

UMA FERRAMENTA PARA APOIO AO USO DO ATDD

Pedro Almir Martins de Oliveira (bolsista do PIBITI/UFPI), Pedro de Alcântara dos Santos Neto (Orientador, Depto. de Informática e Estatística – UFPI)

Introdução

As exigências por software com maior qualidade têm motivado a definição de métodos e técnicas para que o desenvolvimento de software atinja os padrões de qualidade impostos. Com isso, o interesse pela atividade de teste de software vem crescendo nos últimos anos (MYERS, 2004). Vários pesquisadores têm investigado os diferentes critérios de teste, buscando uma estratégia de teste com baixo custo de aplicação, mas ao mesmo tempo, com grande capacidade de revelar erros.

Esta busca por novas estratégias deu origem a uma nova técnica denominada Desenvolvimento Dirigido por Testes de Aceitação (ATDD – Acceptance Test Driven Development). O ATDD procura aperfeiçoar o desenvolvimento fazendo uso dos aspectos benéficos do teste de aceitação, como, por exemplo, ajudar na comunicação dentro da equipe de desenvolvimento, permitir que o progresso dos testes seja medido e monitorado, além de promover maior aproximação dos usuários finais do processo de desenvolvimento (AZDIC, 2009).

As ferramentas para o ATDD permitem a geração de um teste bastante inteligível aos usuários finais, ao tempo em que servem como uma forma de documentação executável do sistema. Alguns autores também chamam os testes de aceitação criados com o apoio dessas ferramentas como requisitos executáveis (CRISPIN e GREGORY, 2009).

No entanto, essas ferramentas possuem espaço para melhoria, principalmente na diminuição do esforço necessário para seu uso. Um exemplo disso é a ferramenta FitNesse (FITNESSE, 2012). Seu principal objetivo é facilitar a automação de testes de aceitação a partir da disponibilização de um mecanismo simples de definição de testes e padrões para facilitar sua ligação a um sistema de software.

Neste contexto, apresentamos a ferramenta SELF¹, que foi desenvolvida com o objetivo de aliar a utilidade do teste de regressão com a clareza do teste de aceitação, preenchendo as lacunas deixadas pelas ferramentas de apoio ao ATDD. A ideia é criar um ambiente para teste que reduza o esforço nessa atividade e facilite o acompanhamento por parte dos clientes.

Metodologia

Para alcançar os objetivos e metas traçados neste projeto, ele foi dividido em cinco fases: Fase Teórica (FT); para levantamento do estado da arte sobre o tema; Fase de análise (FA), para o estudo sobre as principais ferramentas de testes de software; Fase de Desenvolvimento (FD), para o desenvolvimento da ferramenta proposta; Fase de Experimentação (FE), para a realização de um estudo experimental com o intuito de avaliar o ganho obtido com a utilização da ferramenta; e Fase de Finalização (FF), para análise dos resultados obtidos, criação da documentação da ferramenta e escrita do artigo referente à mesma.

¹ Disponível em: <http://ufpi.br/pasn>

Resultados e Discussão

Atualmente, a SELF trabalha com as ferramentas de teste funcionais Selenium IDE (SELENIUM, 2004) e Canoo Web Test (CANOO, 2003), além da ferramenta de teste de aceitação FitNesse. No entanto, ela foi arquitetada para ser extensível, podendo-se adicionar componentes para a extração de testes a partir de scripts de qualquer ferramenta de teste funcional e componentes que gerem testes para qualquer ferramenta de teste de aceitação.

Para uma melhor abrangência da SELF, outra ferramenta foi adicionada à sua arquitetura, a TWork (OLIVEIRA *et. al.* 2011), responsável por extrair os dados do teste funcional, tais como valores de entrada e saída, funcionalidade e campos utilizados.

