

Utilizando Ferramentas de Software Livre para Implementação de Testes a partir do Processo de Verificação constante no Modelo de Referência do MPS.BR

Olavo Nylander Brito Neto¹, Sandro Ronaldo Bezerra Oliveira¹

¹Programa de Pós-Graduação em Ciência da Computação – Instituto de Ciências Exatas e Naturais - Universidade Federal do Pará (UFPA)
Caixa Postal 479 – 66075-110 – Belém – PA – Brasil

onbn@ufpa.br, srbo@ufpa.br

Abstract. *Considering the importance of the verification process present in MR-MPS in organizations that develop products and product components, this paper describes a systematic implementation of Tests by means verification process with the use of free software tools Spider-CL, TestLink, JUnit and Selenium, based on best practices suggested by MPS-BR.*

Resumo. *Considerando a importância do processo de verificação presente no MR-MPS nas organizações desenvolvedoras de produtos e componentes de produtos, este artigo descreve uma implementação sistematizada de Testes a partir do processo de Verificação com o uso das ferramentas de software livre Spider-CL, TestLink, JUnit e Selenium, com base nas boas práticas sugeridas pelo MPS-BR.*

1. Introdução

Objetivar qualidade é um fator crucial nas empresas desenvolvedoras de produtos e serviços correlatos, que implica tanto na melhoria destes resultados quanto no processo de produção e distribuição (SOFTEX, 2011a). Neste contexto, surgiu o MPS-BR (Melhoria de Processo de Software Brasileiro), que trata-se de um programa mobilizador coordenado pela Associação para Promoção da Excelência do Software Brasileiro (SOFTEX), tendo como objetivos a criação e aprimoramento do modelo MPS (MR-MPS) e a disseminação e adoção deste modelo em todo país com foco principal em pequenas e médias empresas (PME) e abrangendo grandes organizações.

Dentre os processos adotados pelo MR-MPS, destaca-se o processo de Verificação (VER). Este processo tem como propósito confirmar que cada serviço e/ou produto de trabalho do processo ou do projeto atende apropriadamente os requisitos especificados (SOFTEX, 2011a). Seu objetivo é assegurar que o software cumpra com suas especificações e atenda às necessidades dos clientes que estão pagando por eles (Sommerville, 2003). Assim, nota-se na literatura especializada a ausência de uma metodologia que incorpore recomendações que realizem a implementação deste processo tomando como base ferramentas de software livre. Tais informações podem ser melhor evidenciadas a partir dos trabalhos relacionados discutidos na Seção 2.

O Projeto SPIDER – *Software Process Improvement: DEvelopment and Research*, no qual este trabalho está inserido, agrega subprojetos que analisam a aderência de cada processo organizacional constante no MR-MPS a partir do uso de ferramentas de software livre presentes no mercado e/ou desenvolvidas pelo projeto, quando necessário (Oliveira *et al.*, 2011). No contexto deste trabalho, que foca na implementação de Testes usando como base as recomendações do processo de

Verificação do MR-MPS, a ferramenta Spider-CL foi desenvolvida com o objetivo de sistematizar a criação e aplicação de critérios objetivos nas análises promovidas pela organização (Barros & Oliveira, 2010). Complementando a proposta de sistematização, foi utilizada a ferramenta *TestLink* para gerir as demais atividades necessárias no processo a fim de planejar e acompanhar as etapas necessárias. A versão da Spider-CL encontra-se disponível em <http://www.spider.ufpa.br/>, acessando o item de menu Downloads.

Este trabalho objetiva analisar uma proposta de sistematização de Testes a partir do processo de Verificação do MR-MPS, utilizando um conjunto de ferramentas de software livre (todas as ferramentas utilizadas possuem licença GPL – General Public License (GNU, 2012)). Além desta seção introdutória, o trabalho está organizado da seguinte forma: na Seção 2 são apresentados trabalhos relacionados presentes na literatura especializada que tratam de abordagens similares; na Seção 3 é apresentada a metodologia de implementação do processo de Verificação com apoio de softwares livres, como as ferramentas Spider-CL na criação de *checklist*, *TestLink* para gerência do processo e ferramentas de teste como *JUnit* para testes unitários e *Selenium* para testes automatizados; a Seção 4 faz uma análise de aderência das ferramentas ao MR-MPS; e, por fim, a Seção 5 realiza uma visão geral do trabalho apontando resultados.

