

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 de Sousa Britto

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

CEP 64049-550 – Ininga – Teresina – PI - Brasil
{janielton, pasn, ismayle, rbritto}@ufpi.edu.br

Abstract. *Software testing is the main approach used for assurance of information system quality. 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 an evaluation of the main free functional and performance testing tools using a mechanism that allows comparative analysis of the tools to support software testing demands.*

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

1. Introdução

No processo de desenvolvimento de sistemas de informação, 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]. Além disso, a escolha de ferramentas de testes inadequadas pode trazer desperdício de tempo, fator relevante no desenvolvimento de *software*, além de influenciar negativamente no sucesso do produto caso ela, por exemplo, não tenha sido desenvolvida para identificar os defeitos que tenhamos o propósito de encontrar.

Dessa forma, é importante que existam meios para se avaliar e comparar ferramentas de teste. Para isso, os requisitos da equipe de teste, com relação a ferramentas, devem ser identificados e priorizados, de forma a garantir que aquilo que é mais importante, na visão dos testadores, será considerado. Um método bastante difundido para se identificar e priorizar requisitos é o QFD (*Quality Function Deployment*). O QFD pode ser usada para traduzir as necessidades dos clientes em requisitos técnicos de produto [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].

Neste trabalho é apresentado uma evolução do método baseado no SQFD para analisar e comparar ferramentas de testes para sistemas de informação, proposto no trabalho de Veloso [Veloso 2009]. Além de fornecer aos desenvolvedores de ferramentas de testes um mecanismo para avaliá-las com relação aos concorrentes, o método proposto auxilia as empresas e os testadores a escolher as ferramentas de testes mais adequadas para atender às suas necessidades. Para validar o método proposto, foi feita a avaliação das principais ferramentas de testes de sistemas de informação gratuitas e os resultados dessa avaliação, bem como uma discussão sobre o próprio método, são apresentados neste artigo.

O restante deste artigo está organizado da seguinte forma: na Seção 2 são apresentados alguns trabalhos relacionados; na Seção 3 é apresentado o SQFD, bem como o método de avaliação de ferramentas de testes de sistemas de informação; na Seção 4 é feita uma discussão sobre a aplicação do método e são apresentados os resultados obtidos na avaliação das ferramentas 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* [Karlson 1997, Alves e Padua 2001, Ioannou et al. 2004]. 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 apresentado neste artigo é o trabalho de Lara [Lara 2001]. Ela especifica 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*. Além disso, ela utiliza 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* na área de sistemas de informação em relação aos requisitos definidos.

O método de comparação proposto por Lara se inicia pelo estabelecimento de critérios de medida, os quais são usados para avaliar os pacotes de *software*. Na definição dos níveis de pontuação faz-se a quantificação de quanto a ferramenta satisfaz os requisitos. 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 o método utilizado neste trabalho se diferencia ao aplicar o SQFD juntamente com 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 utilizada para avaliar as ferramentas de teste através da definição das métricas de avaliação, dos níveis de pontuação e da 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. 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

Esta seção tem por objetivo detalhar os passos do SQFD, bem como apresentar o método de avaliação de ferramentas de testes de sistemas de informação que foi utilizado neste trabalho.

3.1 SQFD (*Software Quality Function Deployment*)

A adaptação do QFD para o desenvolvimento de *software* deu origem ao 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, e com isso se pode 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.

Segundo Haag [Haag 1996], a 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.

Vale destacar que a Matriz de Qualidade tradicional utilizada no QFD também foi adaptada para o SQFD. Esta é 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.

A aplicação do SQFD é feita seguindo os cinco passos numerados na Figura 1: 1) levantamento dos requisitos do cliente; 2) Identificação das especificações técnicas do produto a partir dos requisitos que foram levantados; 3) Criação da matriz de correlação entre os requisitos do cliente e as especificações técnicas do produto; 4) Priorização dos requisitos do cliente; 5) Priorização das especificações técnicas. A seguir tem-se um detalhamento destes passos com base no que é apresentado em [Haag 1996].

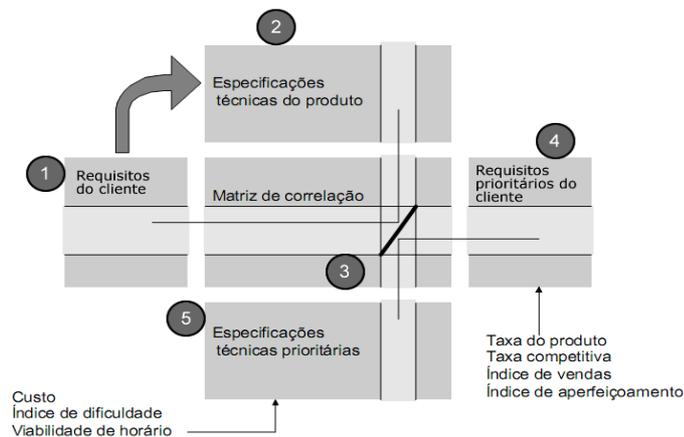


Figura 1. Modelo básico do SQFD adaptado de [Haag 1996]

