

Avaliação de Ferramentas de Apoio ao Teste de Sistemas de Informação

Janielton de Sousa Veloso, Pedro de Alcântara dos S. Neto, Ismayle de Sousa Santos, Ricardo Britto

Departamento de Informática de Estatística - Universidade Federal do Piauí (UFPI)

CEP 64049-550 – Ininga – Teresina – PI - Brasil

{janielton, ismayle, pasn, rbritto}@ufpi.edu.br

***Abstract.** The software testing is the main approach used for software quality assurance. However, this activity consumes many resources. It is fundamental to use mechanisms to facilitate its execution. The use of testing tools is a way to achieve this objective, but it is important to select the most suitable tool for the identified needs. This paper presents a mechanism that allows comparative analysis of the main tools to support software testing demands.*

***Resumo.** A atividade de testes é um dos meios mais utilizados para se garantir a qualidade de um software. Contudo, realizar testes demanda tempo e recursos, o que provoca a busca por mecanismos que auxiliem essa atividade. O uso de ferramentas de testes representa um importante auxílio. No entanto, como existem muitas opções, deve-se ter cuidado para adquirir as mais adequadas às necessidades existentes. Este artigo apresenta um método para avaliação de ferramentas de teste para sistemas de informação, baseado no uso do SQFD. Esse método permite a comparação e seleção da ferramenta mais adequada, a partir da identificação daquilo que é considerado mais importante pelos usuários de tais ferramentas.*

1. Introdução

No processo de desenvolvimento de produtos de *software*, a atividade de teste é fundamental para garantir a qualidade dos resultados. Isso porque os testes de *software* podem ser usados para revelar a presença de defeitos [Myers 2004]. Contudo, a realização dessa atividade demanda tempo e recursos, fazendo com que ela seja muitas vezes responsável por cerca de 50% dos custos do desenvolvimento de um *software* [Harrold 2000]. Dessa forma, as empresas procuram meios para reduzir custos na realização dessa atividade, normalmente a partir do uso de ferramentas de apoio aos testes de *software*.

Mesmo com a grande disponibilidade de ferramentas de testes, ainda existem muitas necessidades que não são supridas por completo. Um dos principais problemas, por exemplo, é a indisponibilidade de ferramentas adequadas para a gestão das atividades de teste [Weber 2001]. A escolha de ferramentas de testes indevidas pode trazer desperdício de tempo, fator relevante no desenvolvimento de *software*, além de influenciar negativamente no sucesso do produto, caso ela atrapalhe de alguma forma a identificação dos defeitos.

É importante que existam meios para se avaliar e comparar ferramentas de teste. Para isso, os requisitos da equipe de teste devem ser identificados e priorizados, de forma a garantir que aquilo que é mais importante, na visão dos seus usuários finais, será considerado. Um método bastante difundido para se identificar e priorizar requisitos é o

QFD (*Quality Function Deployment*). O QFD é uma técnica que pode ser usada para traduzir as necessidades dos clientes em requisitos técnicos de *software* [Akao 1996]. Existe uma adaptação do QFD para o desenvolvimento de *software* conhecido como SQFD (*Software Quality Function Deployment*), que busca melhorar o desenvolvimento de *software* aplicando técnicas de melhoria de qualidade durante a especificação de requisitos [Haag 1996].

Este artigo apresenta um método, baseado no SQFD, para analisar e comparar ferramentas de testes de *software*. Dessa forma, além de fornecer aos desenvolvedores de ferramentas de testes um mecanismo para avaliá-las com relação aos concorrentes, a técnica proposta auxilia as empresas e os testadores a escolher as ferramentas de testes mais adequadas para atender às suas necessidades. O restante deste artigo está organizado da seguinte forma: na Seção 2 são apresentados alguns trabalhos relacionados; na Seção 3 é apresentado o método utilizado no trabalho, descrevendo seus passos principais; na Seção 4 são apresentados os resultados já obtidos com a utilização do método e, na Seção 5 são apresentadas as conclusões e trabalhos futuros.

2. Trabalhos Relacionados

Existem muitos trabalhos que abordam o uso do QFD para diversos fins, como por exemplo, no desenvolvimento de produtos, serviços ou processos, na avaliação de qualidade de serviços, ou mesmo, no desenvolvimento de *software*. No entanto, pouco foi encontrado com relação à aplicação do QFD para avaliação de ferramentas. Na realidade, o único trabalho diretamente relacionado e considerado relevante com o que é proposto neste artigo é o trabalho de Lara [Lara 2001]. Eles especificaram um método de comparação entre produtos de *software* utilizando a Casa de Qualidade do QFD para conversão das necessidades e desejos dos clientes em requisitos de pacote de *software*. Nesse trabalho foi utilizada a norma NBR 12119 [NBR 12119 1998], que estabelece requisitos de qualidade para pacotes de *software*, para guiar a avaliação de um pacote de *software* em relação aos requisitos definidos.