2. Trabalhos Relacionados

Muitas propostas de implementação do MPS-BR a partir da utilização de ferramentas são encontradas na literatura especializada, como pode ser visto em (Almeida *et al.*, 2011), que propõem um apoio ferramental de forma parcial aos processos de Gerência de Requisitos, Verificação e Validação em um ambiente integrado. Outro trabalho pode ser visto em (Barreto, 2006) que, através da estação TABA, apresenta a ferramenta *VerificationManager* com objetivo de realizar verificação de software voltada para modelo de construção de Ambientes de Desenvolvimento de Software Orientados a Organização (ADSOrgs).

No contexto da criação de um protótipo de ferramentas para geração automática de testes focando no domínio da área espacial, Ambrósio *et al.* (2008) apresentam o projeto Ambientes de Testes baseados em Injeção de Falhas por Software (ATIFS), que tem como objetivo explorar técnicas e métodos de testes para a validação de software em aplicações espaciais.

De forma semelhante ao apresentado em (Yoshidome *et al.*, 2010), este artigo aborda uma estratégia para a implementação de Testes usando o processo de Verificação do MR-MPS, fundamentado em boas práticas de uso de ferramentas de software livre, desenvolvidas no contexto de um projeto acadêmico, e analisa a aderência de ferramentas presentes no mercado semelhante aos artigos supracitados. A presente pesquisa apresenta uma estratégia de utilização de ferramentas já existentes de software livre (*TestLink*, *JUnit*, *Selenium*) em conjunto com uma ferramenta desenvolvida no Projeto SPIDER (Spider-CL) para apoio à implementação do MPS-BR.

3. Metodologia de Implementação do VER com Apoio de Ferramentas Livre

Esta seção apresenta a metodologia proposta para implementação sistematizada de Testes a partir do processo de Verificação (VER) constante no MR-MPS, através do uso das ferramentas de software livre: (1) Spider-CL para criação de *checklist*, desenvolvido na linguagem Java e requer um servidor de aplicação e banco de dados Mysql; (2) *TestLink* (<http://www.teamst.org/>), para definição e gerência de casos de teste, oferecendo suporte ao processo de verificação, desenvolvido em PHP e requer servidor

web com suporte a linguagem e banco de dados Mysql; e as ferramentas de teste (3) JUnit para a implementação de casos de teste unitários integrado ao código fonte (<http://www.junit.org/>), desenvolvida para a linguagem Java e adaptada para C#, Python, Fortran e C++, e (4) Selenium para testes automatizados disponível para *browsers* (<http://seleniumhq.org/>). Ressalta-se que a escolha das ferramentas deu-se com base nos resultados obtidos e relatados em (Oliveira *et al.*, 2011).

A ferramenta Spider-CL foi desenvolvida pelo Projeto SPIDER. Apresenta uma abordagem através de desenvolvimento de *checklists*, usando critérios objetivos, que permite a aplicação em diversos contextos e mantém um registro permanente de todas as utilizações dos modelos desenvolvidos. Seu objetivo é compor um conjunto de critérios organizados, funcionando como um *template* que pode ser reutilizado diversas vezes em várias aplicações. Mais detalhes acerca do uso da ferramenta, consultar o seu manual disponível em <http://www.spider.ufpa.br>.

Neste trabalho compreende-se metodologia como a codificação de um conjunto de práticas recomendadas, por vezes acompanhada de material de treinamento, programas de educação formal, planilhas, diagramas, tomando como parte um método (processo com uma sequência de passos, para construir um software) (Pressman, 2006). Esta metodologia procura agregar boas práticas para o uso de softwares livres de apoio ao processo de VER.

A metodologia foi organizada tomando como base o processo proposto em (Ackerman *et al.*, 1989), que contempla as atividades de: Planejar, responsável pela identificação de produtos de trabalho e a definição de uma estratégia de verificação contendo cronograma, métodos, ferramentas, entre outros; Especificar, responsável pela definição de critérios e procedimentos para a realização da verificação nos produtos de trabalho; Executar, responsável pela realização das atividades de verificação previamente planejadas e pela identificação e registro dos defeitos detectados; e Reportar, responsável pela análise dos resultados das atividades de verificação e disponibilização destes para as partes interessadas.