A utilização da SELF baseia-se nos seguintes passos (Figura 1a): o primeiro passo é o planejamento dos testes, com seleção das funcionalidades a serem testadas. Após o planejamento do teste, o testador deve especificar os testes, ou seja, pensar nos casos de teste associados à funcionalidade a ser testada. A partir desse ponto, deve ser utilizada uma ferramenta para o teste funcional (*record & playback*) para gravação do teste desejado. Com o teste funcional gerado, é possível executar a SELF para gerá-lo no formato de teste de aceitação, permitindo o reuso do arcabouço gerado para a inserção de mais casos de testes, sendo preciso para isso apenas a inclusão das entradas e saídas esperadas para os novos casos de teste.

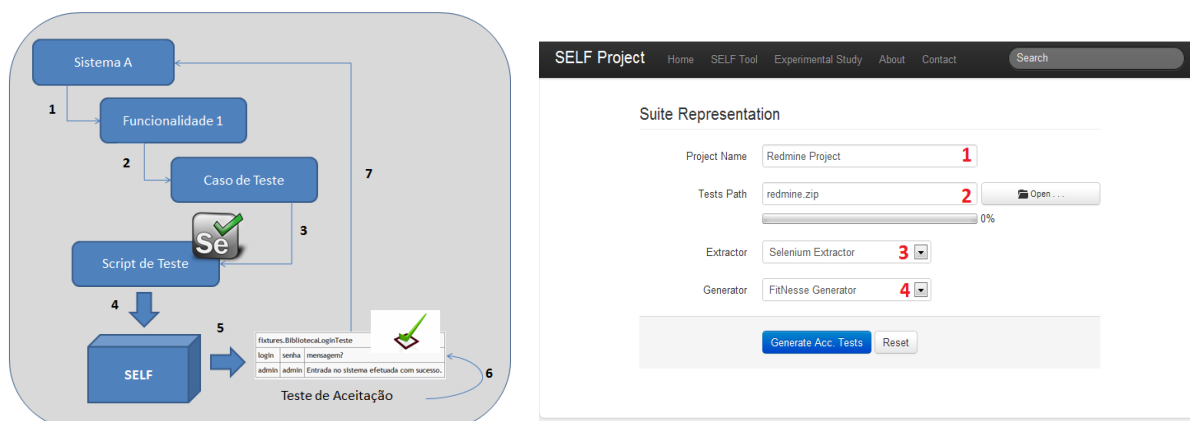


Figura 1: a) Metodologia SELF. b) Tela principal da SELF.

A Figura 1b mostra a versão final da tela principal da SELF, onde o usuário deve informar o nome do projeto, o caminho dos scripts de teste, o extrator e o gerador a serem utilizados. Com a ferramenta pronta foram realizados dois estudos experimentais. O primeiro foi um estudo piloto, que serviu para presumir e mitigar os riscos envolvidos ao se realizar uma avaliação experimental com vários participantes. O segundo estudo foi realizado com um número maior de participantes e seguiu com rigor as prescrições existentes na Engenharia de Software Experimental (WOHLIN *et. al.* 2000).

O propósito desse estudo experimental foi verificar se a SELF consegue gerar testes mais inteligíveis, mais simples de criar (com menor esforço) e com maior qualidade se comparado com a forma usual, ou seja, usando apenas as ferramentas Selenium IDE e JUnit².

Ambos os experimentos foram realizados em duas etapas, com os participantes (alunos do curso de Ciência da Computação da UFPI) divididos em dois grupos, um grupo de controle que criou testes apenas de forma manual e outro grupo que criou testes usando a abordagem tradicional e

² O JUnit é um framework open source, com suporte à criação de testes automatizados na linguagem de programação Java.

abordagem proposta no trabalho (SELF). Antes de cada etapa de execução houve uma etapa para treinamento e após cada etapa de execução os tempos foram colhidos. Analisando a Figura 2 podemos observar o tempo que cada grupo realizou as etapas do estudo.