- Passo 1:** a primeira etapa corresponde à 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 (níveis), 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 ferramentas que auxiliem a realização de testes funcionais de um determinado sistema de informação, o primeiro passo corresponde ao levantamento de requisitos para tais ferramentas junto à equipe de teste e o agrupamento desses requisitos para construção da Tabela de Desdobramento das Qualidades Exigidas. A Figura 2 mostra em (a) um exemplo dessa tabela. Nesse caso, têm-se quatro requisitos organizados 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 no 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
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	

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) (b)

Figura 2. Exemplos de a) Tabela de Desdobramento das Qualidades Exigidas. b) Tabela de Desdobramento dos Elementos da Qualidade adaptados de [Veloso 2009].

- 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), “Fraco” (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.
- Passo 4:** ainda com a ajuda dos clientes deve ser feita a priorização dos requisitos levantados. Uma forma de se fazer isso é solicitar para cada cliente que participou da identificação das 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 clientes (testadores) participaram do levantamento de requisitos.

		Nível 1							Grau de Importância
		Nível 2							
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. Exemplo de Casa de Qualidade com os requisitos e aspectos técnicos priorizados adaptado de [Veloso 2009].

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. Exemplo de priorização dos requisitos adaptado de [Veloso 2009]

- 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 (Fórmula 1). 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 a **9** (grau de correlação dele com o requisito *Agendamento da execução dos testes*) x **1** (grau de prioridade do requisito) = **9**. 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.

$$Pa = \sum_{r=1}^n (Pr * Cra) \quad (1)$$

Onde:

- Pa é o peso do aspecto técnico;
- Pr é o peso do requisito;
- Cra é o grau de correlação entre o requisito e o aspecto técnico;
- n é o número de requisitos identificados.

3.2 Método de avaliação de ferramentas de testes de sistemas de informação

Em um trabalho prévio, foi apresentado um método de avaliação de ferramentas de testes de sistemas de informação [Veloso 2009] que foi criado com base no procedimento proposto por Lara [Lara 2001] para escolha de um pacote de *software* na área de sistemas de informação. O método apresentado [Veloso 2009] se diferencia por utilizar como base informações provenientes dos usuários finais das ferramentas. Para isso, requisitos foram levantados e priorizados de acordo com os passos do SQFD, conforme descrito anteriormente. Contudo, da mesma forma do procedimento apresentado por Lara, esse trabalho utiliza-se da definição de métricas de avaliação, níveis de pontuação e forma de julgamento para avaliação das ferramentas.

3.2.1 Definição de métricas de avaliação

Definir as métricas de avaliação consiste em estabelecer os critérios de medida que serão usados para avaliar o atendimento às 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. É importante salientar que pode existir mais de uma pergunta para cada aspecto técnico.

3.2.2 Definição dos níveis de pontuação

Após a definição das métricas de avaliação, os níveis de pontuação devem ser definidos, pois eles serão usados para demonstrar o quanto a ferramenta satisfaz os requisitos. Diferentemente do utilizado em [Veloso 2009], neste trabalho foi considerado o uso de três níveis: “0” (Não), indicando que a ferramenta não atende o requisito avaliado; “0,5” (Satisfatório), no caso da ferramenta atender satisfatoriamente o requisito avaliado; “1”, que indica o perfeito atendimento ao quesito avaliado. Contudo, dependendo das questões elaboradas podem ser utilizados apenas “0” ou “1”.

3.2.2 Definição da forma de julgamento

Tendo definido o questionário e os níveis de pontuação, o passo seguinte é a definição da forma de julgamento. O aperfeiçoamento no método apresentado em [Veloso 2009] proposto neste trabalho prescreve três etapas para o julgamento.

Na primeira etapa, deve-se utilizar na prática as ferramentas de teste de sistemas de informação a serem avaliadas.

Na segunda etapa, deve-se aplicar o questionário definido pelo método. Para cada questão do questionário aplicada a uma ferramenta de teste é necessário escolher uma resposta dentre as possíveis, que por conseguinte irá resultar em um dos três níveis de pontuação definidos no método. Para cada um dos aspectos técnicos, os quais são elencados com o uso do SQFD, deve existir apenas um valor resultante das respostas ao questionário. Se por ventura existir mais de uma pergunta no questionário para um determinado aspecto técnico, o avaliador deverá tirar a média das pontuações resultantes da resposta de cada pergunta.