3.1. Planejar

Considerando um ambiente onde estejam devidamente instaladas e configuradas as ferramentas de apoio ao processo de VER, a metodologia inicia com a definição dos critérios de maior interesse ao processo. Estes critérios têm como objetivo selecionar quais produtos de trabalho serão verificados a partir de uma análise entre todos que serão produzidos ao longo do projeto (SOFTEX, 2011b). Ainda, segundo a SOFTEX, recomenda-se como estratégia para seleção de produtos de trabalho as contribuições para o alcance dos objetivos e requisitos do projeto. Para esta implementação, será levado em consideração o Plano de Projeto (produto de trabalho que define como o projeto é executado, monitorado, controlado e encerrado, documentando o conjunto de saídas dos processos de planejamento do projeto) no qual estes artefatos estão presentes. A Figura 1 apresenta duas telas da ferramenta Spider-CL, que permite a manutenção e a aplicação de critérios objetivos a partir do uso de *checklists* como forma de seleção dos produtos de trabalho.

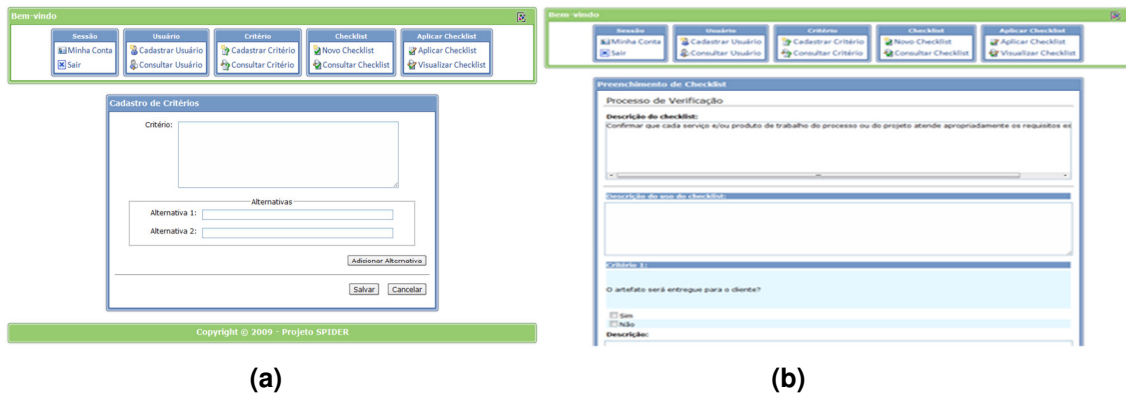


Figura 1. Informações do Checklist utilizando a ferramenta Spider_CL: (a) Cadastro de Critério; (b) Aplicação do Checklist.

Adicionalmente, a estratégia adotada para a execução do processo deve ser registrada num Plano de Verificação, que deverá ser construído a partir do Plano de Projeto citado anteriormente, definindo uma estratégia com: os (1) procedimentos; (2) a infraestrutura necessária; e (3) as responsabilidades pelas atividades pré-definidas neste plano. Estas informações podem ser registradas na ferramenta *TestLink*, a partir da funcionalidade “Gerenciar Plano de Teste”, que auxiliará outras funcionalidades no projeto. Devem estar presentes os métodos que serão utilizados para verificação de cada produto de trabalho, garantindo revisão por pares e testes (SOFTEX, 2011b). Este plano pode ser cadastrado usando a ferramenta *TestLink*, no qual destacam-se alguns pontos deste plano presente em itens específicos da ferramenta, como as funcionalidades “Gerenciar Plano de Teste”, “Especificar”, “Inventário”, “Executar Testes” e “Relatórios e Métricas”, a fim de facilitar a organização durante o acompanhamento deste processo, como pode ser visto nas colunas laterais da ferramenta, a partir da Figura 2.

