execução ou como receptor de resultados, gera expectativas sobre o que deve ser exercitado na aplicação. Dessa forma, todos os fatores apresentados anteriormente são ainda mais críticos, ou seja:

- o prazo destinado à atividade de testes nessa fase é muito curto. Sabe-se que em muitas situações os gerentes destinam o tempo para ajustes no desenvolvimento [5];
- existem muitas técnicas e tipos de testes disponíveis para essa avaliação final; e,
- o software apresenta muitas funcionalidades, o que torna impossível o exercício total da combinação de entradas, transformações e saídas produzidas pelo software.

Dessa forma, faz-se necessário uma estratégia de testes que busque testar o máximo possível (no que se refere às funcionalidades do sistema) com foco nos requisitos principais do cliente.

#### **4. Abordagem de Priorização de Testes de Software**

Com a preocupação de definir uma estratégia que apoiasse a atividade de testes antes da liberação do produto para o cliente, foi definida: i) uma abordagem, adaptada do QFD, para

escolher prioritariamente as técnicas de teste de software que refletissem os requisitos do cliente e ii) a escolha de um processo sistemático de testes de sistemas para realizar esses testes a partir da abordagem adaptada. Apresenta-se a seguir esses dois focos da abordagem proposta.

#### **4.1 QFD para Testes de Software**

Para a realização desta proposta, foi construída uma adaptação da casa da qualidade que pudesse servir de base para apoiar a decisão quanto às técnicas de teste mais adequadas para atender aos requisitos do cliente. Uma vez adaptada, a casa da qualidade para teste de software, será reaproveitada nos planejamentos subseqüentes.

Dessa forma, o método do QFD foi adaptado para ser utilizado na fase de planejamento de testes de software. A construção dessa casa da qualidade para testes implicou na análise de cada um dos oito elementos definidos na seção 2, ou seja, (i) analisar os itens “o quê (requisitos do cliente)”, (ii) o valor para cada item “o quê”, (iii) os itens “como (requisitos do produto)”, (iv) as relações entre os itens “o quê” e “como”, (v) as correlações entre os itens “como”, (vi) a avaliação competitiva do cliente, (vii) a avaliação competitiva técnica, e (viii) a taxa de importância (ver figura 1).

##### **(i) Definindo os requisitos do cliente**

Para definir quais eram os requisitos do cliente, adotou-se como referência as características de qualidade da ISO-9126 [3], considerando que essas características traduzem o que o cliente percebe sobre o produto final. A própria norma recomenda que as características de qualidade podem ser utilizadas para priorização de testes de software. Portanto, os requisitos definidos como requisitos do cliente foram as sub-características<sup>1</sup> propostas para as seis características de qualidade: funcionalidade, usabilidade, confiabilidade, efetividade, manutenibilidade e portabilidade, tendo em vista o modelo original do QFD indicar o desdobramento dos requisitos primários do cliente. Desta forma, está definida a primeira linha da matriz de QFD, ou seja, os itens “o quê”.

##### **(ii) Definindo o valor para os requisitos de qualidade**

Para estabelecer o valor para as características de qualidade, definiu-se um formulário para capturar dois tipos de informação. A primeira informação é a necessidade dos interessados<sup>2</sup> acerca das características de qualidade. Para cada característica de qualidade o interessado deve responder no “Quadro de Avaliação de Características de Qualidade” (quadro 1) a importância da característica de qualidade com valores que variam de 1 a 5 representando, respectivamente, “sua falta não causa impacto”, “pouco importante”, “desejável”, “muito importante” e “imprescindível”.

---

<sup>1</sup> Ao longo do texto, para simplificar, as sub-características serão referenciadas como características de qualidade.

<sup>2</sup> Interessado é uma pessoa que pode ser afetada pelo produto. A opinião de um interessado pode resultar em mudança no produto.