Para melhor exemplificar esta etapa, vamos considerar a avaliação do grupo “Grupo 1” de uma dada ferramenta chamada de “Ferramenta 1” (Figura 5). Este grupo possui cinco aspectos técnicos. Para cada aspecto técnico existe uma pergunta vinculada no questionário criado para avaliar uma dada ferramenta, com exceção do “Aspecto Técnico 3”, que possui duas perguntas vinculadas no questionário. Considerando as respostas no questionário para a “Ferramenta 1”, teríamos as pontuações para os aspectos técnicos 1, 2, 4 e 5 respectivamente 1, 0, 1 e 0,5 (considerando os níveis de pontuação especificados). Uma vez que o “Aspecto Técnico 3” possui 2 perguntas vinculadas no questionário, a pontuação do aspecto técnico é o resultado da média das pontuações das perguntas 3 e 4, respectivamente 0,5 e 1, o que resulta em 0,75.

				Ferramenta 1	
				Questionário	
Grupos	Aspectos Técnicos	Perguntas	Níveis	Respostas	Pontuação
Grupo 1	Aspecto Técnico 1	Pergunta 1	Não		1
			Sim	x	
	Aspecto Técnico 2	Pergunta 2	Não	x	0
			Sim		
	Aspecto Técnico 3	Pergunta 3	Não		0,75
			Satisfatório		
		Pergunta 4	Sim	x	
			Não		
	Aspecto Técnico 4	Pergunta 5	Satisfatório	x	1
			Sim		
Aspecto Técnico 5	Pergunta 6	Não		0,5	
		Satisfatório	x		
			Sim		

Figura 5. Resultado da aplicação das perguntas de um questionário relacionadas ao “Grupo 1” da “Ferramenta 1”.

Na última etapa, para o julgamento das ferramentas de testes deve-se elaborar uma tabela onde são relacionadas as necessidades dos clientes e os valores decorrentes da aplicação do questionário em cada uma das ferramentas. Essa tabela é constituída pelas colunas “Grupos” (que descreve os grupos de aspectos técnicos), “Aspectos Técnicos” (enumera os aspectos técnicos elencados por meio do SQFD), “Pesos” (apresenta os níveis de necessidade do cliente para cada um dos aspectos técnicos), “Pontuação” (contém os

valores decorrentes da aplicação do questionário definido pelo método) e “Atendido” (apresenta o quanto da necessidade do cliente em cada grupo de aspectos técnicos é atendida pela ferramenta). Para preencher essa tabela, seguem-se os seguintes passos:

1. Transcrever os grupos de aspectos técnicos definidos por meio do SQFD para a coluna “Grupos”.
2. Transcrever as necessidades do cliente em porcentagem para a coluna “Pesos”. Essas porcentagens estão presentes na linha “Grau de Importância” da Casa de Qualidade que fora previamente construída.
3. Transcrever para a coluna “Pontuação” as pontuações obtidas para cada aspecto técnico, após a aplicação do questionário nas ferramentas avaliadas. Para cada ferramenta avaliada deve existir uma coluna “Pontuação”. É importante salientar que no caso de existir mais de uma pergunta no questionário para um determinado aspecto técnico, na coluna “Pontuação” deverá ser colocada a média das pontuações do conjunto de perguntas vinculadas ao aspecto técnico em questão.
4. Realizar a multiplicação de cada valor contido na coluna “Pesos” com os valores de igual posição na coluna “Pontuação”.
5. Somar os produtos do passo 4 que estejam dentro de um mesmo grupo de aspectos técnicos (visível por meio da coluna “Grupos”).
6. Colocar cada uma das somas de produtos do passo 5 na coluna “Atendido”. Para cada ferramenta avaliada deve existir uma coluna “Atendido”.

Para se determinar dentro um conjunto de ferramentas qual delas satisfaz mais a necessidade dos clientes, deve-se somar todas as linhas da coluna “Atendido” de cada uma das ferramentas.

Consideremos mais uma vez a “Ferramenta 1”. Para compor a tabela necessária para o julgamento deste exemplo (Figura 6), foram aproveitadas as colunas “Grupos”, “Aspectos Técnicos” e “Pontuação” da tabela apresentada na Figura 5. Juntou-se a essas colunas outras duas, a coluna “Pesos” (que no exemplo possui valores aleatórios) e a coluna “Atendido”. Após a construção da tabela, foram executados os cálculos descritos nos passos 4 e 5 e os resultados colocados na coluna “Atendido”. O “Grupo 1” representa 7,91% da necessidade total do cliente (encontrada somando todas as linhas da coluna “Peso”). Após o procedimento descrito, constatou-se que a “Ferramenta 1” atende a apenas 5,58 % da necessidade do cliente dos 7,91% possíveis para o “Grupo 1”, o que gera uma diferença de 2,33%.