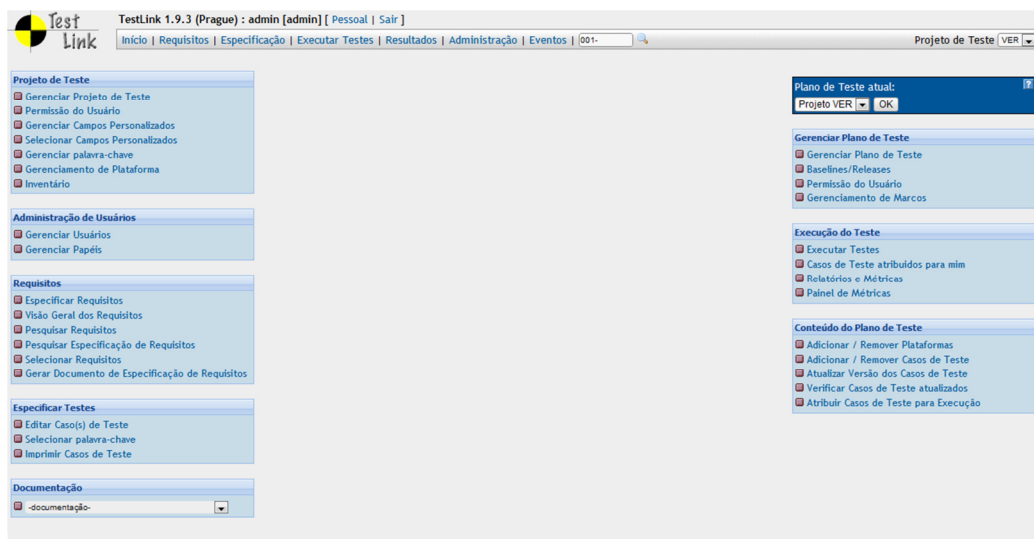


Figura 2. Tela Principal da Ferramenta TestLink

3.2 Especificar

Após a execução da atividade de planejamento, os critérios e os procedimentos que serão utilizados na verificação de cada produto de trabalho, previamente selecionado, são de fato definidos. Quanto aos critérios, estes visam representar de forma confiável os objetivos especificados junto ao cliente, visando manter a integridade do processo e consequentemente seu objetivo. Os critérios de aceitação dos produtos de trabalho

podem ser definidos e, posteriormente, geridos pela ferramenta *Spider-CL* e sua definição poderá ser feita a critério da organização que está implantando. O MR-MPS sugere alguns critérios que podem ser usados a partir de algumas referências citadas em (SOFTEX, 2011b).

Já os procedimentos (seleção dos critérios objetivos através da *Spider-CL* e os casos de teste mantidos na ferramenta *TestLink* a partir da sua funcionalidade “Especificação”, que contém a descrição do casos de testes e suas peculiaridades como pré-condição, pós condição, resumo, ação e demais seções que serão utilizadas de acordo com a organização que fará uso) passam ser armazenados na ferramenta *TestLink* a fim de auxiliar na execução dos mesmos, usando a funcionalidade “Executar Testes”. Na Figura 3 é possível identificar um caso de teste que representa um procedimento a ser executado.

Após o registro dos procedimentos a serem executados, a aplicação dos mesmos será realizada a partir da ferramenta, facilitando o acompanhamento até o final do processo. Contudo, será necessário definir um ambiente para verificação, podendo ser gerenciado através da ferramenta *TestLink* que apresenta a funcionalidade “Inventário”, específica para o registro dos itens necessários na alocação de recursos. Assim, esta funcionalidade apresenta um planejamento dos itens a serem utilizados como ferramentas, recursos de hardware, infraestrutura de rede e qualquer outro item necessário.

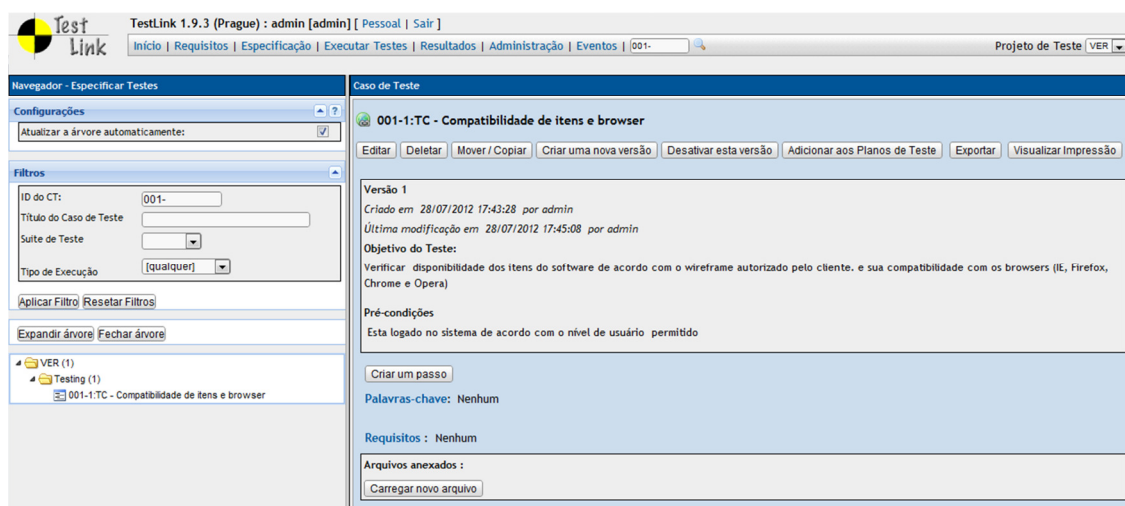


Figura 3. Procedimento a ser executado no processo de VER.