O método de comparação proposto por Lara, inicia-se pelo estabelecimento de critérios de medida que serão usados para avaliar os pacotes de *software*. Na definição dos níveis de pontuação faz-se a quantificação de quanto a ferramenta satisfaz os requisitos e, por fim, no julgamento, ocorre a comparação das pontuações alcançadas por cada ferramenta e a escolha daquela que melhor se enquadra às necessidades do cliente.

É importante destacar que este trabalho se diferencia do proposto por Lara ao aplicar o SQFD e as informações fornecidas por profissionais da área de teste com experiência no desenvolvimento de sistemas de informação. Contudo, a forma de avaliação proposta por Lara é preservada e utilizada para avaliar as ferramentas de teste, definindo as métricas de avaliação, os níveis de pontuação e a forma de julgamento.

Outro trabalho relacionado [Dórea 2008] aborda a avaliação de ferramentas de automação para engenheiros de testes a partir do conhecimento empírico das ferramentas. O trabalho não expõe quais mecanismos das ferramentas foram utilizados para elaborar as conclusões e nem um roteiro para a avaliação. De modo geral, a avaliação foi totalmente empírica e dificilmente reproduzível com os mesmos resultados por outras pessoas. Dessa forma, não existe no trabalho uma quantificação das características das ferramentas, bem como da funcionalidade que os profissionais da área desejam obter, requisitos esses que o QFD tem como objetivo primordial. O trabalho proposto neste artigo, diferente do mencionado anteriormente, possui um embasamento técnico-científico capaz de justificar

todas as etapas do método ao destacar referências, estudos anteriores e artefatos utilizados no processo de avaliação. Isso torna o trabalho mais robusto e consistente, facilitando sua utilização em caráter industrial.

3. Metodologia

3.1 SQFD (*Software Quality Function Deployment*)

A adaptação do QFD para o desenvolvimento de *software* originou a técnica do SQFD. O objetivo do *Software Quality Function Deployment* (SQFD) é melhorar o desenvolvimento de *software* aplicando técnicas de melhoria de qualidade, principalmente durante a especificação de requisitos. Para isso, as necessidades dos clientes são confrontadas com as restrições próprias do projeto, de forma a concentrar os melhores esforços nos aspectos com maior importância [Krogstie 1999].

Essa abordagem propõe o desenvolvimento de *software* centrado no usuário, onde a participação dos usuários do sistema é um fator indispensável para se determinar os fatores de qualidade do produto final.

O levantamento feito da aplicação do SQFD nas empresas mostra que a técnica permite a obtenção de benefícios consideráveis, destacando-se a redução de custos, a definição mais rápida das características do produto e quantificação qualitativa dos requisitos do cliente, além do registro de informações para a equipe de desenvolvimento e para a manutenção [Haag 1996]

A Matriz de Qualidade tradicional utilizada no QFD foi adaptada também para o SQFD, sendo aplicada na tradução da voz do cliente (o quê fazer), em requisitos técnicos (como fazer), e prioriza essas últimas de acordo com a sua importância.

Neste trabalho é utilizada uma lista encontrada no Anexo A. Ela contém os aspectos técnicos priorizados para ferramentas de teste, obtidos a partir da aplicação do SQFD por uma equipe de profissionais ligados ao desenvolvimento de sistemas de informação [Santos 2009]. O cliente, nesse caso, foi essa equipe, formada por profissionais de Belo Horizonte (MG), ligados ao desenvolvimento de sistemas de informação, composta por arquitetos de *software*, testadores, gerentes de projeto, engenheiros de requisitos e engenheiros de usabilidade, totalizando 17 pessoas envolvidas. A aplicação do SQFD foi feita seguindo cinco passos detalhados a seguir e que podem ser acompanhados na Figura 1 através dos pontos numerados [Haag 1996]:

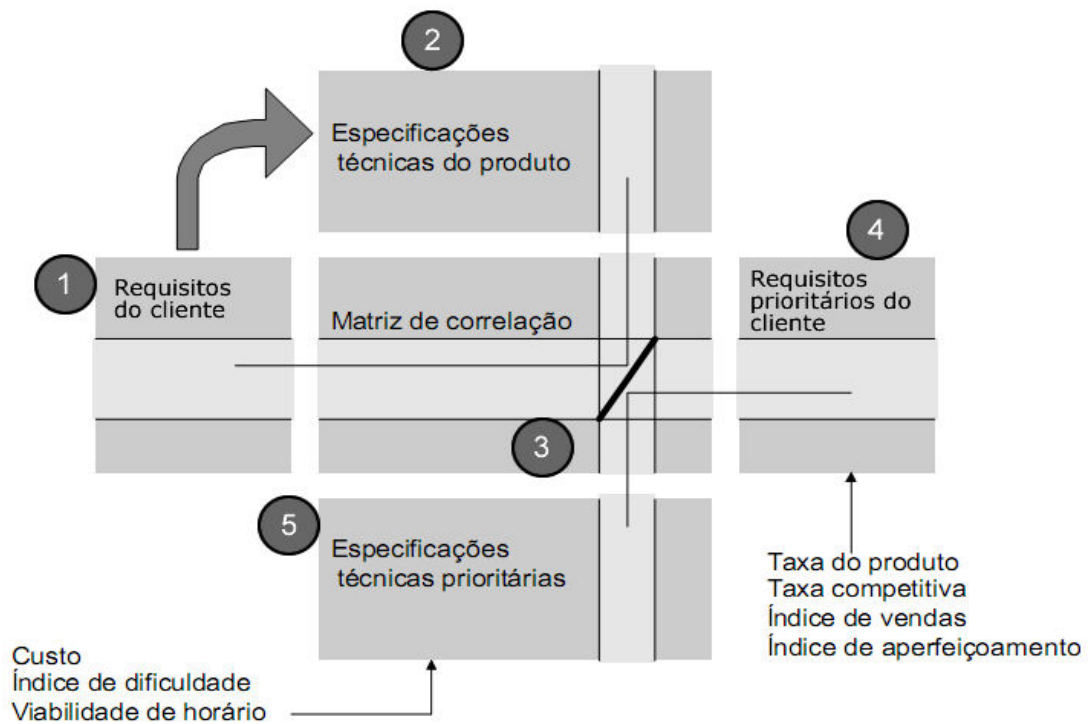


Figura 1. Modelo básico do SQFD

- Passo 1:** realização do levantamento dos requisitos (qualidades exigidas) do cliente. A partir de entrevistas, enquetes ou até mesmo técnicas de levantamento de idéias como o *Brainstorm*, as necessidades dos clientes devem ser identificadas. Perguntas como: “que tipo de ferramenta de teste você gostaria que existisse?” e “quais foram os problemas que você já enfrentou usando ferramentas de testes de *software*?”, podem ser utilizadas para auxiliar o levantamento dos requisitos. Após a identificação dos requisitos, eles são organizados em grupos hierárquicos, agrupando os que possuem conteúdo semelhante. Por último, para cada grupo deve ser definida uma expressão que traduza o conteúdo de todos os seus elementos. Considerando o exemplo de uma empresa interessada em adquirir alguma ferramenta que auxilie a realização de testes funcionais, o primeiro passo corresponde ao levantamento de requisitos para tais ferramentas, levantadas junto à equipe de teste, e agrupamento desses requisitos, para construção da Tabela de Desdobramento das Qualidades Exigidas. A Figura 2 mostra em (a) um exemplo dessa tabela. Nesse caso, foram identificados quatro requisitos, organizando-os em dois agrupamentos.
- Passo 2:** em cooperação com os usuários do produto, os requisitos são convertidos para especificações técnicas mensuráveis, que devem ser registradas na Tabela de Desdobramento dos Elementos da Qualidade. Essas especificações técnicas correspondem às funções que as ferramentas de testes devem conter para atender aos requisitos levantados. É importante notar que alguns requisitos podem ser “convertidos” para múltiplas especificações técnicas do produto. Após reuniões com a equipe de teste, esta poderia entrar num consenso de que para atender ao requisito “*Geração automática de testes funcionais*”, por exemplo, a ferramenta deveria ter as seguintes funções: *interpretador de condições modeladas em uma linguagem formal*, *gerador de entradas utilizando critérios e oráculo para gerar as*

saídas esperadas. A Figura 2 mostra em (b) os aspectos técnicos que poderiam ser levantados com base nos requisitos apresentados na parte (a). Vale notar que da mesma forma que os requisitos, os aspectos técnicos foram organizados em dois níveis hierárquicos.