### Quadro 1 - Trecho do questionário de avaliação da característica de funcionalidade

Características de Qualidade	VALOR
<b>Adequação</b> Qual a importância que você considera do sistema sendo avaliado <u>oferecer exatamente o que foi especificado</u> no início do projeto, ou seja, no ato da sua entrega, suas funções não possuem mais ou menos detalhes, sem com isso prejudicar o seu objetivo final?	
<b>Interoperabilidade</b> Qual a importância que você considera do sistema sendo avaliado necessitar para seu correto funcionamento a <u>troca e confiabilidade de dados de outros sistemas?</u>	

A segunda informação é a percepção de falhas que o interessado visualiza na equipe de desenvolvimento que costuma atendê-lo. A percepção de falhas somente deve ser respondida caso os interessados já tenham trabalhado com toda ou parte da equipe que está desenvolvendo o software. O objetivo desta informação é equilibrar os anseios sob cada característica de qualidade respondida no quadro de avaliação de características de qualidade com a qualidade do serviço prestado pela equipe conhecida. A percepção de falhas pode aumentar ou diminuir a prioridade dos testes para cada característica de qualidade, onde o interessado responde com valores que variam de 1 a 5 representando, respectivamente, “nunca aconteceu”, “acontece raras vezes”, “acontece com certa frequência”, “acontece várias vezes” e “sempre acontece” obtendo-se a média de cada característica de qualidade com a percepção de falhas relativa.

#### (iii) Definindo como avaliar

O item “como” no QFD estabelece como avaliar ou garantir o item “o quê”. O modelo original do QFD, pressupõe o desdobramento de itens “o quê” em itens “como”. Entretanto, as características de qualidade não podem ser desdobradas além das subcategorias de qualidade descritas na norma ISO/IEC 9126, portanto foram definidas técnicas de teste que tinham aplicabilidade no estágio de aceitação. Foram escolhidas 20 técnicas com base nos seus objetivos e que seus procedimentos fossem possíveis no estágio de aceitação. Algumas dessas técnicas são [15]: Caso de Uso Estendido, Cobertura em CRUD<sup>3</sup>, alocação por Perfil, Grafo de Causa e efeito, Particionamento de Equivalência e Teste de Usabilidade.

#### (iv) Definindo a relação entre “o quê” e “como”

Para estabelecer (v) a relação entre as características de qualidade (“o quê”) e as técnicas de teste (“como”) analisou-se como a técnica de teste ajudaria a avaliar cada característica de qualidade. Conforme definido na ISO/IEC 9126 [3] para que se avalie as características de qualidade de um produto deve-se considerar as métricas de qualidade. Dessa forma, foram consideradas as métricas de qualidade externa<sup>4</sup> definidas na ISO/IEC TR 9126-2 [20]. Dessa forma, analisou-se cada técnica de teste no que se refere à possibilidade da mesma gerar insumos para calcular as métricas. Considerando, por exemplo, a técnica Caso de Uso Estendido<sup>5</sup> [15], os relacionamentos com as características de qualidade interoperabilidade<sup>6</sup> e adequação<sup>7</sup> foram

<sup>3</sup> CRUD, do acrônimo em inglês Create, Read, Uppdate e Delete. Inserção, leitura, atualização e exclusão de dados.

<sup>4</sup> Métricas de qualidade externa permitem avaliar o produto de software durante o teste e operação, ou seja, quando o produto de software é executado. [20]

<sup>5</sup> A técnica caso de uso estendido visa exercitar os cenários de cada caso de uso, incluindo navegação, entrada, processamento e obtenção de dados [15]

<sup>6</sup> Interoperabilidade refere-se na atributos do software que evidenciam sua capacidade de interagir com sistemas especificados[3]

definidos, respectivamente com valores moderado (3) e forte (9). O primeiro valor foi determinado devido às métricas *troca de dados baseado em formato* e *troca de dados baseada nas tentativas do usuário* estabelecidas para interoperabilidade serem possíveis de coletar na técnica. Entretanto, nem todas as funcionalidades do sistema avaliado trocam dados com outros sistemas. O segundo relacionamento foi declarado como forte (9) devido ao fato da técnica gerar todos os insumos necessários para as métricas das características de adequação quais sejam: *adequação funcional*, *completeza da implementação*, *cobertura da implementação* e *estabilidade da especificação*. Foram analisados 540 relacionamentos (considerando 27 características de qualidade x 20 técnicas de teste) para um total de 66 métricas propostas pela norma.