3.3 Executar

Durante a execução do processo, conforme o planejado, deve estar presente nos procedimentos de verificação a revisão por pares e testes, a fim de garantir a integridade dos produtos de trabalho gerados pelo projeto durante o processo. O acompanhamento da execução é registrada na ferramenta *TestLink* (ver Figura 4) e descrições podem ser incluídas para facilitar futuras correções e, assim, entregar produtos de trabalho de forma íntegra com o especificado nos requisitos. Os registros podem ser feitos similares aos casos de teste, facilitando a execução e registro dessas informações.

Os defeitos encontrados podem ser classificados a critério da organização com o intuito de definir parâmetros que possam ser considerados suficientes para o risco do projeto e o impacto na qualidade do produto. Após a identificação dos defeitos, estes devem sofrer ações corretivas, ficando a critério da organização executar novamente as estratégias de verificação planejadas, para garantir que os defeitos foram removidos

adequadamente e que novos defeitos não foram incluídos a partir dessas correções (SOFTEX, 2011b). Na funcionalidade “Especificação” do *TestLink* são adicionadas sugestões de ações corretivas com o propósito de corrigir erros gerados durante o desenvolvimento. E em seguida, se necessário, o registro das ações corretivas efetivamente realizadas de fato são cadastradas na mesma ferramenta.

Os testes a serem executados sobre o código fonte são escolhidos a critério da organização responsável, com o intuito de expor defeitos latentes presentes e corrigi-los, antes da entrega do produto. Segundo Sommerville (2003), para um teste bem-sucedido a detecção de defeitos é aquela que faz com que o sistema opere incorretamente, e consequentemente, expõe os defeitos existentes. Com esse propósito são utilizadas ferramentas de testes voltadas para o código-fonte.

Uma delas é o Selenium que se apresenta integrado ao browser facilitando a ação de testes automatizados para aplicações *web*, facilitando a execução dos testes automatizados e gerando seu código fonte integrado com outros tipos de testes, como o JUnit. Este possui o propósito de testes unitários o qual, quando aplicados juntos, facilita a homogeneidade dos testes, uma vez que os testes podem ser tratados como testes unitários seguindo o padrão do JUnit, como podem ser vistos na Figura 5.

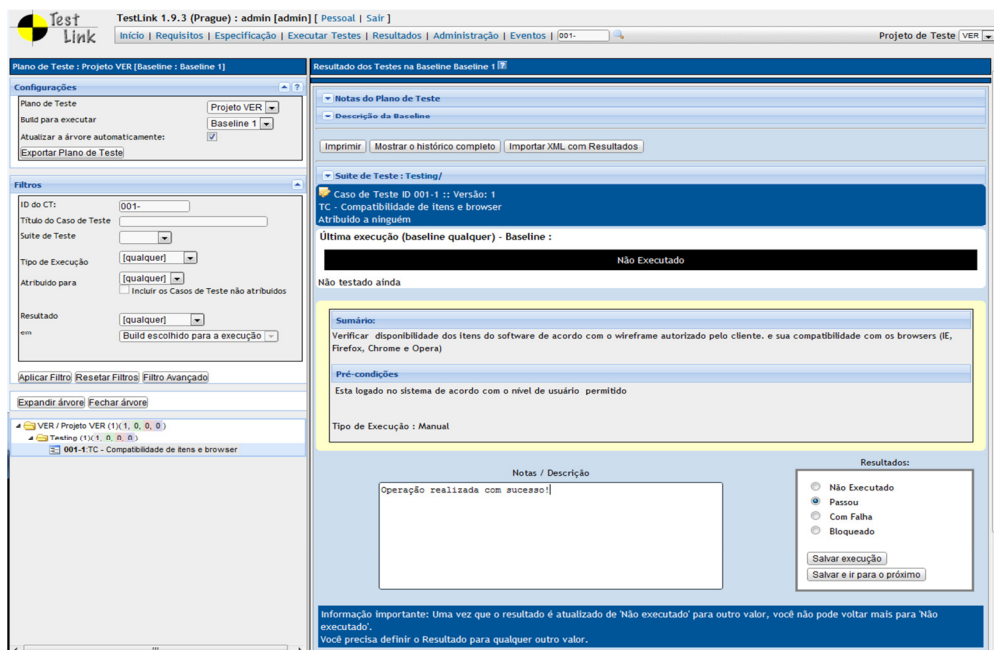


Figura 4. Execução dos procedimentos pré-cadastrados.