Nível 1	Nível 2	Grau de Importância
Apoio a Execução dos testes	Agendamento da execução dos testes	
	Execução automática	
Apoio para preparação dos testes	Geração automática de testes funcionais	
	Geração de dados para execução de um teste	

(a)

Nível 1	Nível 2	Grau de Importância
Executor de testes	Agrupador e escalonador de testes	
	Executor de teste com possibilidade de pausa, cancelamento, anotação e retomada da execução	
	Gerador de log de execução de testes	
Gerador de Dados	Gerador de entradas utilizando critérios	
	Oráculo para gerar as saídas esperadas	
	Interpretador de condições modeladas em uma linguagem formal	
	Povoador de dados	

(b)

Figura 2. a) Tabela de Desdobramento das Qualidades Exigidas. b) Tabela de Desdobramento dos Elementos da Qualidade.

Passo 3: a partir de perguntas aos clientes, é criada uma matriz de correlação identificando os pesos dos relacionamentos entre os vários requisitos do cliente e as especificações técnicas do produto. O grau de correlação pode ser “*Possível*” (com valor 1), “*Fraca*” (com valor 3), “*Forte*” (com valor 9) ou pode não existir (em branco). Nesse momento a equipe de teste deve se reunir e construir a Casa de Qualidade, uma das matrizes mais utilizadas no QFD. Ela é construída a partir da combinação da Tabela de Desdobramento das Qualidades Exigidas e da Tabela de Desdobramento dos Elementos de Qualidade [Akao 1996]. A parte destacada na Figura 3 ilustra a Casa de Qualidade que poderia ser construída com base nas tabelas ilustradas na Figura 2. Nela podemos observar que o requisito *Agendamento da execução dos testes* possui correlação “*Forte*” com o aspecto técnico *Agrupador e escalonador de testes*; correlação “*Possível*” com *Gerador de log de execução de testes*; “*Fraca*” com *Povoador de dados*; e não possui correlação com os demais aspectos técnicos do exemplo.

Nível de correlação	
vazio	- sem correlação
1	- possível
3	- fraco
9	- forte

Nível 1	Nível 2	Nível 1						Grau de Importância	
		Agrupador e escalonador de testes	Executor de teste com possibilidade de pausa, cancelamento, anotação e retomada da execução	Gerador de log de execução de testes	Gerador de entradas utilizando critérios	Oráculo para gerar as saídas esperadas	Interpretador de condições modeladas em uma linguagem formal		Povoador de dados
Apoio a Execução dos testes	Agendamento da execução dos testes	9		3				1	1
	Execução automática		9	9				3	3
Apoio para preparação dos testes	Geração automática de testes funcionais				9	9	9		3
	Geração de dados para execução de um teste				9	9	9	3	2
Grau de Importância		9	27	30	45	45	45	16	
Grau de Importância (%)		4,15	12,44	13,82	20,74	20,74	20,74	7,37	
Necessidades dos Clientes (%)		30,41			69,59				

Figura 3. Casa de Qualidade com os requisitos e aspectos técnicos priorizados.

- Passo 4:** ainda com a ajuda dos clientes deve ser feita a priorização dos requisitos levantados. Uma forma de se fazer isso é solicitando para cada cliente que participou da identificação de necessidades, a atribuição de um peso (uma nota) para cada requisito, gerando assim o seu grau de importância. Pode-se utilizar como valores para o peso um número de um (1) a três (3), sendo três o mais importante. Em seguida a média dos valores atribuídos por cada participante é usada como o grau de importância do requisito em questão. A Figura 4 ilustra como pode ser feita a priorização dos requisitos da Figura 2, considerando que apenas três testadores participaram do levantamento de requisitos.

Requisitos	Cliente 1	Cliente 2	Cliente 3	Média	Grau de Importância
Agendamento da execução dos testes	1	1	2	1,33	1
Execução automática	3	3	3	3	3
Geração automática de testes funcionais	3	3	2	2,67	3
Geração de dados para execução de um teste	1	3	2	2	2

Figura 4. Priorização dos requisitos

- **Passo 5:** as especificações técnicas são priorizadas com base no somatório das multiplicações dos pesos dos requisitos pelos graus de intensidade da correlação entre os requisitos e as especificações técnicas. Logo, esse cálculo pode ser feito em três etapas: (1) para cada aspecto técnico, verificar quais são os requisitos com os quais ele possui alguma correlação; (2) multiplicar o grau de importância do requisito pelo grau de correlação com o aspecto técnico em questão; (3) somar os valores obtidos, resultando no peso (prioridade) do aspecto técnico. A Figura 3 completa, incluindo a parte destacada, mostra a Casa de Qualidade com a priorização dos requisitos e aspectos técnicos. O cálculo do aspecto técnico *Agrupador e escalonador de testes*, por exemplo, é igual 9 (grau de correlação dele com o requisito *Agendamento da execução dos testes*) x 1 (grau de prioridade do requisito). Além disso, os graus de importância são convertidos em valores percentuais e para cada grupo do nível 1 os valores obtidos na penúltima linha da matriz são somados resultando nas porcentagens da última linha.