Os valores para as relações entre as características de qualidade e as técnicas são: 1, 3 e 9, para relações fracas, moderadas e fortes, respectivamente.

#### (v) Definindo a correlação entre os itens “como”

As correlações entre as técnicas de teste (que representam os “comos”) foram definidas considerando a similaridade de preparação de ambiente e forma de execução entre duas técnicas, bem como a capacidade das técnicas em gerar os mesmos insumos para as características de qualidade. No modelo original do QFD, podem-se ter correlações negativas, entretanto na metodologia proposta essas foram retiradas pois não existem técnicas de teste que possam mutuamente se prejudicar.

#### (vi) Definindo a avaliação competitiva técnica

A avaliação da competitividade técnica é realizada através de um questionário (quadro 2) submetidos a empresas de credibilidade no mercado de teste de software. Esse questionário está baseado nas 21 técnicas de teste escolhidas. Os valores a serem respondidos para cada técnica variam de 1 a 5 sendo respectivamente: “não conhece”, “conhece, mas não usa”, “conhece e começa a usar”, “conhece e usa sistematicamente” e “especialidade da empresa”.

**Quadro 2 – Questionário de Avaliação Competitiva Técnica**

Técnica	Objetivo	Resposta
Caso de Uso Estendido	Exercitar os cenários de cada caso de uso, incluindo navegação, entrada, processamento e obtenção de dados;	
Cobertura em CRUD	Verificar se todas as operações básicas foram exercitadas para cada objeto do domínio do problema. Utiliza-se a matriz CRUD (Create, Read, Update e Delete)	
Alocação por Perfil	Alocar esforços para cada caso de uso em conformidade com sua frequência relativa;	
Grafo de Causa e efeito	Através da representação concisa das condições lógicas e das ações correspondentes, gera-se o grafo de causa e efeitos que é convertido em tabelas de decisão para que se possa exercitar cada regra da tabela	

#### (vii) Definindo a avaliação competitiva do cliente

A avaliação competitiva do cliente refere-se a uma comparação do “o que” (requisitos de qualidade), item a item, entre um produto da companhia e produtos similares da concorrência. Essa informação é necessária para decidir se um determinado requisito pode ser ignorado pelo fato da equipe interna ter nível de superior em relação aos concorrentes. Contudo, considerou-se que mesmo que a equipe de desenvolvimento interna seja especialista em uma determinada

<sup>7</sup> Adequação refere-se a atributos do software que evidenciam a presença de um conjunto de funções e sua apropriação para as tarefas especificadas [3].



característica, deve-se sempre avaliar a mesma. Dessa forma, esse elemento da casa da qualidade não foi considerado.

#### **(viii) Definindo a taxa de importância**

O cálculo da taxa de importância é semelhante a qualquer casa da qualidade da matriz de QFD, ou seja, considera-se o peso de cada requisito do cliente (“o quê”, nesse caso peso para as características de qualidade) multiplicando-se o valor da relação entre o requisito do cliente e do produto (nesse caso as técnicas de teste). A fórmula para se calcular a taxa de importância é

$$ICQ = \sum_{i=1}^n PCQ_i * IR_{ij}, \text{ onde } i \text{ varia de } 1 \text{ a } n, \text{ sendo } n \text{ o número de linhas da matriz}$$

(características de qualidade) e  $j$  o número da coluna (técnicas de teste) onde se está fazendo o cálculo. PCQ é a média do valor de cada característica de qualidade considerando-se a percepção de falhas. A taxa de importância determinará qual a priorização das técnicas de teste podem atender a maior parte dos requisitos de qualidade do cliente.