			Ferramenta 1	
Grupos	Aspectos Técnicos	Pesos	Pontuação	Atendido
Grupo 1	Aspecto Técnico 1	0,92	1	5,58
	Aspecto Técnico 2	0,78	0	
	Aspecto Técnico 3	1,17	0,75	
	Aspecto Técnico 4	2,52	1	
	Aspecto Técnico 5	2,52	0,5	

Figura 6. Resultado do julgamento do “Grupo 1” da “Ferramenta 1”.

É importante salientar que a forma de julgamento apresentada neste trabalho é um aperfeiçoamento da apresentada em [Veloso 2009]. Tal aperfeiçoamento foi necessário uma vez que no método original poderiam existir distorções na avaliação das ferramentas.

O método anterior calculava o nível de necessidade do cliente atendido por determinada ferramenta utilizando os valores contidos na última linha da Casa de Qualidade

(Necessidade do Cliente) e os pontos advindos da aplicação do questionário. Os dados contidos na linha “Necessidade do Cliente” se referem a grupos de aspectos técnicos, e o questionário contém perguntas relacionadas a cada aspecto técnico individualmente, o que os coloca em níveis diferentes de granularidade. As distorções na avaliação de ferramentas poderiam ser geradas pelo fato do método calcular o nível de necessidade atendido por uma dada ferramenta multiplicando valores de granularidades diferentes. O aperfeiçoamento apresentado neste trabalho utiliza, para calcular a necessidade do cliente atendida por uma dada ferramenta, os valores contidos na linha “Grau de Importância” e os valores advindos da aplicação do questionário (utilizando a média caso exista mais de uma pergunta para um dado aspecto técnico). Uma vez que os valores utilizados estão em um mesmo nível de granularidade, a possibilidade de distorções na avaliação de ferramentas de teste por meio do método proposto foi removida.

4. Resultados

Objetivando analisar a situação atual das ferramentas de testes de sistemas de informação, além de avaliar os aperfeiçoamentos efetuados no método proposto em [Velooso 2009], foi realizada uma avaliação comparativa entre as principais ferramentas gratuitas de apoio ao teste funcional e ao teste de desempenho no contexto de sistemas de informação para *Web*. Foram selecionadas inicialmente essas duas categorias de testes devido à grande variedade de ferramentas existentes que apóiam estes testes, dificultando a escolha da ferramenta adequada.

O teste funcional é feito para verificar se as especificações funcionais (comportamento) foram corretamente implementadas [IEEE 2004]. As ferramentas de apoio a este teste são, em geral, ferramentas do tipo *record-and-playback*, ou seja, elas gravam as ações executadas pelo testador gerando um *script* dos passos executados. As ações gravadas podem então ser re-executadas e, além disso, o *script* pode ser alterado, permitindo uma alteração nos testes sem a necessidade de realizar nova gravação.

O teste de desempenho, por sua vez, verifica o atendimento aos requisitos de desempenho como tempo de resposta e quantidade de usuários simultâneos que o sistema pode atender [IEEE 2004]. Ferramentas de apoio ao teste de desempenho, em geral, são preocupadas em auxiliar o testador a montar requisições HTTP¹, as quais são submetidas ao sistema sob teste para verificar o tempo de resposta e quantas requisições o sistema consegue responder simultaneamente.

A seguir são apresentados os resultados da aplicação do SQFD na priorização dos aspectos técnicos de ferramentas de testes de sistemas de informação, bem como é descrita a forma como foram avaliadas as principais ferramentas de testes de sistemas de informação gratuitas. Além disso, é feito na sub-seção 4.3 uma discussão sobre a avaliação realizada e sobre o método de avaliação utilizado.

4.1 Aplicação do SQFD no contexto de ferramentas de testes de sistemas de informação

Conforme mencionado anteriormente, para aplicarmos o método é preciso fazer um levantamento das necessidades dos clientes utilizando o SQFD. Para a avaliação das ferramentas descritas neste trabalho, os dados da aplicação do SQFD utilizados foram extraídos de [Santos et al. 2010], onde o cliente foi uma equipe formada por profissionais de Belo Horizonte (MG). Os profissionais desta equipe eram ligados ao desenvolvimento de

¹ Hypertext Transfer Protocol

sistemas de informação, sendo a mesma composta por arquitetos de *software*, testadores, gerentes de projeto, engenheiros de requisitos e engenheiros de usabilidade, totalizando 17 pessoas envolvidas. A aplicação do SQFD no contexto de ferramentas de testes está descrita em detalhes em [Santos et al. 2010].