3.2 Processo de Avaliação de Ferramentas

O processo de avaliação de ferramentas de testes descrito nesta seção foi criado com base no procedimento proposto por [Lara 2001], para escolha de um pacote de *software* na área de Sistemas de Informação. Nesse procedimento são utilizadas a norma NBR 12119 [NBR 12119 1998] da Associação Brasileira de Normas Técnicas (ABNT) e o *Quality Function Deployment* (QFD) do *American Supplier Institute* (ASI). Da mesma forma do procedimento apresentado por Lara, este trabalho necessita da definição de métricas de avaliação, níveis de pontuação e forma julgamento para que a avaliação seja possibilitada.

Definir as métricas de avaliação consiste em estabelecer os critérios de medida que serão usados para avaliar o atendimento as características de qualidade identificadas [Lara 2001]. No caso das ferramentas de testes, isso pode ser feito a partir da elaboração de questionários para cada grupo de aspectos técnicos identificados. Considerando a Figura 2, por exemplo, isso significa que para os dois grupos (*Executor de testes* e *Gerador de Dados*) deve existir um conjunto de questões que serão usadas para avaliar o atendimento aos aspectos técnicos reunidos nesses grupos. Em seguida, os níveis de pontuação devem ser definidos, pois eles que serão usados para demonstrar o quanto a ferramenta satisfaz os requisitos. Vários níveis podem ser usados, mas este trabalho considera o uso de apenas três: “0”, indicando que a ferramenta não atende o requisito avaliado; “5”, no caso da ferramenta atender satisfatoriamente o requisito avaliado; “10”, que indica o perfeito atendimento ao quesito avaliado. Contudo, dependendo das questões elaboradas podem ser utilizados apenas “0” ou “10”.

A Figura 5 ilustra um questionário que poderia ser utilizado para verificar o atendimento aos aspectos técnicos do grupo “*Executor de testes*” (Figura 2). Com o questionário pronto, o passo seguinte é a análise das ferramentas de testes a partir do uso prático. Para cada questão avaliada é necessário escolher uma resposta dentre as possíveis. Em seguida, os avaliadores devem somar as notas atribuídas em cada item para obter a “Nota” da ferramenta. Por fim, deve-se transformar esse valor em percentual, dividindo-se o valor da “Nota” pela “Nota Máxima. Utilizando como exemplo a Figura 5, supondo que uma ferramenta obteve “sim” (10) nas cinco primeiras perguntas, “às vezes” (5) na última e “não” (0) nas demais, e observando que a nota máxima é 80 (8 perguntas x 10), o resultado será 55/80 (68,75%).

Nível Avaliado	Nota	Nota Máxima	Resultado (%)
Executor de Testes			
<i>Agrupador e Escalonador de Testes</i>			
A ferramenta permite escalonar quais casos de testes serão executados dentre uma bateria de testes?		Não (0) Sim (10)	
A ferramenta permite agrupar os casos de testes?		Não (0) Sim (10)	
<i>Executor de teste com possibilidade de pausa, cancelamento, anotação e retomada da execução</i>			
A ferramenta permite parar a execução do teste a qualquer instante?		Não (0) Sim (10)	
A ferramenta permite reiniciar a execução dos testes após a execução ser cancelada?		Não (0) Sim (10)	
A ferramenta apresenta a possibilidade de adicionar comentários aos casos de testes criados?		Não (0) Sim (10)	
<i>Gerador de log de execução dos testes</i>			
A ferramenta gera log de execução no formato HTML com visualização gráfica?		Não (0) Sim (10)	
A ferramenta gera log de execução com estatísticas?		Não (0) Sim (10)	
O log de execução da ferramenta permite a identificação da causa de possíveis problemas?		Não (0) Às vezes (5) Sim (10)	

Figura 5. Questionário para avaliação do atendimento aos requisitos dos clientes.

