

Gerenciamento de Teste de Software: Um comparativo entre Ferramentas Open Source

Software Test Management: A Comparison of Open Source Tools

Deivison Gomes de Amorim¹, Ivaldir de Farias Júnior², Nelson Leitão Júnior³, Marcelo Mendonça Teixeira⁴

¹Unibratex, Recife, Brasil

²Estácio, Recife, Brasil

³CESAR EDU, Recife, Brasil

Correspondência: Deivison Gomes de Amorim, Endereço: Rua Vital Barbosa de Brito, 53 – São Sebastião CEP.: 55.750-000 Surubim, PE, Brasil. Tel.: 55 81 99730-1715. E-mail: amorim.deivison@outlook.com

Recebido: 08 de outubro de 2016 Aceito: 20 de dezembro de 2016 Publicado: 01 de maio de 2017

DOI: 10.21714/1679-18272016v14Esp2.p296-302

Resumo

Tendo em vista a atenção dada à qualidade dentro de um projeto de desenvolvimento de sistemas de informação, uma das maiores preocupações atuais é a busca por novas técnicas que auxiliem o processo de execução e planejamento de teste. A utilização de ferramentas de gestão de teste tem agregado confiança e qualidade ao desenvolvimento, mas ainda é um problema encontrar a ferramenta certa para cada projeto. Diante dessa dificuldade o objetivo deste artigo é analisar ferramentas *open source* para gestão de teste de software, visando facilitar a escolha das mesmas para utilização em projetos de desenvolvimento de sistemas de informação. Visando alcançar este objetivo, executamos uma revisão da literatura, realizamos um comparativo entre as ferramentas analisadas e fizemos uma análise de cada ferramenta comparada a características de qualidade oriundas da NBR ISO/IEC 9126. Como resultado, identificamos qual das ferramentas estudadas dá suporte ao maior número de situações vividas dentro do desenvolvimento de sistemas de informação. Além disso, descobrimos qual ferramenta atende de forma mais satisfatória aos pontos analisados diante da NBR ISO/IEC 9126. Assim esperamos facilitar e dar uma melhor compreensão na escolha de ferramentas de gestão de teste utilizadas em projetos de desenvolvimento de sistemas de informação.

Palavras-chave: Teste de Software, Sistemas de Informação, Gerenciamento de Teste de Software.

Abstract

Considering the focus on software quality within an information systems development project, one of the current concerns is the search for new techniques to assist the implementation and test planning process. The use of test management tools has added reliability and quality to the development, but it is still a problem finding the right tool for each project. Given this difficulty, the purpose of this article is to analyze open source test management tools, to facilitate the choice thereof for use in information systems development projects. In order to achieve this goal, we performed a literature review, we made a comparison between the tools analyzed and made an analysis of each tool compared to quality characteristics derived from the ISO / IEC 9126. As a result, we identified which of the analyzed tools supports the largest number of situations experienced in the information systems development. Furthermore, we identified which tool serves more satisfactorily the points analyzed on the ISO / IEC 9126. Therefore, we hope to facilitate and give a better understanding on the choice of test management tools used in information systems development projects.

Keywords: Software Testing, Information Systems, Software Test Management.

Esta obra está licenciada sob uma Licença Creative Commons Attribution 3.0.

1. Introdução

Um projeto de desenvolvimento de sistemas de informação possui várias etapas, onde a utilização de métodos e ferramentas podem auxiliar o seu bom andamento. Porém, mesmo com a utilização desses recursos, podem existir erros no produto gerado. Esses erros podem levar o projeto a ter muitas dificuldades, elevar os custos ou até mesmo levar ao seu insucesso (Pressman, 2000). Visando minimizar a ocorrência desses erros são executadas atividades que buscam uma maior qualidade no produto gerado. Essas atividades são conhecidas como VV&T

(Verificação, Validação e Teste).