## **4.2 Uma Metodologia para Testes de Software**

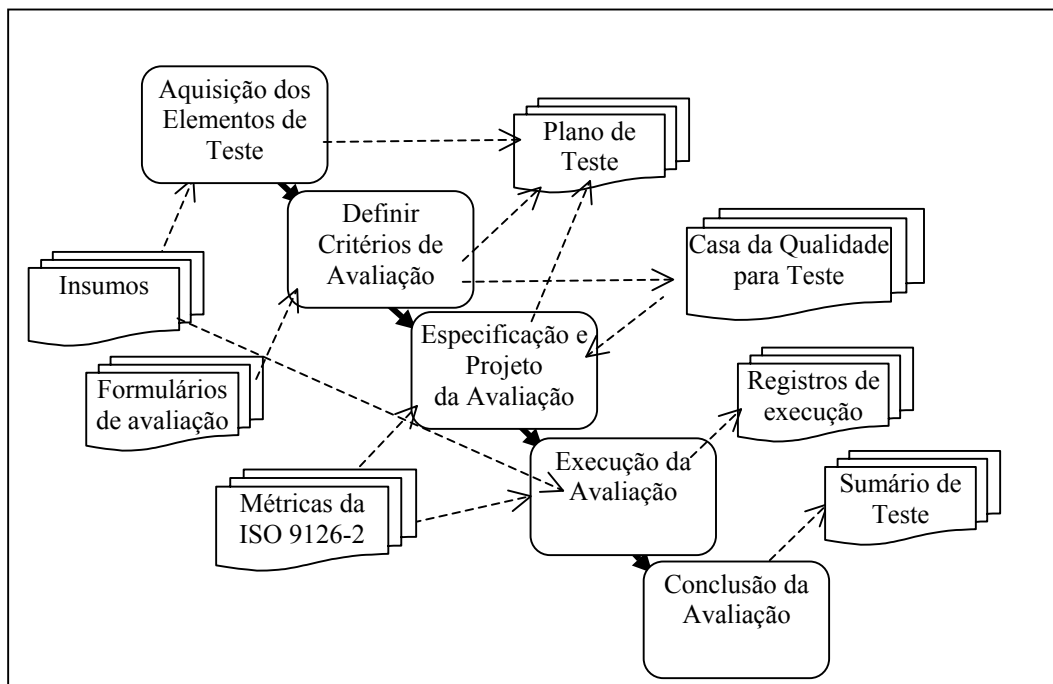
Uma vez definido o QFD para testes de software criou-se um processo para testar sistematicamente o sistema de forma a garantir a priorização de testes com foco nos requisitos do cliente. É importante destacar, que o objetivo foi realizar uma metodologia de planejamento de testes para o momento de aceitação, não excluindo os testes anteriores. Isso implica, que os testes de unidade, integração e sistema já foram realizados.

Esse processo foi definido baseado na norma ISO/IEC 14598-5 [6] sendo composto de cinco atividades (Figura 2): aquisição dos elementos de testes, definição dos critérios de avaliação, especificação e projeto da avaliação, execução da avaliação e conclusão da avaliação. Para execução desse processo são necessários:

- Insumos – todo e qualquer artefato gerado no desenvolvimento que possa ser útil para a execução de testes no sistema (dentre eles destacamos, a especificação de requisitos e diagramas e descrição de casos de uso);
- Formulários de avaliação – questionários para coletar o grau de exigência de cada característica de qualidade, a percepção de falhas, a prioridade das funcionalidades, e a avaliação competitiva técnica;
- Casa da Qualidade para Testes – casa de qualidade definida na seção anterior; e,
- Métricas – conjunto de métricas da ISO/IEC TR 9126-2 [20] de forma a estabelecer os dados que são necessários coletar para verificar o atendimento aos requisitos de qualidade.

Durante a execução do processo são gerados os seguintes artefatos:

- Plano de Testes - é o instrumento contratual entre os avaliadores e interessados e estabelece para os avaliadores a guia para os testes;
- Casa de qualidade para testes preenchida para o sistema específico;
- Registros de execução - registro dos resultados obtidos a cada execução de caso de teste demonstrando a falha encontrada ou sucesso encontrado e os horários de início e término de execução para definição do esforço na execução; e,
- Sumário de testes - resultados obtidos na execução do Plano de Testes através das métricas relacionadas, apresentando sugestões pertinentes à melhoria da qualidade.