Foram levantados 43 requisitos e a partir deles e com a participação da equipe de teste foram identificados 67 aspectos técnicos. Foi então elaborada a matriz de correlação e os requisitos e aspectos técnicos foram priorizados, os quais podem ser vistos com mais detalhes em [Santos et al. 2010]. Esses aspectos técnicos correspondem às características que as ferramentas de testes devem ter para atender aos requisitos que foram levantados. Dentre os grupos de aspectos técnicos destacam-se os grupos Executor de testes com 16,82% e Modelos de testes com 14,52%. Isso demonstra um maior desejo dos testadores com respeito a funções que auxiliem a execução do teste e a ferramentas que utilizem modelos. Por outro lado, os grupos ligados à avaliação de testes (Avaliador de testes) e ao auxílio fornecido pela ferramenta (Auxílio da ferramenta) representam os aspectos técnicos menos relevantes na opinião dos testadores entrevistados.

4.2 Avaliação de ferramentas de testes de sistemas de informação

Para avaliar as ferramentas, utilizamos um questionário feito com base na tabela de aspectos técnicos priorizados extraída de [Velo 2009]. Para cada aspecto técnico foi elaborada uma ou mais perguntas, as quais verificam se a ferramenta de teste a ser avaliada atende ou não ao aspecto técnico em questão. O questionário é extenso e cobre todos os aspectos desejados pelos profissionais entrevistados para ferramentas de teste, independente de categoria. Por conta do tamanho do questionário apenas um excerto do questionário é exibido na Figura 7. É importante salientar que algumas perguntas admitem apenas “Não” ou “Sim” como resposta, enquanto outras aceitam também a opção “Satisfatório”, indicando um meio termo entre um totalmente satisfeito (Sim) e não satisfeito (Não). Os questionários completos encontram-se disponíveis para consulta a partir do endereço <http://www.ufpi.br/pasn>. É importante ressaltar que alguns agrupamentos não foram utilizados nas avaliações descritas nesta seção, devido à ausência de correlação com a categoria de ferramentas avaliada.

Nível Avaliado	Função/característica do produto	Perguntas
Avaliador de testes	Avaliador de cobertura	A ferramenta se integra com ferramentas de avaliação de cobertura, permitindo a obtenção do grau de cobertura alcançado?
	Analisador de qualidade de testes baseado em mutação	A ferramenta implementa ou é compatível com ferramentas de análise de mutantes?

Figura 7. Trecho do questionário utilizado da avaliação das ferramentas de teste funcionais, adaptado de Velo 2009]

Quanto à execução da avaliação, uma equipe composta por três participantes utilizou as ferramentas e, em conjunto, atribuíram respostas ao questionário. A maioria das respostas

foi consensual e nos casos de discordância uma análise mais profunda foi realizada até que se chegasse a um consenso. Assim, o resultado obtido reflete a adequação das ferramentas aos requisitos expostos pelo SQFD.

4.2.1 Avaliação de ferramentas de apoio ao teste funcional

Com relação ao teste funcional, foram analisadas três ferramentas: Selenium², Badboy³ e Canoo⁴.

O Selenium é um conjunto de ferramentas *Open Source* que pode ser usada para criação de testes funcionais automatizados para aplicações Web. Atualmente as principais ferramentas que compõem o Selenium são: Selenium-IDE, Selenium-RC e Selenium-GRID. O Selenium-IDE é um ambiente de desenvolvimento integrado para construção de casos de testes. Ele opera como *plug-in* do FireFox, sendo uma ferramenta do tipo *record-and-playback* que provê interfaces amigáveis para o desenvolvimento e execução de suítes de testes (conjunto de testes). O Selenium RC (*Remote-Control*), por sua vez, possibilita uma maior flexibilidade ao testador, pois permite a construção de lógicas de teste mais complexas, a partir do uso de uma linguagem de programação. Por fim, o Selenium-Grid permite distribuir os testes em múltiplas máquinas, reduzindo assim o tempo gasto na execução de uma suíte de testes.

O *Badboy Web Tool* é uma ferramenta que auxilia no teste e desenvolvimento de aplicações dinâmicas complexas. Dentre as suas características destacam-se a interface de captura/reprodução e os relatórios detalhados. Por fim, tem-se o Canoo WebTest, que é uma ferramenta *Open Source* para testes automatizados de aplicações web. O diferencial desta última ferramenta é que os testes podem ser feitos com arquivos XML ou como códigos escritos na linguagem Groovy.

A Tabela 1 ilustra o resultado da avaliação realizada. Com base nela, pode-se notar que a ferramenta Selenium se adequou mais às necessidades identificadas ao conseguir 16,45%, em relação às outras duas: Badboy, com 12,18% e Canoo, com 13,88%.