Segundo Maldonado (1991), dentre as tarefas de VV&T, o teste se torna uma das atividades mais executadas. Essa atividade é uma ferramenta poderosa na localização e eliminação de erros no processo de desenvolvimento. Basicamente, a execução dos testes se divide em quatro etapas: planejamentos de teste, projeto de casos de teste, execução dos testes e avaliação dos resultados encontrados (Beizer, 1990; Myers, 1979; Pressman, 2000). Para que se possa ter uma melhor avaliação de qualidade do produto de software gerado é importante que o processo de teste seja executado durante toda a duração do projeto, contudo, mesmo com a realização deste processo, não existe a garantia de qualidade no produto final.

Para ser utilizado como métrica de qualidade podemos nos basear em normas e diretrizes que dão subsídios ao desenvolvimento visando uma maior qualidade. Temos como referencial a NBR ISO/IEC 9126-1 que nos dá parâmetros para atingir uma maior qualidade do produto de software (ABNT, 2003).

Harrold (2000) afirma que o processo de teste de software é uma das atividades com custos mais altos dentro do projeto, podendo chegar a comprometer até 50% dos custos. Na busca por soluções para melhorar a execução da atividade de teste de software e amenizar os custos do processo, temos como solução paliativa a utilização de ferramentas que auxiliam nos testes de software. Essas ferramentas trazem um aparato necessário para o melhor funcionamento do processo de teste e muitas delas são encontradas no cenário *open source*. A terminologia "*open source*" se refere aos softwares, que por livre vontade do seu desenvolvedor, tem seu código fonte acessível ao público e podem ser alterados e lidos livremente (Torres, 2015). No caso da pesquisa em voga, as ferramentas que estudamos são as de gerenciamento de teste de software.

A utilização de ferramentas de gerenciamento de teste é uma forte aliada do processo de desenvolvimento de sistemas de informação. No entanto, existe uma dificuldade em encontrar a ferramenta que corresponda de forma mais significativa e abrangente às necessidades gerais do projeto, tendo em vista que cada projeto tem suas particularidades.

Em virtude do exposto, definiu-se, como questão principal da pesquisa:

“Quais as principais ferramentas *open source* que auxiliam mais efetivamente o desenvolvimento de sistemas de informação no processo de gerenciamento de teste de software a luz de características oriundas da NBR ISO/IEC 9126”

Este trabalho foi dividido em quatro seções, organizados da seguinte forma: Na seção 2 tratamos da revisão bibliográfica dos principais assuntos da pesquisa: teste de software e ferramentas de gerenciamento teste de software. Na seção 3 são descritas as ferramentas de gerenciamento de teste e suas características principais. Também são apresentados os quadros comparativos entre as ferramentas, um deles exibindo o comparativo entre as ferramentas e outro expondo o comparativo diante de características da NBR ISO/IEC 9126. Por fim, na seção 4 é exposta a conclusão diante das análises e pesquisas desenvolvidas.

2. Fundamentação Teórica

2.1. Teste de Software

O processo de teste utiliza várias formas, critérios, ferramentas e técnicas visando através desse grupo de possibilidades evitar que erros permaneçam no software desenvolvido. Dessa forma, a atividade de teste se torna uma etapa fundamental para busca da qualidade durante todo período do desenvolvimento do projeto de software (Maldonado, 1991; Pressman, 2000).

Myers (1979) ressalta que a atividade de teste tem como principal finalidade descobrir erros e defeitos através do planejamento e execução de casos de teste. Segundo ele, um bom caso de teste é aquele que nos proporciona uma maior chance de encontrar defeitos, já um caso de teste bem-sucedido é aquele que nos revela erros ou defeitos durante o desenvolvimento do projeto. Vincenzi (1998) afirma que técnicas e critérios são elaborados para disponibilizar mais possibilidades de execução dos testes de forma mais correta e assim ter melhores resultados. Essas técnicas precisam respeitar alguns fatores de restrição, tais como tempo e custo. Weiss (2013), afirma que um problema enfrentado pelo processo de teste é que ele é muito suscetível a erros e se for executado de forma manual se torna inviável. Para auxiliar a atividade são utilizadas ferramentas, que auxiliam tanto na gestão do projeto de teste como na sua execução e automatização.