**Figura 2 – Processo Sistemático de Testes**

A atividade de *Aquisição dos Elementos de Testes* tem como objetivo identificar através de entrevistas e da documentação de projeto fornecida pelos interessados na avaliação quais são os elementos que compõe o conjunto de especificações de requisitos do sistema a ser avaliado. Essa atividade não está explicitamente definida na ISO/IEC 14598-5 [4] mas é necessária para se ter conhecimento do sistema a ser testado. Para realizar essa atividade é necessário coletar e fazer a leitura dos documentos (insumos) fornecidos pelos interessados na qualidade, identificar os requisitos funcionais do sistema (obtidos através dos casos de uso) e as regras de negócio mapeando com as funcionalidades que devem implementá-las. Deve-se ainda definir qual o contexto do sistema no que se refere às suas características principais e quem são os interessados na avaliação. A seção 2 (Descrição do Produto) do Plano de Teste é preenchida como resultado dessa atividade.

A atividade *Definir Critérios de Avaliação* tem como objetivo principal capturar a percepção de qualidade da aplicação a ser entregue sob a perspectiva dos interessados (usuário) e priorizar as funcionalidades mais importantes para os mesmos. A definição dos requisitos de qualidade considera as características de qualidade da norma ISO/IEC 9126. Para isso é submetido o questionário apresentado no quadro 1 identificando o que o usuário deseja em termos de qualidade. Além disso, é solicitado definir a percepção de falhas conforme apresentado na seção anterior (item (iii)). A priorização das funcionalidades considera a média aritmética de dois fatores que são respondidos pelos interessados (usuário):

- O Perfil Operacional da funcionalidade, ou seja, sua frequência de uso. O interessado responde para cada funcionalidade valores de 1 a 3 que correspondem, respectivamente, a

“raramente (algumas vezes no mês ou ano)”, “regularmente (alguns dias da semana)” e “freqüentemente (todos os dias)”;

- O Impacto Negocial, que representa em caso de falha ou indisponibilidade da funcionalidade no sistema, o prejuízo para a reputação da instituição. O interessado responde para cada funcionalidade valores de 1 a 3 que correspondem, respectivamente, a: “baixo (afeta somente algumas pessoas dentro da instituição e sem perdas financeiras)”, “médio” (afeta grande parte das pessoas envolvidas no uso do sistema na organização ou causa prejuízo financeiro pequeno)” e “alto (afeta as pessoas internas e externas à instituição ou causa prejuízos financeiros expressivos)”.

Nessa atividade deve-se ainda capturar informações sobre a avaliação competitiva técnica (ver quadro 2). Essa avaliação pode ser realizada uma única vez na organização, sendo a mesma utilizada para testes de diferentes sistemas. Deve-se apenas garantir uma periodicidade de atualização.

Como resultado dessa atividade devem ser preenchidas a seção 3 (Critérios de Avaliação) do Plano de Teste e as informações referentes à Casa da Qualidade coletadas.

A atividade *Especificação e Projeto da Avaliação* têm como objetivo planejar as execuções dos testes no que se refere às técnicas utilizadas, recursos (hardware, software e pessoas), cronograma, as métricas de avaliação a serem consideradas e os casos de teste a serem realizados. Com a Casa de Qualidade pronta, deve-se definir os ciclos de execução de testes, da seguinte forma:

- Observar qual a técnica obteve maior taxa de importância. O primeiro ciclo de teste deve começar com esta técnica, a menos que: a equipe de testes tenha nível de experiência inferior com a técnica em relação a um dos concorrentes (observado pela avaliação competitiva técnica) e se deseja propor uma terceirização com o mesmo; ou, não se tenha disponível a documentação do sistema necessária para executar a técnica. Em ambos os casos, pode-se optar pela próxima técnica com maior taxa de importância. No caso de não executar uma técnica sugerida deve-se registrar a justificativa.
- Definir os objetivos, um resumo do procedimento e o prazo previsto para o ciclo de teste com a técnica escolhida.
- Estabelecer as métricas a serem coletadas.
- Projetar os Casos de Teste de acordo com as funcionalidades priorizadas e observando as métricas que devem ser colhidas na sua execução. A depender do prazo que se tenha para execução dos testes pode-se avaliar inicialmente as funcionalidades mais prioritárias.

Deve-se definir ciclos testes tantos quanto possíveis de serem realizados dentro do prazo definido pelos interessados. Com os ciclos definidos deve-se aprovar junto aos interessados o cronograma de execução das atividades de teste contemplado no plano.