Por fim, para o julgamento das ferramentas de testes é necessário elaborar um quadro comparativo das necessidades do cliente com relação aos níveis de pontuação obtidos das avaliações das ferramentas. Para isso seguimos os seguintes passos:

- Transcrever as necessidades do cliente em porcentagem para a coluna “Necessidades do cliente”. Essas porcentagens estão presentes na última linha da Casa de Qualidade que fora previamente construída.
- Transcrever para a coluna “Resultado” as pontuações obtidas por cada ferramenta avaliada no questionário com respeito a cada grupo de aspectos técnicos.
- Realizar a multiplicação desses valores (“Necessidades do cliente” e “Resultado”) preenchendo a coluna “Proporção”. Os valores dessa coluna representam a porcentagem da necessidade do cliente atendida pela ferramenta, com relação a cada grupo de aspecto técnico analisado.
- Obter a porcentagem de satisfação dos requisitos somando-se os valores da coluna “Proporção”, para cada ferramenta analisada.

A ferramenta que alcançar a pontuação mais próxima de 100% tem maiores chances, em relação às outras, de atender aos requisitos de qualidade esperados e assim de se tornar a ferramenta mais adequada. Considerando, por exemplo, as 8 perguntas da Figura 5 para o grupo “Executor de teste” e que foram criadas outras perguntas para avaliar o atendimento aos aspectos técnicos do grupo “Gerador de dados”, pode-se então fazer a avaliação de duas ferramentas de testes.

Se os resultados da avaliação delas forem, respectivamente, 70% e 30% para o para o grupo “Executor de testes”, e 40% e 80% para “Gerador de dados”, então, teríamos como resultado a Tabela 1. Com base nela, pode-se afirmar que, levando em conta as necessidades identificadas e a forma de avaliação aplicada, a ferramenta Y é a mais adequada, pois atende a 65% das necessidades do cliente.

Tabela 1. Quadro comparativo entre duas ferramentas de teste X e Y

Características de Qualidade das ferramentas de testes funcionais	Necessidades do cliente	Avaliação da ferramenta X		Avaliação da ferramenta Y	
		Resultado	Proporção	Resultado	Proporção
Executor de testes	30,0%	70,0%	21,0%	30,0%	9,0%
Gerador de dados	70,0%	40,0%	28,0%	80,0%	56,0%
Satisfação dos Requisitos	100,0%		49,0%		65,0%

4. Resultados

Utilizamos o método proposto neste trabalho para auxiliar a avaliação de ferramentas para o desenvolvimento de testes funcionais. Foram analisadas três ferramentas: Selenium [SELENIUM 2010], Badboy [BADBOY 2010] e Canoo [CANOO 2010]. Essas ferramentas são normalmente utilizadas para o teste de sistemas de informação para a Web. O questionário utilizado foi baseado no trabalho já citado que utilizou o SQFD para obter requisitos e aspectos técnicos para ferramentas de teste (Anexo A). Esse questionário é bastante extenso e cobre todos os aspectos desejados pelos profissionais entrevistados para ferramentas de teste, independente de categoria. Por conta do tamanho do questionário não foi possível sua inclusão neste artigo, mas uma versão dele encontra-se disponível para consulta a partir do endereço <http://www.ufpi.br/pasn>. É importante ressaltar que alguns agrupamentos não foram utilizados para avaliação das ferramentas de teste funcional, devido a ausência de correlação com tal categoria de ferramentas.

Tabela 2. Quadro comparativo entre as ferramentas de testes funcionais Selenium, Badboy e Canoo

Grupo	Necessidade do cliente (%)	Selenium		BadBoy		Canoo	
		Resultado (%)	Proporção (%)	Resultado (%)	Proporção (%)	Resultado (%)	Proporção (%)
Gerador de planos de teste	9,58	0,00	0,00	0,00	0,00	0,00	0,00
Gerador de dados	6,99	0,00	0,00	0,00	0,00	0,00	0,00
Gerador de testes funcionais	11,69	0,00	0,00	0,00	0,00	0,00	0,00
Gerador manual de teste	7,91	90,00	7,12	40,00	3,16	50,00	3,96
Rastreador	3,05	0,00	0,00	0,00	0,00	0,00	0,00
Integrador	3,52	0,00	0,00	0,00	0,00	0,00	0,00
Modelo de teste	12,88	0,00	0,00	0,00	0,00	0,00	0,00
Avaliador de testes	1,82	25,00	0,46	0,00	0,00	25,00	0,46
Gerador de testes não-funcionais	0,70	0,00	0,00	0,00	0,00	0,00	0,00
Gerador de relatórios	4,57	0,00	0,00	12,50	0,57	25,00	1,14
Suporte da ferramenta	8,07	0,00	0,00	0,00	0,00	0,00	0,00
Arquitetura da ferramenta	3,00	100,00	3,00	91,67	2,75	66,67	2,00
Auxilio da ferramenta	2,63	75,00	1,97	75,00	1,97	58,33	1,53
Executor de teste	14,70	44,74	6,58	34,21	5,03	42,10	6,19
Satisfação do cliente	100		19,12		13,49		15,28