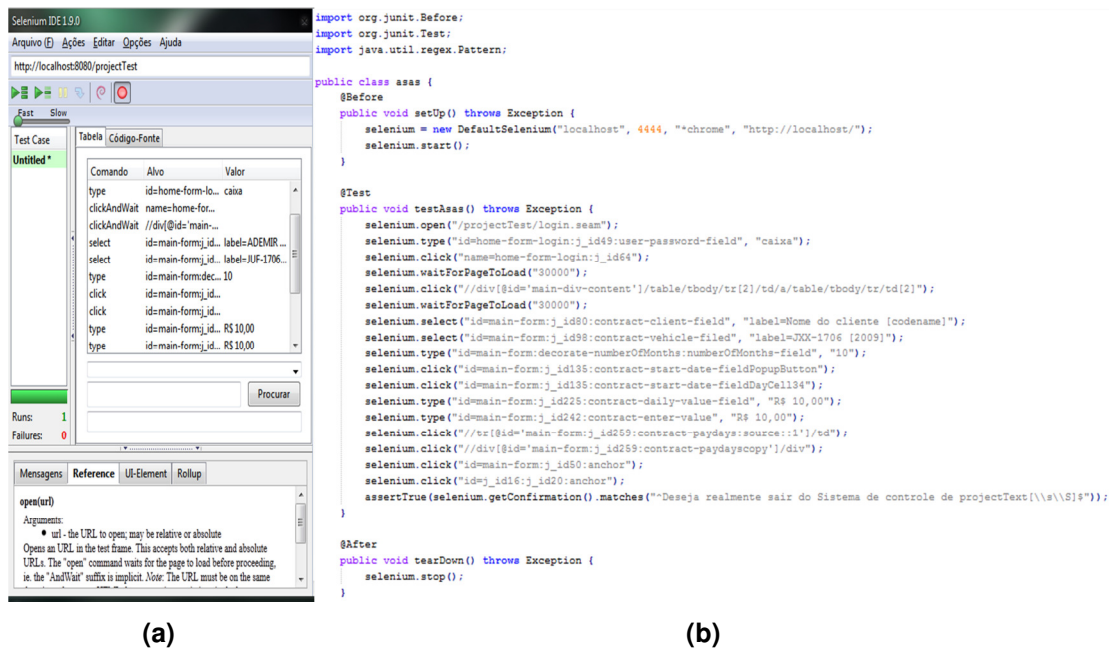


Figura 5. Ferramentas de Teste: (a) Selenium; (b) JUnit aplicado junto com o Selenium.

3.4 Reportar

Ao fim da execução e correções necessárias, são realizadas análises dos resultados obtidos em cada atividade de verificação e disponibilizada para as partes interessadas do projeto (SOFTEX, 2011b). Para este objetivo a ferramenta *TestLink* gera de forma automática relatórios que facilitam o acompanhamento do projeto, numa visão mais ampla, a partir da funcionalidade “Relatórios e Métricas”, mantendo presente qualquer informação que sofra alteração durante o processo. Estes relatórios podem, ainda, ser exportados em alguns formatos de arquivos populares, como podem ser observados nos diversos modelos de relatórios disponíveis na Figura 6. Por fim, estes devem ser disponibilizados para as partes interessadas, especificados nos recursos humanos definidos no Plano de Projeto, usando algum meio de comunicação: e-mail, publicação no site interno, repositório do projeto, etc.