Nessa atividade as seções 4 e 5 (dados de Projeto de Teste e Execução) do Plano de Teste são preenchidas.

A atividade de *Execução da Avaliação* implica em testar o sistema de acordo com os ciclos de testes definidos. Para alcançar esse objetivo deve-se preparar o ambiente de testes para que a técnica de teste definida no ciclo possa ser executada sem problemas; e, caso tenha sido decidido utilizar ferramentas de automatização de testes, deve-se implementar os casos de teste na ferramenta escolhida. Deve-se então executar os casos de teste projetados e implementados. Todas essas tarefas consomem tempo e recurso devendo ser registrado no cronograma.

Finalmente, a medida que os testes forem executados devem ser atualizados os valores para cálculo das métricas projetadas para o caso de teste nos Registro de Execução. Caso o tempo necessário para testes seja suficiente para a execução de outro ciclo, essa atividade de Execução da Avaliação é reiniciada. Caso contrário, esta atividade se encerra.

Finalmente a atividade de *Conclusão da Avaliação* tem como objetivo fornecer aos interessados os resultados obtidos na avaliação do produto através das métricas estabelecidas. Deve-se, portanto, calcular as métricas adotadas para os ciclos de execução com base nos valores dos Registros de Execução obtidos na atividade anterior. Após as métricas terem sido calculadas gera-se o Sumário de Teste, contendo informações sobre a cobertura de testes, os custos, os defeitos encontrados e sugestões de melhorias.

## 5. Aplicação Prática

Essa abordagem foi utilizada para realizar os testes de aceitação de um Sistema de Gestão de Pesquisa (SIGEP). Esse sistema tem como objetivo apoiar a elaboração de projetos de pesquisa, a avaliação desses projetos com a aprovação de recursos, o acompanhamento e a análise de resultados. Esse sistema é composto de 25 casos de uso e foi desenvolvido em Java, JSP, com banco de dados PostgreSQL. O sistema foi desenvolvido por uma equipe de 5 pessoas (entre analistas e programadores) levando em torno de 5 meses num total de 4.650 horas de trabalho.

A execução dos testes de aceitação foi realizada por uma equipe de 2 pessoas que não participaram do desenvolvimento. A execução das atividades do processo apresentado na seção anterior consumiu 16 horas de trabalho, conforme apresentado na Tabela 1.

**Tabela 1 – Tempo gasto com as atividades do estudo de caso SIGEP**

<b>Atividade</b>	<b>Horas</b>
Aquisição de Elementos de Teste	14
Definir Critérios de Avaliação	14
Especificação e Projeto da Avaliação	18
Execução da Avaliação	16
Conclusão da Avaliação	8
<b>TOTAL</b>	<b>70</b>

Na atividade de *Aquisição de Elementos de Teste* foram lidos os casos de uso do sistema e as regras de negócio bem como toda a documentação referente aos requisitos não funcionais e modelos (classe, caso de uso, seqüência) da aplicação. Os casos de uso estavam com certa tendência de fatoração funcional, visto que a maioria deles possuía de três a quatro relacionamentos de extensão e inclusão sem que houvesse reaproveitamento desses casos de uso por outros. Contudo a modelagem e a especificação escrita estavam bem legíveis e de fácil entendimento, fato que foi observado devido à necessidade de se realizar apenas uma reunião com a equipe de desenvolvimento para esclarecer dúvidas sobre as funcionalidades. Tendo em vista que o sistema desenvolvido foi concebido para funcionar como aplicação web a identificação dos alvos de teste foi simples.

Na atividade *Definir os Critérios de Avaliação*, 4 interessados (usuários) na qualidade responderam os questionários de avaliação e os formulários de avaliação competitiva foram encaminhados para 3 empresas de renome nacional para que fizessem uma auto-avaliação em relação ao conhecimento de cada técnica de teste contida no formulário. Como resultado da

aplicação dos questionários foram estabelecidas as características mais importantes na avaliação conforme pode ser observado na tabela 2. O fato interessante observado foi que acurácia e funcionalidade, duas características geralmente exercitadas em primeiro lugar nos planos de teste de validação/aceitação ficaram com o 8º e 9º lugar, respectivamente, enquanto que maturidade e modificabilidade ocuparam o 1º e 2º lugar, fato interessante mas justificável, pois os interessados na qualidade demonstravam preocupação com a longevidade do projeto.