É importante salientar que para maior legibilidade na apresentação dos resultados, foi suprimida da Tabela 1 os grupos de aspectos técnicos que não tiveram a necessidade do cliente atendida por nenhuma das ferramentas avaliadas.

4.2.2 Avaliação de ferramentas de apoio do teste de desempenho

Na categoria de teste de desempenho foram selecionadas para análise as seguintes ferramentas: JMeter⁵, Webload⁶ e The Grinder⁷. Essas ferramentas são amplamente utilizadas na verificação, por exemplo, do tempo de resposta, rendimento e capacidade do *software*.

O Apache JMeter é um aplicação desktop *Open Source* feita em Java para testes de desempenho. Ele pode ser usado para testar o desempenho tanto de recursos estáticos como

² <http://seleniumhq.org/>

³ <http://ww.badboy.com.au/>

⁴ <http://webtest.canoo.com/>

⁵ <http://jakarta.apache.org/jmeter/>

⁶ <http://www.webload.org/>

⁷ <http://grinder.sourceforge.net/>

dinâmicos (arquivos, servlets, banco de dados, dentre outros). Com ele é possível simular uma carga pesada em um servidor, rede ou objeto.

O Webload e o The Grinder, assim como o JMeter, são *Open Sources* e dedicam-se a auxiliar o testador na verificação dos requisitos de desempenho da aplicação através da submissão de várias requisições à aplicação sob teste.

Conforme pode ser observado na Tabela 2, a ferramenta Webload mais se adequou as necessidades identificadas ao conseguir 18,71%, em relação às outras duas JMeter, com 14,72% e The Grinder, com 15,77%.

Da mesma forma como foi feito com a apresentação dos resultados da avaliação das ferramentas de teste funcional, foi suprimida da Tabela 2 os grupos de aspectos técnicos que não tiveram a necessidade do cliente atendida por nenhuma das ferramentas avaliadas.

4.3 Discussão

Dois pontos referentes à avaliação precisam ser destacados. Em primeiro lugar, todas as ferramentas avaliadas tiveram um baixo atendimento aos aspectos técnicos levantados. Nem mesmo as ferramentas mais adequadas (Selenium e Webload) atenderam a uma porção significativa. Isso se justifica, em parte, devido ao fato de que todos os aspectos técnicos levantados foram considerados, muito embora alguns deles não fossem apropriados para a análise de uma dada categoria de ferramentas. Como por exemplo, pode-se considerar que os aspectos técnicos do grupo “Gerador de testes funcionais” não são adequados na avaliação das ferramentas de teste de desempenho uma vez que esse não é o objetivo desse tipo de ferramenta.

Contudo, alguns aspectos técnicos importantes, e devidamente apropriados para a avaliação de um determinado grupo de ferramentas, não foram atendidos por nenhuma das ferramentas analisadas. É o que acontece com os aspectos do grupo “Gerador de teste funcional”, na avaliação das ferramentas de testes funcionais, e os do grupo “Gerador de dados”, na avaliação das ferramentas de teste de desempenho.

Assim, pode-se perceber que ainda falta muito a se fazer com relação às ferramentas de apoio aos testes funcionais e de desempenho. Ressalta-se que dificilmente teríamos uma ferramenta que atendesse a todos os aspectos técnicos levantados. Porém, necessidades como aquelas relacionadas à geração de testes funcionais e de dados precisam ser atendidas.

Em segundo lugar, embora as avaliações aqui apresentadas tenham sido feitas com base em necessidades levantadas por 17 profissionais na área de testes, a definição da “melhor” ferramenta depende do contexto no qual ela vai ser aplicada. Isso porque dependendo das pessoas que irão utilizar a ferramenta, o peso dos requisitos e até os próprios requisitos podem variar. Contudo, os resultados apresentados neste trabalho podem auxiliar as empresas desenvolvedoras de *software* a realizarem suas avaliações, bem como pode direcionar os esforços de pesquisadores e desenvolvedores de ferramentas de testes para atender as necessidades identificadas.

Vale destacar que não foi possível fazer uma comparação entre o método utilizado com o proposto por Lara [Lara 2001] porque este último necessita de uma norma que estabeleça os requisitos. Assim, para se fazer uma comparação entre os dois métodos (fazendo a avaliação das ferramentas com a nossa proposta e com a de Lara) seria preciso que existisse alguma norma explicitando requisitos para ferramentas de testes e tal norma não existe.