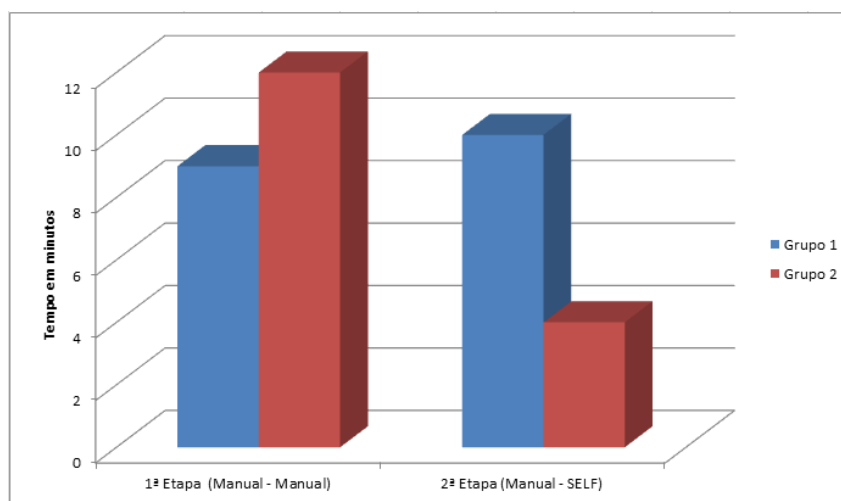


Figura 2: Resultados para o esforço gasto durante o estudo.

Conclusões

Neste trabalho apresentamos a ferramenta SELF, que tenta aliar a utilidade do teste de regressão com a clareza do teste de aceitação. A ideia é criar um ambiente para teste que reduza o esforço nessa atividade e facilite o acompanhamento por parte dos clientes, ao mesmo tempo em que também auxilia aos desenvolvedores.

Com a ferramenta criada, um estudo experimental foi realizado com objetivo de avaliar a SELF, e conseqüentemente o método no qual ela implementa. O estudo mostrou que o uso da ferramenta pode reduzir o esforço para a geração de testes de aceitação a partir de teste funcionais, no âmbito de alunos do curso de Ciência da Computação, desenvolvendo testes para sistemas de informação.

Referências Bibliográficas

- AZDIC, G. **Bridging the Communication Gap: Specification by Example and Agile Acceptance Testing**. [S.l.]: Neuri Limited, 2009.
- CANOO Web Test. **Web Test tool**. Disponível em: <<http://webtest.canoo.com/webtest/manual/WebTestHome.html>>. Acesso em 08/08/2011.
- CRISPIN, G.; GREGORY, J. **Agile Testing: A Practical Guide for Testers and Agile Teams**. [S.l.]: Addison-Wesley Professional, 2009.
- FITNESSE. **The Fully integrated standalone wiki, and acceptance testing framework**. Disponível em: <<http://fitnesse.org>>. Acesso em 08/08/2012.
- IEEE. **Guide to the Software Engineering Body of Knowledge**. Los Alamitos, California, 2004.
- MYERS, G. J. **The Art of Software Testing**; 2. ed. [S.l.]: John Wiley & Sons, 2004.
- OLIVEIRA, W. P.; NETO, P. S.; MOURA, C. D. S.; MEIRA, S. R. L. **Uma ferramenta para automação da geração do relatório de alteração de software**, In: Congresso Brasileiro de Software: Teoria e Prática (CBSOFT) – Sessão de Ferramentas. [S.l.]: [s.n.], 2011.
- SELENIUM. **Web application testing system**. Disponível em <<http://seleniumhq.org/>>. Acesso em 08/08/2011.
- WOHLIN C.; RUNESON, P.; HOST, M; OHLSSON, M.; REGNELL, B.; WESSLEN, A. **Experimentation in Software Engineering: An Introduction**. [S.l.]: Kluwer Academic Publisher, 2000.
- Palavras-chave:** Teste de Software, Teste de Aceitação e Automação de Testes.