Uma equipe composta por três membros utilizou as três ferramentas e, em conjunto, atribuíram respostas ao questionário. Embora tenha havido alguma discordância em alguns pontos, a grande maioria das respostas foi consensual. Assim, o resultado obtido reflete a adequação das ferramentas aos requisitos expostos pelo SQFD. A Tabela 2 ilustra o resultado da avaliação realizada. Com base nela, pode-se notar que a ferramenta Selenium se adequou mais as necessidades identificadas ao conseguir 19,12%, em relação às outras duas: Badboy, com 13,49% e Canoo, com 15,28%.

5. Conclusão

Este artigo apresenta um método para avaliação de ferramentas de teste de *software* baseado no uso do SQFD. As características do SQFD permitem focar nos aspectos considerados mais importantes para os profissionais de teste, tornando o resultado da avaliação mais relevante.

O método proposto neste trabalho possibilita a comparação entre diversas ferramentas existentes no mercado, podendo ser usado para a identificação de qual ferramenta de teste atende melhor às necessidades de uma empresa. Como consequência, ao utilizar ferramentas mais apropriadas, os testadores terão maior produtividade, reduzindo assim o tempo gasto na realização dos testes. Além disso, avaliar as ferramentas de testes antes de adquiri-las pode evitar prejuízos provenientes do uso de ferramentas inadequadas.

Para validar o método proposto foi realizada a avaliação de três ferramentas de testes funcionais: Selenium, Badboy e Canoo. A avaliação demonstrou que a ferramenta Selenium é a mais adequada, levando em conta as necessidades identificadas pelos participantes da etapa de levantamento de requisitos realizada durante este trabalho.

Como trabalho futuros, estamos refinando os questionários e realizando análises de diferentes tipos de ferramentas de teste, para identificarmos com mais clareza a adequabilidade do mesmo. Além disso, estaremos estudando a aplicação de um limiar de corte para definirmos até que ponto uma ferramenta supera a outra quando os resultados dos julgamentos apontarem valores próximos.

Agradecimentos

Este trabalho recebeu apoio financeiro do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

Referências

- Akao, Y. (1996) Introdução ao Desdobramento da Qualidade. Tradução de Zelinda Tomie Fujikawa Belo Horizonte: Fundação Chistiano Ottoni, 187p.
- BADBOY. (2010) Disponível em <<http://ww.badboy.com.au/>>
- CANOO. (2010) Disponível em <<http://webtest.canoo.com/>>
- Dórea, Alexsandro D. de Oliveira, Carvalho, F. de S., Santos, M. T., Neto, M. C. M., Moises, D. (2008) "Avaliação de Ferramentas de Automação para Engenheiros de Testes". In: IV Simpósio Brasileiro de Sistemas de Informação, 2008, Rio de Janeiro. Anais do IV Simpósio Brasileiro de Sistemas de Informação. Rio de Janeiro: SBC. v. 1. p. 23-34.

- Haag , S., Raja, M. K., Schkade, L. L. (1996) Quality function deployment usage in *software* development. Communications of the ACM, 39(1):41–49, January.
- Harrold, M.J.(2000) Testing: Aroadmap. In The Future of *Software* Engineering, pages61–72. ACM Press.
- Krogstie, J.(1999) Using Quality Function Deployment in *Software* Requirements Specification. Anais do REFSQ'99. Heidelberg, Alemanha.
- Lara, J.F.(2001) "Um procedimento para escolha de pacote de *software* na área de sistemas de informação". In VI Simpósio de Teses e Dissertações Defendidas. ICMC-USP.
- Myers, G. J.(2004) The Art of *Software* Testing. Jonh Wiley & Sons. 2ª edição.
- NBR ISO/IEC 12119. (1998) Tecnologia de Informação – Pacotes de software – Teste e requisitos de qualidade.
- SELENIUM. (2010) Disponível em <<http://seleniumhq.org/>>
- Veloso, J de.S., Santos, I. S., Santos Neto, P. (2009) “Aplicação do SQFD na escolha de uma ferramenta de apoio ao teste de *software*”. In ERCEMAPI – Encontro Regional de Computação Ceará – Piauí – Maranhão, Parnaíba-PI, outubro.
- Weber, K. (2001) Qualidade de *Software*: Teoria e Prática. Prentice Hall, São Paulo.

