

# AVALIAÇÃO DE UM PRODUTO DE *SOFTWARE* LIVRE: PRODUTO DE AUTOMAÇÃO DE TESTES

Angela Maria Alves  
*Centro de Tecnologia da Informação Renato Archer  
Campinas – São Paulo- Brasil*

Claudio André da Silva Junior  
*Universidade Federal de Lavras  
São Paulo - Brasil*

## RESUMO

O crescente sucesso e a conseqüente adoção do *software* livre têm sido impactados negativamente pela ausência de um método reconhecido e aceito pela comunidade para a avaliação e seleção de um pacote de *software*. Nesse contexto, este trabalho apresenta a utilização do modelo de avaliação de *software* livre denominado OpenBRR, e realiza um estudo de caso que avalia o *software* de automação de testes Selenium, permitindo, assim, atestar-se a aplicabilidade do modelo na execução das tarefas de avaliação e seleção de um produto de *software*.

## PALAVRAS-CHAVE

*Software* Livre; Modelo de Avaliação de *Software*; OpenBRR; Testes de *Software*; Selenium.

## 1. INTRODUÇÃO

O movimento de *software* livre (SL) é um fenômeno que está provocando uma grande alteração no modo como as pessoas e empresas produzem e consomem *software*. Parte dessa mudança se deve ao aspecto político-social como defende Stallman (2007), parte se deve às inovações econômicas como observam Lerner e Tirole (2005), ou ainda, aos possíveis ganhos em qualidade como sugeriu Raymond (2001, p. 41). Sejam quais forem as razões, o SL está cada vez mais presente no cotidiano da sociedade: o Linux avança nos celulares, o *software* BIND é peça fundamental na internet (*Internet Systems Consortium* [ISC], 2008).

É natural, portanto, que mais pessoas e empresas se interessem por utilizar SL, inclusive em tarefas consideradas essenciais e críticas. Contudo, essa empreitada oferece alguns desafios:

- Como selecionar um SL dentre os vários produtos livres disponíveis;
- Como verificar se determinado SL está adequado para uso;
- Como aferir de forma objetiva o grau de maturidade de um SL.

Para sugerir uma abordagem para essas preocupações, o presente trabalho mostra a aplicação do método de avaliação de *software* denominado *Open Business Readiness Rating* [OpenBRR] (OpenBRR, 2005) na avaliação do *software* de automação de testes Selenium.

Por fim, apesar de existirem algumas diferenças entre *software* livre (*free software*) e código aberto (*open source*) (Stallman, 2007), neste trabalho o termo *software* livre será sempre usado.

## 2. A AVALIAÇÃO

Neste trabalho a definição de *software* inclui “também todos os dados de documentação e configuração associados, necessários para que o programa opere corretamente” (Sommerville, 2007, p. 4).

## 2.1 Avaliação do Selenium utilizando o Modelo OpenBRR

### 2.1.1 Testes de *software*

Podemos definir testes de *software* como o conjunto de atividades que tem como objetivo encontrar erros em um produto de *software*, permitindo que esses erros sejam corrigidos antes que o produto seja entregue para o usuário final. Basicamente, essa tarefa é realizada através da aplicação de um conjunto de casos de teste durante a execução do programa. Desta forma, como lembra o SWEBOK (*Institute of Electrical and Electronics Engineer* [IEEE], 2004) teste de *software* é uma atividade realizada para avaliar a qualidade de um produto e para melhorá-lo mediante a identificação de defeitos e problemas.

Como o objetivo final dos testes é buscar a qualidade do *software*, optou-se por utilizar abordagem semelhante à de Chua e Dyson (2004), e propor a norma ISO 9126 (*International Organization for Standardization* [ISO], 2001) como um guia mestre para auxiliar a visualização de alguns dos pontos nevrálgicos relacionados com as atividades de teste de *software*. Com este intento, a tabela 1 apresenta as características do produto definidas pela ISO 9126, seu significado e suas respectivas subcategorias.

Tabela 1. Características do produto conforme a ISO 9126.

Categoria	Descrição
Funcionalidade	As funções requeridas estão disponíveis no <i>software</i> . Subcategorias: Adequação, Precisão, Interoperabilidade e Segurança
Confiabilidade	O quão confiável é o <i>software</i> . Subcategorias: Maturidade, Tolerância a Falhas e Recuperabilidade
Usabilidade	O <i>software</i> é de fácil manuseio. Subcategorias: Inteligibilidade, Compreensibilidade, Operacionalidade e Atratividade.
Eficiência	O quão eficiente é o <i>software</i> . Subcategorias: Comportamento Temporal e Utilização de recurso.
Manutenibilidade	O quão fácil é a manutenção do <i>software</i> . Subcategorias: Analisabilidade, Modificabilidade, Estabilidade e Testabilidade.
Portabilidade	É difícil transferir o sistema para outros ambientes. Subcategorias: Adaptabilidade, Instalabilidade, Conformidade e Substituibilidade.