Infelizmente, não é possível, através da atividade de teste ter a garantia que um produto gerado esteja livre de erros ou defeitos. Porém, a execução dos testes de forma organizada, sistemática e criteriosa agrega confiança e aumenta o nível de qualidade do sistema produzido (a atividade atua como um termômetro de qualidade). O processo de teste ajuda a identificar erros, mas se após essa identificação não forem aplicadas as devidas correções o produto gerado deixará de elevar sua qualidade (Vincenzi, 2000; Beizer, 1995). Para que esta atividade seja executada de maneira mais rápida e efetiva, se faz necessário a adoção de ferramentas de teste para organizar, agilizar e aumentar a visibilidade dos testes de software.

2.2 Ferramentas de Teste de Software

Segundo Prikladnicki (2006), a engenharia de software busca cada vez mais melhorar seus processos e suas etapas. Este pensamento não é diferente na atividade de teste software, onde o que se procura é mais rapidez, eficácia e confiança na sua execução. Assim, fatores negativos como altos custos e o longo tempo necessário para o cumprimento das etapas de teste são amenizados.

Para Sommerville (2007), a utilização de ferramentas voltadas para teste de software se consegue atender parcialmente essa busca por melhorias no processo. Quando não se utiliza ferramentas durante as fases de teste, tornamos essa etapa mais suscetível a erros e muitas vezes limitamos o seu potencial, deixando assim de encontrar um maior número de defeitos. Vincenzi (2000) expõe que a utilização de ferramentas de teste produz a transferência da tecnologia para a indústria, contribuindo assim para a evolução desses cenários, fatores que são indispensáveis para o desenvolvimento de software com maior qualidade.

Além dessa gama de possibilidades oriundas da inserção de ferramentas no processo de teste, Maldonado et al. (1998) afirma que a utilização de ferramentas no contexto de teste de software reduz os custos do processo. Fato esse que é considerado significativo diante dos altos investimentos em que esta área requer durante todo o processo de desenvolvimento de sistemas de informação, nesta ótica, uma forma de redução de custos é a utilização de ferramentas *open source* que auxiliem o processo de teste. Torres (2015) expõe que o termo “*open source*” referencia os softwares que tem seu código livre para modificações e leitura do público em geral, que vão desde programadores profissionais que buscam melhorar o software até estudantes que visam aprofundar seus conhecimentos.

2.3 NBR ISO/IEC 9126-1

A NBR ISO/IEC 9126 manteve as mesmas abordagens da NBR 13596 para qualidade de software. A norma traz o conjunto de características de qualidade do produto de software e determina especificações de requisitos funcionais e não funcionais tanto para o usuário como para o cliente. A NBR ISO/IEC 9126 recebeu alguns adendos se comparada com a NBR 13596, dentre eles destacam-se: as subcaracterísticas de qualidade; especificação de um modelo de qualidade; introdução de qualidade em uso; remoção do processo de avaliação; coordenação de seu conteúdo com a NBR ISO/IEC 14598-1 (ABTN 2003).

A partir do diálogo teórico entre os autores podemos salientar que a utilização de ferramentas, em conjunto ou não de normas e diretrizes de qualidade de produto de software, pode proporcionar uma melhor visualização e entendimento de todo o processo de teste. Assim podemos, a partir delas, ter bons parâmetros de qualidade, obter relatórios dos resultados dos testes executados, criar e executar planos de teste e ainda dispor de toda a bateria de testes executados, dando a possibilidade de re-execução dos testes falhos e exibir todos os testes não falhos.

3. Descritivo das ferramentas open source para gerenciamento de teste

3.1. TestLink

A ferramenta TestLink foi criada em 2005 e é mantida pela TestLink *Community*, sendo destinada a dar suporte a projetos de desenvolvimento de software nas atividades de gerenciamento de teste de software. A ferramenta foi desenvolvida em linguagem web PHP e possui código aberto, fato que a classifica na categoria *open source*. O TestLink tem sua interface baseada na *web*, favorecendo sua utilização em equipes distribuídas geograficamente, e atua sob licença GPL (*General Public License*) (CAETANO, 2008).