ANEXO A – Tabela de aspectos técnicos priorizados e seus graus de importância para os testadores.

Agrupamento	Aspecto Técnico	Peso (%)	Total (%)						
Gerador de plano de teste	Gestão de dados do plano de teste	1	9,58	Gerador de testes não-funcionais	Verificador de arquitetura	1,25	4,94		
	Estimador de complexidade de caso de uso	1,41			Gerador de testes de desempenho e estresse	1,06			
	Estimador de prazo para execução de tarefa de teste	1			Integração com ferramentas para manipulação de requisições e respostas em transações	0,7			
	Base de dados histórica de projetos	2			Integração com sistema de varredura de portas	0,7			
	Calculador de produtividade	1,47			Gerador de testes de segurança	1,23			
	Gerador de dados	Registro de tarefas		1,23	6,99	Gerador de Relatórios	Gerador de relatório com formato definido pelo usuário	1,58	4,57
		Alocador inteligente de tarefa		0,65			Gerador de gráficos com fonte de dados e formato definido pelo usuário	1,06	
		Integração com ferramentas de gerenciamento de projetos		0,82			Acesso a qualquer informação de teste existente no modelo de testes	1,58	
	Gerador de objetos	3,29		Gerador de entradas utilizando critérios	3,29		Uso de hiperlinks e agrupadores nos relatórios	0,35	
Gerador de testes funcionais	Gerador de povoador	0,88	11,69	Suporte da ferramenta	Cadastro de usuários	1,84	8,95		
	Gerador de valores	2,82			Cadastro de grupos	1,84			
	Gerador de entradas utilizando critérios	3,29			Cadastro de projeto	2,02			
	Oráculo para gerar as saídas esperadas	2,47			Cadastro de equipe	2,02			
Interpretador de condições modeladas em uma linguagem formal	3,29	Mecanismo de busca para qualquer padrão informado pelo usuário	0,35						
Extrator de dados de modelos descrevendo o sistema	2,64	Cadastro de perfis com possibilidade definição de permissões por grupo, projeto, equipe	0,88						
Gerador manual de testes	Mecanismo de captura-reprodução	0,92	7,91	Arquitetura da ferramenta	Uso de <i>software</i> livres	0,47	3		
	Gerador de teste com apoio da especificação	0,78			Uso de tecnologia Web	1,12			
	Uso de linguagens de alto nível	1,17			Seguir um guia de estilo	0,53			
	Acesso às funções do SO	2,52			Utilizar terminologia adequada ao contexto	0,88			
	Acesso ao mecanismo de persistência	2,52		Auxílio da ferramenta	Help on-line	0,7	2,63		
Rastreador	Integração com ferramentas de gestão de requisitos	2,52	3,05	Integrador	Manual de usuário	0,7			
	Detector de alterações entre as visões, exibindo partes possivelmente afetadas.	0,53			Sítio de apoio com exemplos de uso	0,7			
Integrador	Integração com ferramentas de acompanhamento de bugs	1,76	3,52	Modelo de teste	Curso de formação	0,53			
	Integração com ferramentas de gestão de configuração	1,76			Representação dos testes e do sistema utilizando um modelo independente de tecnologia	3,83	14,52		
Modelo de teste	Regras de transformação entre modelos	3,35	14,52	Executor de teste	Executor de teste com possibilidade de pausa, cancelamento, anotação e retomada da execução	1,76		16,82	
	Especificação de modelos para descrição do sistema	2,86			Agrupador e escalonador de testes	1,25			
	Adoção de uma linguagem para especificação de restrições no modelo	2,84			Gerador de log de execução de testes	1,64			
	Utilização de um padrão para uso de elementos descrevendo a arquitetura de <i>software</i>	1,64			Simulador de interfaces de hardware e <i>software</i>	1,06			
	Avaliador de testes	Avaliador de cobertura			1,47	1,82			Integração com uma linguagem de script para configuração do ambiente de teste
		Analisador de qualidade de testes baseado em mutação			0,35		Comparador de arquivos ignorando padrões configuráveis		1,23
Avaliador de testes						Povoador de dados	2,11		
						Gerenciador de transações	3,17		
						Analisador de falhas	0,57		
						Cadastramento automático de falhas	0,80		
						Simulador de plataformas de hardware	1,06		
						Simulador de banda de rede	1,06		