### 2.1.2 O modelo de Avaliação OpenBRR

O OpenBRR é um modelo de avaliação de maturidade de *software* padronizado e aberto, desenvolvido pelo Centro para Investigação de *Software* Livre da Universidade Carnegie Mellon West, O'Reilly CodeZoo, SpikeSource e Intel. Seus objetivos são auxiliar os gestores de TI a avaliarem pacotes de *software* livre, a verificarem qual pacote é mais adequado a uma determinada necessidade e, também, permitir um entendimento comum dos resultados destas avaliações. Para garantir sua robustez, confiabilidade e aplicabilidade, esse modelo foi estruturado sobre alguns requisitos, conforme tabela 2.

Tabela 2. Requisitos do modelo OpenBRR.

Requisito	Descrição
Deve ser completo	O modelo deve evidenciar as características importantes do produto analisado, sejam elas favoráveis ou não
Deve ser simples	O modelo precisa ser de fácil compreensão e aplicação.
Deve ser consistente	Os índices e escalas devem ser consistentes e comparáveis, inclusive entre pacotes de <i>software</i> diferentes.
Deve ser adaptável	O modelo precisa ser extensível, e se adequar a potenciais necessidades e usos futuros

A utilização deste modelo é vantajosa (OpenBRR, 2005), pois fornece um mecanismo objetivo e transparente de avaliação que permite a reprodução de resultados, tornando possível uma uniformização no seu entendimento, oferecendo, ainda, uma visão mais adequada aos usuários que ficarão melhor aparelhados na escolha de um *software*. Some-se a isso uma melhor visualização pelos desenvolvedores dos pontos fracos do *software*, permitindo uma priorização mais adequada às necessidades de seus usuários.

O OpenBRR define doze categorias para levantamento de *software*, que possuem grande semelhança com as características definidas na ISO 9126, conforme se vê na tabela 3. Este fato mostra que o modelo aproveita-se das boas práticas disponíveis na engenharia de *software*.

Tabela 3. Categorias de levantamento definidas pelo OpenBRR.

Categoria	Descrição
Funcionalidade	O quanto o <i>software</i> é adequado aos requisitos do usuário médio?
Usabilidade	A interface com o usuário é boa? Qual a facilidade de uso do <i>software</i> por usuários finais? É fácil instalar, configurar e manter o <i>software</i> ?
Qualidade	Qual é a qualidade do projeto, da codificação e dos testes? O quanto é completo e livre erros?
Segurança	Quão bem o <i>software</i> lida com questões de segurança? O quanto seguro é?
Performance	Quão bom é o desempenho do <i>software</i> ?
Escalabilidade	Quão bem o <i>software</i> pode escalar para um ambiente maior?
Arquitetura	Quão bem o <i>software</i> foi projetado? Quão modular, portátil, flexível, extensível, aberto e fácil de integrar ele é?
Suporte	Qual o nível de suporte oferecido ao <i>software</i> ?
Documentação	Qual é a qualidade da documentação deste <i>software</i> ?
Adoção	Qual o nível de adoção do pacote pela comunidade, mercado, indústria?
Comunidade	O quanto a comunidade deste <i>software</i> é ativa?
Profissionalismo	Qual o nível de profissionalismo do processo de desenvolvimento e da organização do projeto?

### 2.1.3 O Processo de Avaliação

O Selenium (<http://selenium.openqa.org/>), atualmente na versão 1.0 beta 2, é uma ferramenta de testes para aplicações web que executa diretamente no navegador, e funciona no Internet Explorer e Firefox no Windows, Firefox no Linux e Firefox e Safari no Macintosh. Como os testes do Selenium são executados diretamente no navegador, exatamente como um usuário os executaria, estes serão normalmente testes de sistema.

Por se tratar este trabalho de uma pesquisa orientada à avaliação, a metodologia utilizada foi o estudo de caso único (Yin, 2005), que neste caso "representa uma maneira de se investigar um tópico empírico seguindo-se um conjunto de procedimentos pré-especificados". (Yin, 2001, p. 35).

Embora o OpenBRR defina doze categorias de avaliação, para evitar que as categorias menos relevantes diluam a importância das de maior relevância recomenda-se que se compile uma lista de todas as categorias em ordem decrescente de importância; lista esta definida de acordo com o *software* em análise e com as necessidades que este deve atender. Feito isso, selecionam-se entre cinco e sete das categorias mais relevantes, e processa-se a análise utilizando apenas este subconjunto.

Neste trabalho definiu-se a importância de cada categoria e seus respectivos pesos através de uma pesquisa feita junto a cinco desenvolvedores, buscando identificar de forma rápida quais características seriam as mais valorizadas por este conjunto de possíveis usuários da ferramenta, e de que forma eles estabeleciam uma gradação entre as diferentes categorias. O resultado obtido pode ser visto na tabela 4.

Tabela 4. Categorias do OpenBRR selecionadas para avaliação do Selenium com seus respectivos pesos.