Segundo Bittencourt (2013) e Reis (2012), com a ferramenta é possível realizar funções de gerenciamento de projetos, criação e execução de casos e planos de teste, permite executar uma bateria de teste e definir se um determinado teste foi executado corretamente ou se houve falha. Além disso, a mesma ainda permite gerenciar os privilégios de acesso dos usuários e possibilita criar relatórios com os resultados das execuções dos casos de teste. A ferramenta também conta com uma função de escrita e gerenciamento de requisitos, onde é possível anexar os requisitos aos casos de teste a serem executados facilitando a rastreabilidade entre ambos.

Se tratando de integração com ferramentas externas de gerenciamento de defeitos (*bug tracking*), o TestLink está entre uma das melhores do segmento, conseguindo desenvolver um ótimo trabalho de forma integrada com ferramentas como Mantis ou Bugzilla, por exemplo (TESTLINK, 2015). A mesma está atualmente na sua versão 1.9,14 (Padawan) e possui suporte nativo aos bancos de dados PostgreSQL e MySQL.

3.2 Rth

Construída em linguagem PHP, a ferramenta de gerenciamento de teste Rth é mais uma ferramenta classificada na categoria *open source*, sendo baseada na *web* e atuando sob licença GNU (*General Public License – Versão 2.0*). A ferramenta oferece uma grande variedade de recursos de gerenciamento e permite a criação de casos de teste, execução e monitoramento do progresso dos testes e ainda possibilita a movimentação dessas informações através de arquivos MS Excel.

A Rth permite a geração de relatórios para análise dos testes e possui a funcionalidade de especificação de requisitos, onde cada requisito pode ser adicionado a um caso de teste e assim agregar maior amplitude de informações ao testador. Os requisitos especificados na ferramenta são trabalhados em forma de árvore, assim

permitindo uma melhor visualização das informações e facilitando o processo de entendimento da rastreabilidade dos requisitos.

Com a ferramenta Rth não se consegue fazer integração com ferramentas externas, tais como ferramentas de rastreamento de defeitos. Esse é um dos pontos fracos desta ferramenta, pois esse é um fator importante se tratando de gestão de teste. O Rth possui uma funcionalidade que simula uma ferramenta de rastreamento de defeitos, mas essa funcionalidade é bem limitada e não atende de forma satisfatória a necessidade (RTH, 2015).

3.3 TestMaster

A ferramenta TestMaster é destinada a gestão de testes de software e permite a criação e execução de casos de teste, bem como tem funções de automação de teste. Além disso, a mesma permite a geração de relatórios das informações dos casos de testes executados. A ferramenta, que é escrita em PERL e é baseada na web, atuando sob licença pública e utiliza o Apache como servidor web.

O TestMaster permite a importação e exportação de informações para documentos Microsoft Word e arquivos do tipo CSV (Comma Separated Values), bem como permite a especificação de requisitos funcionais e mapeia os casos de teste para os requisitos (gerando uma rastreabilidade entre ambos). Na execução dos casos de teste é possível sinalizá-los como bem sucedidos ou não e permite o registro de algumas notas do testador referente ao que foi testado. Outra funcionalidade importante da ferramenta é a apresentação das estatísticas dos casos de teste executados, dando assim uma maior e melhor cobertura ao gerenciamento do projeto em percurso (TESTMASTER, 2015).

3.3 TestiTool

TestiTool é uma ferramenta destinada para gerenciamento de testes de software e controle de qualidade, foi desenvolvida em PHP e tem seu funcionamento voltado para web sob licença GNU (General Public License – Versão 2). A ferramenta permite a criação de planos de teste, onde é possível adicionar casos de teste ao plano criado. O TestiTool também possibilita o gerenciamento de requisitos funcionais e mapeá-los para os casos de teste ocasionando a rastreabilidade entre ambos. Tal característica é muito importante para uma ferramenta de gestão de teste de software. Se tratando da execução dos casos de teste é possível marcar os testes executados como bem sucedidos ou mal sucedidos, além de possibilitar a criação de vários tipos de relatórios, tais como: relatórios de bugs, relatórios de progresso, relatório de taxa de falha, entre outros. A ferramenta ainda permite o manuseio dos planos de teste criados, possibilitando que eles sejam importados ou exportados em formato CSV (Testitool, 2015).