Tabela 1. Quadro comparativo entre as ferramentas de testes funcionais Selenium, Badboy e Canoo

Grupos	Aspectos Técnicos	Pesos	Selenium		BadBoy		Canoo	
			Pontuação	Atendido	Pontuação	Atendido	Pontuação	Atendido
Gerador manual de testes	Mecanismo de captura-reprodução	0,92	1	5,87	1	3,44	1	5,285
	Gerador de teste com apoio da especificação	0,78	0		0		0	
	Uso de linguagens de alto nível	1,17	1		0		0,5	
			1		0		0,5	
	Acesso as funções do SO	2,52	1		0		1	
	Acesso ao mecanismo de persistência	2,52	0,5	1	0,5			
Avaliador de testes	Avaliador de cobertura	1,47	0,5	0,735	0	0	0,5	0,735
	Analisador de qualidade de testes baseado em mutação	0,35	0		0		0	
Gerador de Relatórios	Gerador de relatório com formato definido pelo usuário	1,58	0	0	0	0,0875	0	0,175
	Gerador de gráficos com fonte de dados e formato definido pelo usuário	1,06	0		0		0	
	Acesso a qualquer informação de teste existente no modelo de testes	1,58	0		0		0	
			0		0		1	
	Uso de hiperlinks e agrupadores nos relatórios	0,35	0	0,5	0			
Arquitetura da ferramenta	Uso de software livres	0,47	1	3	1	2,78	1	2,03
	Uso de tecnologia Web	1,12	1		1		1	
	Seguir um guia de estilo	0,53	1		1		0	
			1		1		1	
	Utilizar terminologia adequada ao contexto	0,88	1	0,5	0			
Auxílio da ferramenta	Help on-line	0,7	1	1,925	1	1,925	1	1,4
			1		1		1	
	Manual de usuário	0,7	1		1		0,5	
			1		1		1	
	Sítio de apoio com exemplos de uso	0,7	0,5	0,5	0			
	Curso de formação	0,53	0	0	0			
Executor de teste	Gerenciador de grids para execução distribuída de testes	0,76	1	4,915	0	3,95	0,5	4,25083
			1		1		1	
	Executor de teste com possibilidade de pausa e retomada da execução	1,76	1		1		0	
			1		1		1	
	Agrupador e escalonador de testes	1,25	1		1		0,5	
			0		0		1	
	Gerador de log de execução de testes	1,64	0,5		0,5		1	
			1		0,5		1	
			0		0		0	
	Simulador de interfaces de hardware e software	1,06	0,5		0,5		0,5	
			0,5		0,5		0,5	
	Integração com uma linguagem de script para configuração do ambiente de teste	0,35	0		0		0	
	Comparador de arquivos ignorando padrões configuráveis	1,23	0		0		0	
	Povoador de dados	2,11	0		0		0	
Gerenciador de transações	3,17	0	0	0				
Analísador de falhas	0,57	0	0	0				
		0	0	0				
	Cadastramento automático de falhas	0,8	0	0	0			
			Satisfação do Cliente	16,45		12,18		13,88

Tabela 2. Quadro comparativo entre as ferramentas de testes de desempenho Jmeter, Webload e The Grinder

Grupos	Aspectos Técnicos	Pesos	Jmeter		WebLoad		The Grinder		
			Questionário	Atendido	Questionário	Atendido	Questionário	Atendido	
Gerador manual de testes (5,87)	Mecanismo de captura-reprodução	0,92	1		1		1		
	Gerador de teste com apoio da especificação	0,78	0		0		0		
	Uso de linguagens de alto nível	1,17	0	5,96	1	7,13	1	7,13	
	Acesso as funções do SO	2,52	1		1		1		
	Acesso ao mecanismo de persistência	2,52	1		1		1		
Gerador de testes não-funcionais	Integração com ferramentas para manipulação de requisições e respostas em transações	0,7	0		0		0		
	Verificador de arquitetura	1,25	0		0		0		
	Gerador de testes de desempenho e estresse	1,06	1	1,06	1	1,06	1	1,06	
	Integração com sistema de varredura de portas	0,7	0		0		0		
	Gerador de testes de segurança	1,23	0		0		0		
Gerador de Relatórios	Gerador de relatório com formato definido pelo usuário	1,58	0		1		0		
	Gerador de gráficos com fonte de dados e formato definido pelo usuário	1,06	0	0	1	2,64	0	0	
	Acesso a qualquer informação de teste existente no modelo de testes	1,58	0		0		0		
	Uso de hiperlinks e agrupadores nos relatórios	0,35	0		0		0		
Arquitetura da ferramenta	Uso de software livres	0,47	0		0		1		
	Uso de tecnologia Web	1,12	1	2,31	1	2,31	1	2,78	
	Seguir um guia de estilo	0,53	1		1		1		
	Utilizar terminologia adequada ao contexto	0,88	1		1		1		
Auxílio da ferramenta	Help on-line	0,7	0,5		0,5		0,5		
	Manual de usuário	0,7	1	1,575	1	1,93	1	1,575	
	Sítio de apoio com exemplos de uso	0,7	1		0,5		1		
	Curso de formação	0,53	0		1		0		
Executor de teste	Gerenciador de grids para execução distribuída de testes	0,76	1		0		1		
	Executor de teste com possibilidade de pausa e retomada da execução		1		1		0		
			1,76	0		1		0	
	Agrupador e escalonador de testes		1		1		1		
			1,25	0		0		0	
	Gerador de log de execução de testes		0,5		0,5		0		
			1,64	1	3,813333	1	3,64	0,5	2,611667
				0,5		0,5		0	
	Simulador de interfaces de hardware e software	1,06	1		1		1		
	Integração com uma linguagem de script para configuração do ambiente de teste	0,35	1		1		1		
	Comparador de arquivos ignorando padrões configuráveis	1,23	0		0		0		
	Povoador de dados	2,11	0		0		0		
	Gerenciador de transações	3,17	0		0		0		
Analizador de falhas	0,57	0		0		0			
Cadastramento automático de falhas	0,8	0		0		0			
Gerador de plano de teste	Gestão dos dados do plano de teste	1	0		0		0		
	Estimador de complexidade de caso de uso	1,41	0		0		0		
	Estimador de prazo para execução de tarefa de teste	1	0		0		0		
	Base de dados histórica de projetos		2	0	0	0	0	0	0,615
				0		0		0	
	Calculador de produtividade	1,47	0		0		0		
	Registro de tarefas	1,23	0		0		1		
	Alocador inteligente de tarefa	0,65	0		0		0		
Integração com ferramentas de gerenciamento de projetos	0,82	0		0		0			
			Satisfação do Cliente	14,718		18,71		15,772	