Posição	Categoria de Levantamento	Peso
1	Funcionalidade	30%
2	Usabilidade	20%
3	Documentação	15%
4	Adoção	15%
5	Qualidade	10%
6	Performance	10%

As planilhas eletrônicas, oferecidas pelo modelo OpenBRR, definem uma série de métricas para avaliação de *software* para cada uma das categorias de levantamento. Cada métrica também requer um peso relativo dentro da categoria.

Assim, para cada uma das categorias de levantamento selecionadas para avaliação, os pesos foram definidos. Realizou-se um levantamento para se obter os dados necessários para responder a cada uma dessas perguntas ou métricas. Os dados obtidos foram então usados para preencher as planilhas, os cálculos foram realizados, e os resultados normalizados de acordo com os pesos atribuídos.

Tabela 5. Índice de avaliação do Selenium pelo OpenBRR

Nota final do Selenium
<b>3,73</b>

Preenchidas todas as métricas em todas as categorias selecionadas para avaliação, obteve-se a nota final, ou índice de avaliação de preparo para os negócios. O resultado da avaliação do Selenium pode ser visto na tabela 5 acima.

### 3. CONCLUSÃO

Aplicamos o método OpenBRR na avaliação do *software* Selenium e pudemos aferir sua aplicabilidade e méritos, sendo, portanto, recomendado como ferramenta de análise e de documentação do processo de escolha de um novo *software* livre dentro das organizações.

A nota do Selenium foi 3,73, o que se considera um *software* “bom” de acordo com os critérios estabelecidos pelo modelo OpenBRR. Assim, baseada em critérios objetivos dispõe-se de uma análise reproduzível que estabelece que o Selenium é um produto maduro o suficiente para ser adotado pelos departamentos de TI. Como consequência, a divulgação do *software* e análises posteriores podem ser feitas utilizando-se métricas documentadas e passíveis de auditoria.

Ademais, verificamos que a falta de análises conhecidas de benchmark do Selenium foi um dos pontos mais prejudiciais na avaliação e, portanto, a realização destas representará um importante passo rumo à solidificação da qualidade do produto. Ainda, que a existência de alguns bugs críticos não corrigidos em tempo considerado adequado é um dos pontos que precisa ser atacado pelos desenvolvedores.

Por fim, a aplicação do OpenBRR se mostrou útil ao oferecer um mecanismo de avaliação objetivo, reproduzível e auditável, que além de poupar os avaliadores de desenvolverem seus próprios mecanismos de avaliação, produz um valor numérico que permite a comparação dos resultados com outras avaliações dos mais diferentes produtos de *software*.

Porém, o modelo OpenBRR não é conhecido, tampouco adotado largamente. Por essa razão, trabalhos futuros devem seguir incentivando outros pesquisadores a utilizá-lo, confirmando, assim, seus méritos e potencialidades

### AGRADECIMENTO

Agradecemos à Fundação de Apoio ao Ensino, Pesquisa e Extensão da Universidade Federal de Lavras e ao prof. Dr. Ahmed Esmín pelo suporte oferecido para a elaboração deste trabalho.

### REFERÊNCIAS

- Chua, B. B. and Dyson, L. E., 2004. Applying the ISO 9126 Model to the Evaluation of an e-Learning system. *Beyond the Comfort Zone: Proceedings of the 21st*. Perth, Australia, pp. 184-190.
- Institute of Electrical and Electronics Engineers Computer Society, 2004. *Guide to Software Engineering Body of Knowledge (SWEBOK)*. Califórnia, EUA.
- International Organization for Standardization, 2001. *ISO/IEC 9126-1:2001*. Genebra, Suíça.
- Internet Systems Consortium, 2008. *Internet Systems Consortium: ISC BIND*. Disponível em: <<http://www.isc.org/index.pl?sw/bind/index.php>>. Acesso em: julho de 2008.
- Lerner, J. and Tirole, J., 2005. Economic Perspectives on Open Source. *Perspectives on Free and Open Source Software*. The MIT Press, Londres, Inglaterra, pp. 47-78.
- OpenBRR, 2005. *Business Readiness Rating for Open Source: A Proposed Open Standard to Facilitate Assessment and Adoption of Open Source Software*. Disponível em: <<http://www.openbrr.org>>. Acesso em: julho de 2008.
- Raymond, E. S., 2001. *The cathedral and the bazaar: Musings on Linux and open source by an accidental revolutionary*. O'Reilly, Sebastopol, EUA.
- Sommerville, I., 2007. *Engenharia de Software*. Pearson Addison-Wesley, São Paulo, Brasil.
- Stallman, R. M., 2007. *Why “Open Source” misses the point of Free Software*. Disponível em: <<http://www.gnu.org/philosophy/open-source-misses-the-point.html>>. Acesso em: julho de 2008.
- Yin, R. K., 2001. *Estudo de Caso: Planejamento e Métodos*. Bookman, Porto Alegre, Brasil.