4. Estudo de Caso

4.1. Comparativo entre as ferramentas

O quadro 1 apresenta o comparativo entre as ferramentas estudadas com as características gerais do procedimento de desenvolvimento de software. Em seguida é apresentada a análise das ferramentas, com base nas suas respectivas particularidades.

Ao final da análise observamos que as ferramentas TestLink e Rth atendem a um maior quantitativo de características analisadas, assim se destacando das demais. Porém, a ferramenta TestLink se sobressai diante da Rth pelo fato de ter uma interface mais amigável e de fácil utilização. Outro ponto forte da ferramenta TestLink é a sua integração com ferramentas externas de bug track já citada anteriormente, fato que alavanca seu desempenho quando temos em vista o cenário de gestão de teste de software. Portanto, concluímos que ferramenta TestLink é a melhor dentre as analisadas.

FERRAMENTAS	TestLink	Rth	TestMaster	TestiTool
GERAÇÃO DE RELATÓRIOS	✓	✓	✓	✓
PLATAFORMA WEB	✓	✓	✓	✓
TIPO DE CONTROLE DE ACESSO DE USUÁRIO	✓	✓	✓	✓
INTEGRAÇÃO COM FERRAMENTAS BugTrack	✓	✓	✗	✗
HISTÓRICO DE TESTES EXECUTADOS	✓	✓	✓	✓
ESPECIFICAÇÃO DE REQUISITOS	✓	✓	✗	✓
IMPORTAR / EXPORTAR CONTEÚDOS	✓	✓	✓	✓
SUORTE A BANCO DE DADOS	✓	✓	✓	✓
FUNÇÕES PARA AUTOMAÇÃO DE TESTE	✗	✓	✓	✗

LEGENDA:

✓ - A ferramenta possui tal característica

✗ - A ferramenta não possui tal característica

✓ - A ferramenta contempla tal característica com outras particularidade

Quadro 1: Comparativo entre as ferramentas de gerenciamento de teste de software.

Fonte: Elaborado pelos autores (2016).

4.2. Comparativo entre as ferramentas

O quadro 2 apresenta o comparativo entre as ferramentas analisadas e as características de usabilidade, funcionalidade e qualidade em uso a luz da NBR ISO/EIC 9126 (Ver Figura 2). Ao realizarmos a análise, observamos dois pontos dentre as especificações da NBR ISO 9126. Um se trata da qualidade interna e externa e o outro expõe a qualidade em uso. No quesito funcionalidade, analisamos os pontos de adequação, acurácia, interoperabilidade e segurança de acesso. Já se tratando do quesito usabilidade foram analisadas a inteligibilidade, apreensibilidade, operacionalidade e a atratividade. No que se refere à questão de qualidade em uso as ferramentas, foram analisadas segundo quatro características: eficácia, produtividade, segurança e satisfação.

Ao tratarmos de qualidade externa e interna observamos que no quesito funcionalidade e suas ramificações duas ferramentas se sobressaem das demais: TestLink e Rth. A ferramenta TestLink leva pequena vantagem e atende de melhor forma tais características. Ao tratarmos das “ferramentas menos recomendadas” no quesito, a TestiTool e a TestMaster se encontram praticamente empatadas.

Na questão usabilidade e suas características, percebemos que mais uma vez a ferramenta TestLink é a melhor. A ferramenta proporciona um melhor entendimento e tem maior facilidade no manuseio. As demais também trazem um bom entendimento e facilidade em uso, mas não atendem de forma tão ampla como a ferramenta TestLink. Na análise das características de modelo de qualidade em uso chegamos à conclusão que duas ferramentas mais uma vez se destacaram: Rth e TestLink. Elas correspondem de forma bem satisfatória às exigências mencionadas. Damos, mais uma vez, ênfase a ferramenta TestLink que proporciona maior facilidade a seus usuários no seu manuseio.