**Tabela 2 – Características mais priorizadas no SIGEP**

Características de Qualidade		Peso
Confiabilidade	Maturidade	5
Manutenibilidade	Modificabilidade	5
Usabilidade	Inteligibilidade	5
Usabilidade	Apreensibilidade	5
Usabilidade	Operacionalidade	5
Confiabilidade	Recuperabilidade	4
Funcionalidade	Acurácia	4
Funcionalidade	Adequação	4
Manutenibilidade	Testabilidade	4

Na atividade de *Especificação e Projeto da Avaliação* as técnicas escolhidas em função dos valores calculados na casa da qualidade foram caso de uso estendido e usabilidade. O plano de testes foi elaborado e definido os ciclos de execução com base no resultado da casa da qualidade. O projeto de casos de teste implicou na observação das métricas que deveriam ser coletadas quando a execução dos casos de teste fosse realizada. Para o projeto foram projetados 40 casos de teste seguindo um modelo padronizado. Um dos pontos de dificuldade no projeto destes casos de teste foi o relacionamento das métricas, pois exigia uma atenção especial, em cada seção (dados de entrada, procedimento, considerações, etc.) do caso de teste que deveria ser preenchida como apresentado na Figura 3.

Na atividade de execução da avaliação os casos de teste funcionais, alguns deles foram implementados no Rational Robot<sup>8</sup>, uma ferramenta de automação, pois alguns casos de teste necessitavam de um número considerável de entrada de dados e pontos de verificação além de serem executados repetidamente. Os casos de teste referentes à usabilidade foram realizados em conjunto com um dos quatro interessados, o que resultou em observações importantes para a melhoria do sistema, principalmente na característica de operacionalidade. Ao todo foram coletadas 28 métricas de qualidade. Alguns exemplos dessas métricas são apresentados na Tabela 3. O valor indesejado ou ideal é definido na própria norma ISO/IEC TR 9126-2 [20]. Entretanto, algumas métricas não puderam ser coletadas por restrições do projeto, devido ao fato de que estas necessitavam de variáveis em relação ao tamanho do software, acesso ao manual de usuário e requisitos explicitados na documentação. Esses fatores ainda não tinham sido definidos para o projeto, pois o mesmo ainda se encontrava em desenvolvimento.

<sup>8</sup> A ferramenta Rational Robot é marca registrada de propriedade da IBM.

Caso de Teste	
-----	
Nome: Nome do caso de teste	
Pré-condição:	
Dados de entrada:	
Procedimento:	
Considerações	
Pós-condição:	
Resultado esperado:	
Resultado verificado:	
Observações	
Métricas	

**Figura 3 – Modelo de caso de teste**

**Tabela 3 – Métricas coletadas no estudo de caso SIGEP**

Categoria	Subcategoria	Nome da Métrica	Valor indesejado	Valor coletado	Valor ideal	Observação
Confiabilidade	Maturidade	Cobertura de testes	0	1	1	
Confiabilidade	Maturidade	Maturidade de testes	0	0,39024	1	
Confiabilidade	Recuperação	Tempo médio de indisponibilidade	>2	0	0	Horas
Eficiência	Tempo	Tempo de resposta	>30	3	0	segundos
Funcionalidade	Adequabilidade	Adequação funcional	0	0,60976	1	
Funcionalidade	Adequabilidade	Completeza de implementação	0	0,23077	1	

Na atividade de *Conclusão da Avaliação* foi gerado o sumário de testes, onde foram calculadas as métricas obtidas nas execuções dos casos de teste em função da execução dos dois ciclos de execução previstos. Os erros foram relacionados no sumário de testes referenciando às métricas e os casos de teste para que os desenvolvedores pudessem reproduzir a situação encontrada na atividade de execução da avaliação. Além dos casos de teste, evidências como captura de imagens da tela com mensagens de erro, resultados de consultas ao banco de dados e dados utilizados nos testes foram passados aos desenvolvedores no sentido de auxiliar a localização e correção dos erros.