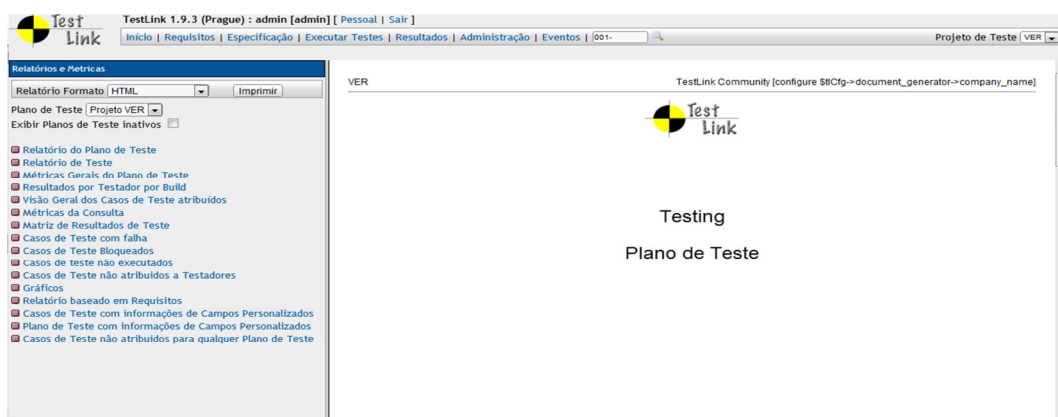


Figura 6. Modelos de Relatórios gerados pela Ferramenta *TestLink*.

De uma maneira geral, a integração entre as ferramentas pode ser vista na Figura 7, onde o processo é iniciado através da ferramenta Spider-CL a partir da geração de *checklist* que terá a finalidade de apoiar a seleção dos produtos de trabalho (artefatos) que serão verificados. A ferramenta TestLink centralizará o acompanhamento das atividades do processo: registro do plano de teste; acompanhamento dos testes em

execução; registro de defeitos e correções; e disponibilização dos resultados. Complementarmente, ressalta-se que as ferramentas Selenium e JUnit são utilizadas para a execução dos testes a partir dos Casos de Testes definidos e mantidos na ferramenta TestLink.

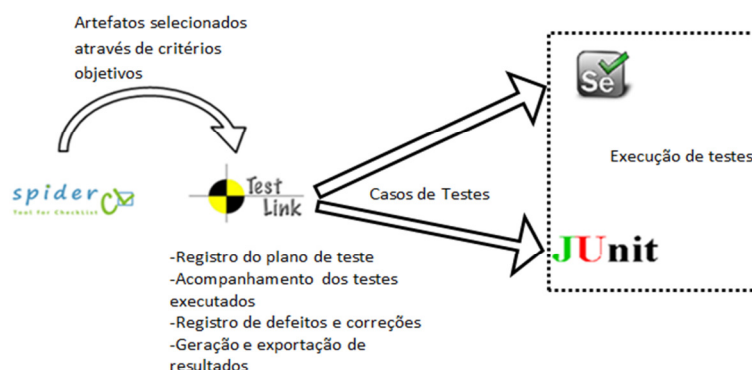




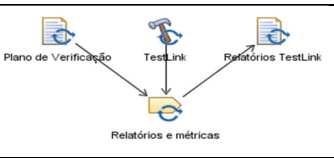


Figura 7. Comunicação entre as Ferramentas propostas.

4. Aderência da Metodologia Proposta ao MR-MPS

A análise de aderência entre a metodologia proposta ao modelo MR-MPS é feita através do mapeamento das práticas sugeridas na Seção 3 com os resultados esperados do processo de Verificação constante no Guia Geral do MPS-BR (SOFTEX, 2011a), como visto no Quadro 1. É importante ressaltar que a metodologia restringe-se a apresentar boas práticas para que seja possível contemplar os resultados esperados do processo de VER em conjunto com práticas definidas no processo de software da organização. Resultado Esperado (RE) é “um resultado observável quando o propósito do processo é alcançado com sucesso” ISO/IEC (2008). O processo de VER do MPS.BR, constante no nível D, é formado por seis resultados esperados.

Quadro 1. Mapeamento da Metodologia com os Resultados Esperados de VER do MR-MPS

RE do MR-MPS	Ferramentas da Metodologia	Funcionalidades/ Práticas da Metodologia	Insumos da Metodologia
VER 1	Spider-CL	Seleção de artefatos através da criação de <i>checklist</i> com critérios objetivos	
VER 2	TestLink	Cadastro do Plano de Verificação (teste)	
VER 3	TestLink	Inclusão no Plano de Verificação dos critérios e procedimentos (Plano de Teste); Registro do ambiente de verificação (Inventário)	
VER 4	TestLink, Selenium, JUnit	Acompanhamento dos testes e registros dos resultados (Casos de Teste); Realização de testes unitários e testes automatizados (Executar Testes e Ferramentas de Teste)	
VER 5	TestLink	Acompanhamento da identificação e registros de defeitos (Casos de Teste)	
VER 6	TestLink	Geração e exportação de relatórios e métricas (Resultados)	

5. Conclusões

Como observado em lições aprendidas de implementações realizadas no programa de melhoria do processo, o emprego de ferramentas sistematizadas pode reduzir consideravelmente o tempo de implementações do MR-MPS, e organizar documentos de forma harmoniosa. Tratando-se de ferramentas livres, pode existir uma redução de custos para a organização.