Destacamos como a menos recomendada a ferramenta a TestiTool que se aparta negativamente por não proporcionar uma interface amigável e não ter funções importantes como integração com ferramentas externas de rastreamento de defeitos. Ao fim da análise, apontamos a ferramenta TestLink como a melhor dentre as ferramentas analisadas, satisfazendo positivamente a um maior número de características de qualidade da NBR

ISSO/IEC 9126.

FERRAMENTAS		TestLink	Rth	TestMaster	TestiTool
		CARACTERÍSTICAS DA NBR 9126			
MODELO DE QUALIDADE PARA QUALIDADE EXTERNA E INTERNA					
FUNCIONALIDADE	ADEQUAÇÃO	✓	✓	✓	✓
	ACURÁCIA	✓	✓	✓	✓
	INTEROPERABILIDADE	✓	✓	✗	✗
	SEGURANÇA DE ACESSO	✓	✓	✓	✓
USABILIDADE	INTELIGIBILIDADE	✓	✓	✓	✓
	APREENSIBILIDADE	✓	✓	✓	✓
	OPERACIONALIDADE	✓	✓	✓	✓
	ATRATIVIDADE	✓	✓	✓	✓
MODELO DE QUALIDADE PARA QUALIDADE EM USO					
QUALIDADE EM USO	EFICÁCIA	✓	✓	✓	✓
	PRODUTIVIDADE	✓	✓	✗	✓
	SEGURANÇA	✓	✓	✓	✓
	SATISFAÇÃO	✓	✓	✓	✓

LEGENDA:

- ✓ - A ferramenta possui tal característica
- ✗ - A ferramenta não possui tal característica
- ✓ - A ferramenta contempla tal característica com outras particularidade

Quadro 2: Comparativo entre as ferramentas de gerenciamento de teste de software segundo características da NBR ISO/IEC 9126.

Fonte: Elaborado pelos autores (2016).

6. Conclusões

No processo de desenvolvimento de sistemas de informação muitas etapas são realizadas e cada uma têm fundamental importância, tanto de sua existência quanto de sua organização e execução. Essas etapas vão desde a elicitação de requisitos a implantação do projeto. A etapa de teste tem como finalidade identificar os erros e ajudar a aumentar a qualidade. Essa etapa visa que o produto gerado seja disponibilizado com o menor número de defeitos/erros possíveis.

Utilizar ferramentas *open source* para o gerenciamento de teste pode ser uma boa alternativa para auxiliar o processo de teste. Porém, ainda é um problema escolher a ferramenta ideal para o projeto desenvolvido, tendo em vista a quantidade de opções disponíveis no mercado. Este trabalho científico visou contribuir para facilitar o desenvolvimento de sistema de informação apresentando um comparativo entre as ferramentas TestLink, Rth, TestiTool e TestMaster. Todas essas ferramentas são destinadas ao gerenciamento de teste de software. Também realizamos uma análise de cada ferramenta seguindo parâmetros de qualidade contidos na NBR ISO/IEC 9126. Os resultados da pesquisa foram obtidos a partir das análises realizadas pelos presentes pesquisadores.

Como resultado, identificamos qual das ferramentas estudadas gerencia de melhor maneira a atividade de teste de software. Apontamos que as ferramentas Rth e TestLink foram as melhores diante das especificidades gerais do processo de gerenciamento de teste de software. Ressaltamos ainda que a ferramenta TestLink foi tida como a melhor ferramenta, pois atende de melhor forma os pontos analisados e as características da NBR ISO/IEC 9126. Portanto, concluímos que a ferramenta TestLink é a melhor ferramenta para utilização em gerenciamento de teste

de software. Esse resultado corrobora a comparação apresentada na Figura 1 e 2.

Por outro lado, com base nos experimentos e análises, aferiu-se que as demais ferramentas analisadas também são boas para gestão de teste de software. Elas não apresentaram ineficiência, ou seja, todas as demais ferramentas também contemplam parcialmente as características analisadas, mas não apresentam algumas particularidades diferenciais como na ferramenta TestLink. A ferramenta Rth foi considerada a segunda melhor ferramenta diante da pesquisa.