Algumas sugestões foram colocadas no sumário de teste como:

- Alteração na especificação da documentação;
- Melhorias nas navegações de tela;
- Tratamento mais rigoroso com as regras de negócio;
- Aspectos estéticos da aplicação;
- Parametrização de informações; e,
- Melhorias na segurança.

## 6. Conclusão

A avaliação de um sistema antes de liberá-lo para uso é de grande importância na garantia da qualidade por permitir capturar o que será realmente entregue ao usuário no que se refere ao software propriamente dito. Diferentes técnicas podem ser utilizadas e diferentes requisitos de qualidade devem ser garantidos. Além disso, essa fase é prejudicada devido a atrasos comuns na fase de desenvolvimento.

Esse artigo apresentou uma abordagem que busca priorizar quais testes devem ser realizados para atender da melhor forma os requisitos de qualidade do cliente definidos através de normas de qualidade dentro de um prazo definido. Para alcançar esse objetivo foi utilizado a norma ISO/IEC 9126 [3] em conjunto com o método Quality Function Deployment [8] de forma a fornecer subsídios para um melhor planejamento dos testes. Essa abordagem já foi utilizada em dois projetos por diferentes equipes. Nesse artigo foi apresentado o resultado da aplicação em um dos projetos. A abordagem apresentou resultados satisfatórios no que se refere aos duas aplicações realizadas oferecendo um melhor retorno para o cliente da qualidade do software que seria liberado para uso.

## Referências

- [1] Maldonado, J. e Fabbri, S., “Testes de Software”, in: Rocha, A., Maldonado, J. e Weber, K.; Qualidade de Software: Teoria e Prática, Prentice Hall, pp. 73-84, 2001.
- [2] Pressman, R. S., Software Engineering A Practioner’s Approach, 5a. Edição, Mc Graw Hill, 2001.
- [3] ISO/IEC 9126-1 Software Engineering - Product Quality – Quality Model, 2001.
- [4] Qualidade e Produtividade no Setor de Software Brasileiro, Ministério de Ciência e Tecnologia, 2001
- [5] Moreira Filho, T. e Rios, E., Projeto e Engenharia de Software: Teste de Software, Alta Books, Rio de Janeiro, 2003.
- [6] ISO/IEC 14598-5, Information Tecnology – Softtware Product Evaluation Part 5: Process Evaluators, 1<sup>st</sup> Edition 1998.
- [7] Akao, Y., QFD: Past, Present, and Future - International Symposium on QFD, 1997.
- [8] Eureka, W. e Ryan, N., QFD Perspectivas Gerenciais do Desdobramento da Função Qualidade, 1992.
- [9] Park, Hee-Sok, Noh, S. J. Enhancement of web design quality through the QFD approach, Vol. 13, No. 3393- 401, 2002.
- [10] Haag, S., Raja M.K., Schkade, L.L., Quality Function Deployment Usage in Software Development, ACM. 39, No. 1, pgs. 41-49, 1996.
- [11] Mazur, G., QFD Applications in Health Care and Quality of Work Life, First International Symposium on QFD, 1995.
- [12] Mazur, G. The Application of QFD to Design a Course in Total Quality Management at The Univ. of Michigan, 1996.
- [13] NBR ISO/IEC 12207, 1998, Tecnologia de Informação – Processos de Ciclo de Vida de Software, Associação Brasileira de Normas Técnicas, Rio De Janeiro, Brasil.
- [14] Myers, G. The Art of Software Testing – Willey, USA, 1979
- [15] Binder, R. Testing Object Oriented Systems: Models, Patterns and Tools, 3<sup>rd</sup> edition, Addison Wesley, Boston, USA, jun. 1999.
- [16] Bach, J. Risk and Requirements-Based Testing, IEEE Computer, June, 1999.
- [17] PMBOK, Project Management Body of Knowledge, PMI, 2000.
- [18] Bach, J. Good Enough Quality: Beyond the Buzzword, IEEE Computer, Aug, 1997.
- [19] Inthurn, C. Qualidade e Teste de Software, Florianópolis, Visual Books, 2001.
- [20] ISO/IEC TR 9126-2: Software engineering - Product quality -Part 2: External metrics, 2003.