Com o objetivo de agilizar o processo de implementação do programa MPS-BR, este trabalho apresentou uma solução para implementação de Testes a partir do processo de VER, utilizando ferramentas de software livre. Apenas o uso da metodologia não é suficiente para atender o MPS-BR, porém define boas práticas para o uso do ferramental de forma aderente ao MR-MPS. Atualmente, a metodologia encontra-se em implementação em uma empresa parceira do projeto SPIDER, fato este sendo o critério definido para utilização das ferramentas expostas acima. Trabalhos futuros descreverão

os resultados obtidos com esta execução, bem como as lições aprendidas com o uso destas ferramentas no contexto definido neste trabalho.

6. Agradecimentos

Este trabalho está recebendo o apoio financeiro da CAPES através de bolsa institucional de pesquisa concedida ao PPGCC-UFPA.

Referências

- Ackerman, A., Buchwald, L., Lewski, F. (1989) "Software Inspections: An Effective Verification Process", IEEE Software, vol. 6, no. 3, p.31-37.
- Almeida, G., Ramos, B., Carvalho, J., Barcelos, M., Silva, S., Vasconcelos, A. (2011) "Apoio aos Processos de Gerência de Requisitos e Verificação e Validação em um Ambiente Integrado", VII Workshop Anual do MPS (WAMPS 2011) Campinas, SP, Brasil, 2011.
- Ambrósio, A., Francisco, F., Cardoso, L., Santiago, V., Arias, R., Loureiro, G. (2008) "Experiência em Projetos e Uso de Técnicas de Verificação e Validação de Software em Aplicações Espaciais no INPE" INPE-15182-NTC/374, Ministério da Ciência e Tecnologia – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP, Brasil.
- Barros, R. S., Oliveira, S. R. B. (2010) "Spider-CL: Uma Ferramenta de Apoio ao Uso de Critérios Objetivos no Contexto da Qualidade de Software". Anais da II Escola Regional de Informática – ERIN 2010. Manaus-AM.
- Barreto, A. O. S. (2006) "Apoio à Verificação de Software em Ambientes de Desenvolvimento de Software", Dissertação de Mestrado - COPPE/UFRJ http://teses.ufrj.br/COPPE_M/AndreaOliveiraSoaresBarreto.pdf, Julho.
- GNU Project (2012) "General Public License", <http://www.gnu.org>, Junho.
- ISO/IEC (2008) "Engenharia de sistemas e software – Processos de ciclos de vida de software", ABNT NBR ISO/IEC 12207:2009.
- Oliveira, S. R. B., Yoshidome, E., Furtado, J., Neiva, J., Paulo, M., Barros, R., Souza, M., Mezzomo, L. (2011) "Uma Proposta de Solução Sistêmica de um SUITE de Ferramentas de Software Livre de Apoio à Implementação do Modelo MPS.BR", Em: Revista do Programa Brasileiro de Qualidade e Produtividade em Software, 2ª edição, páginas 103-107.
- Pressman, R. S. Engenharia de software. Tradução José Carlos Barbosa dos Santos. Rio de Janeiro: McGraw-Hill, 2006. 1056 p.
- SOFTEX (2011a) "MPS.BR – Melhoria de Processo do Software Brasileiro – Guia Geral", ISBN 978-85-99334-22-5.
- SOFTEX (2011b) "MPS.BR – Melhoria de Processo do Software Brasileiro – Guia de implementação: - Parte 4: Fundamentação para Implementação do Nível D do MR-MPS", ISBN 978-85-99334-27-0.
- Sommerville, I. (2003), Engenharia de Software, Addison Wesley, 6th Edição.
- Yoshidome, E. Y. C., Souza, M. R. A., Lira, W. M. P., Oliveira, S. R. B. (2010) "Uma Implementação do Processo de Gerência de Projetos Usando Ferramentas de Software Livre". Anais do VI Workshop Anual do MPS.BR - WAMPS, Campinas-SP.