Como trabalhos futuros, ressaltamos a importância da execução de uma pesquisa com ferramentas proprietárias de gestão de teste de software para fazer uma análise de custo benefício. Além disso, uma análise mais aprofundada das demais particularidades da NBR ISO/IEC 9126 podem ser realizadas, visto que elas podem contribuir positivamente de melhorias no processo de desenvolvimento de sistemas de informação.

Referências

- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 9126-1: Engenharia de software – qualidade de produto**. Parte 1: modelo de qualidade, 2003.
- BEIZER, B. **Software testing techniques, 2nd ed**. New York: Van Nostrand Reinhold Company, 1990.
- BEIZER, B. **Black box testing: Techniques for functional testing of software and systems**. Jhon Willey, 1995.
- BITTENCOURT, M. **Gestão de Teste e Defeitos com Testlink e Mantis**. 1º Edição v.1, 2013
- CAETANO, C. **Gestão de Teste: Ferramentas open source e melhores práticas na gestão de teste**. Revista Engenharia de Software Magazine, Ano I, 3ª Edição, Brasil, 2008.
- HARROLD, M. J. **Testing: A roadmap**. In: 22th International Conference on Software Engineering, 2000.
- MALDONADO, J. C. **Crêterios potenciais usos: Uma contribuição ao teste estrutural de software**. Tese de Doutorado. DCA/FEE/UNICAMP, Campinas, SP, 1991.
- MALDONADO, J. C.; VINCENZI, A. M. R.; BARBOSA, E. F.; SOUZA, S. R. S.; DELAMARO, M. E. **Aspectos teóricos e empíricos de teste de cobertura de software. Relatório 31, Instituto de Ciências Matemáticas e de Computação – ICMC-USP, 1998.**
- MYERS, G. J. **The art of software testing**. Wiley, New York, 1979.
- PRESSMAN, R. S. **Software Engineering – a practitioner’s approach**. 5ª Edição, McGraw-Hill, 2000.
- PRIKLADNICKI, R. **Desenvolvimento Distribuído de Software: Desenvolvimento de software com equipes distribuídas**. Rio de Janeiro: Elsevier, 2006.
- REIS, M. F. S. **VVTESTE: Ambiente de geração e gerenciamento de testes e de defeitos como apoio aos processos de Verificação e Validação do MPS.BR**, 2012.
- RTH. **Requeriments and Testing Hubs**. Disponível em: <<http://www.rth-is-quality.com>>. Acesso em: 25 out. 2015.
- SOMMERVILLE, I. **Engenharia de Software**. 8ª. ed. São Paulo: Pearson, 2007.
- TESTITOOL, M., Disponível em: <<http://majordomo.com/testitool/>>. Acesso em: 25 out. 2015.
- TESTLINK, **Test Management Tool = TestLink**. Disponível em: <[http://www.teamst.org./](http://www.teamst.org/)>. Acesso em: 25 out. 2015.
- TESTMASTER, **About TestMaster**. Disponível em: <<http://testmaster.sourceforge.net/>>. Acesso em: 25 out. 2015.
- TORRES, G.; BRAGA, P. **Open source: uma nova desconstrução de paradigmas na educação moderna**, 2015.
- VINCENZI, A. M. R.; **Subsídios para o estabelecimento de estratégias de teste baseadas na técnica de mutação**. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação, ICMC-USP, São Carlos – SP, 1998.
- VINCENZI, A. M. R.; **Orientação a objetos: Definição e análise de recursos de teste e validação**. Exame geral de qualificação, Instituto de Ciências Matemáticas e de Computação (Doutorado em Andamento), ICMC-USP, São Carlos – SP, 2000.
- WEISS, C. P. **Supporter Test: Ferramenta de apoio ao gerenciamento de testes de software**. Dissertação apresentada ao Centro de Ciências Exatas e Tecnológicas UNIVATES. Lajeado, 2013.