5. Conclusões

Este artigo apresenta uma avaliação comparativa entre as principais ferramentas de testes funcionais e de desempenho, utilizando 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.

Também foi apresentado neste trabalho um aperfeiçoamento do método proposto em [Velooso 2009]. O aperfeiçoamento no método apresentado possibilita a comparação entre diversas ferramentas existentes no mercado, corrigindo possíveis distorções que eram possíveis na primeira versão proposta do método de avaliação. O método proposto pode 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.

Na categoria de ferramentas de testes funcionais, foram avaliadas o Selenium, Badboy e Canoo. Com relação a ferramentas de testes de desempenho foram consideradas: Jmeter, Weblload e The Grinder. A primeira 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, conforme descrito no SQFD. Da mesma forma, a ferramenta Weblload é a mais adequada dentro da categoria desempenho ao cobrir mais necessidades esperadas do que as outras ferramentas.

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.

Agradecimentos

Este trabalho recebeu apoio financeiro do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) através de bolsas de iniciação científicas do programa PIBIC.

Referências

- Akao, Y. (1996) “*Introdução ao Desdobramento da Qualidade*”. Tradução de Zelinda Tomie Fujikawa Belo Horizonte: Fundação Chistiano Ottoni, 187p.
- Alves, N. R.; Padua, C. I. P. S. (2001) “Especificação de Requisitos de Usabilidade utilizando-se o Método Desdobramento da Função Qualidade”. In: *Anais da I Jornada Latino Americana em Engenharia de Software e Engenharia do Conhecimento*, Buenos Aires.
- 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: *Anais do IV Simpósio Brasileiro de Sistemas de Informação*, Rio de Janeiro, 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: A Roadmap. In The Future of Software Engineering*”, pages 61–72. ACM Press.
- IEEE (2004) “*Guide to the Software Engineering Body of Knowledge*”. IEEE Computer Society.
- Ioannou, G.; Pramataris, K. e Prastacos, G. (2004) “Quality function deployment approach to web site development: Applications for electronic retailing”. *Les Cahiers du Management Technologique*, v. 13, n. 3.
- Karlsson, J. (1997) “Managing software requirements using quality function deployment”. *Software Quality Journal*, v. 6, pg 311-326.
- 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 12119. (1998) “*Tecnologia de Informação – Pacotes de software – Teste e requisitos de qualidade*”. ISO/IEC - International Organization for Standardization.
- Veloso, J. S., Santos, I. S., Santos Neto, P. (2009) “Aplicação do SQFD na escolha de uma ferramenta de apoio ao teste de *software*”. In: *III ERCEMAPI – Encontro Regional de Computação Ceará – Piauí – Maranhão*, Parnaíba-PI, outubro.
- Weber, K. (2001). “*Qualidade de Software: Teoria e Prática*”. Prentice Hall, São Paulo.
- Santos, I. S.; Santos Neto, P. A., Resende, R. S. F.; Pádua, C. I. P. S. (2010) “Requisitos e Aspectos Técnicos desejados em ferramentas de testes de software: um estudo a partir do uso do SQFD”. Aceito para publicação na revista RESI em novembro de 2010.