

UM MÉTODO PARA APOIO AO USO DO ATDD

Denise Alves da Costa (Bolsista PIBIC/UFPI), Armando Sousa Soares (Colaborador, DIE-UFPI), Pedro de Alcântara dos Santos Neto (Orientador, Depto de Informática e Estatística-UFPI)

Introdução

Os usuários finais de software exigem cada vez mais sistemas computacionais com qualidade e com baixa tolerância a falhas. Para isso faz-se necessário o investimento na atividade de testes de software, que normalmente é onerosa [Myers 2004].

Na tentativa de diminuir o esforço necessário para a execução de testes, muitas ferramentas para automação têm sido desenvolvidas, como por exemplo: JUnit (para testes de unidade), FitNesse [FitNesse 2011] (para testes de aceitação), Selenium [Selenium 2004] e Canoo [Canoo 2003] (para testes funcionais). No entanto, cada grupo de ferramentas ainda possui limitações. Este trabalho buscar unir a simplicidade dos testes de aceitação, a eficiência dos testes funcionais e a garantia de que o sistema não irá regredir com o passar do tempo, proporcionada pelos testes de regressão [Bourque 2005].

Metodologia

A proposta inicial do trabalho foi realizar a integração de duas dessas ferramentas de automação de testes, a FitNesse e o Selenium. A idéia foi a partir de um script de gravação de testes gerado pelo Selenium IDE gerar os casos de testes na FitNesse prontos para a execução.

Após um estudo aprofundado acerca do que já tinha sido desenvolvido anteriormente no âmbito de pesquisa da UFPI e no meio acadêmico como um todo, foi desenvolvido o método para extração dos dados relevantes dos testes gravados com o Selenium para então serem construídos os casos de testes correspondentes na FitNesse. O método é explicitado na Figura 1.

Inicialmente o testador deve definir quais os testes devem ser criados para o sistema em questão. De acordo com os requisitos e funcionalidades do sistema, o mesmo define os passos que devem ser executados pelo teste, assim como os dados de entrada e saída. Em seguida, os passos definidos pelo testador devem ser executados no sistema enquanto são gravados pelo Selenium IDE. O Selenium IDE gera um script contendo informações como dados utilizados no teste, campos da página que foram acessados, comandos acionados e saídas retornadas pelo sistema.

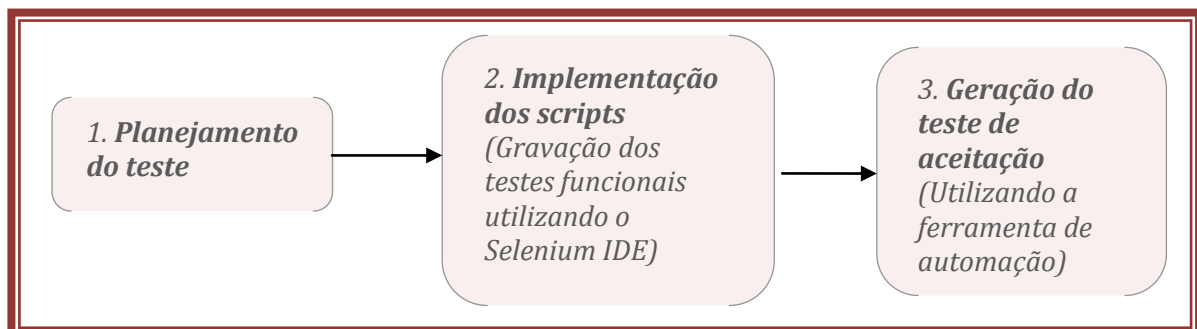


Figura 1: Esquemático do método desenvolvido na IC.

Conjuntamente ao desenvolvimento do método, em um outro trabalho de iniciação, porém tecnológica (PIBIT), foi realizado a implementação de uma ferramenta de apoio ao método. A

ferramenta permitiu a realização de um experimento para avaliar a efetividade do método em uso prático.

Resultados de Discussões

Normalmente os trabalhos desenvolvidos na área de Engenharia de Software são avaliados em sua eficácia e eficiência por meio de experimentos [Wohlin et al., 2000], onde um grupo de pessoas é selecionado para fazer uso do método desenvolvido para posterior avaliação. Até o momento foram realizados dois experimentos para a análise do método. O primeiro, denominado experimento piloto, contou com a participação de quatro alunos. Para o segundo experimento, convocaram-se alunos voluntários da disciplina de Engenharia de Software I do curso de Bacharelado em Ciência da Computação da UFPI.

O resultado obtido com o experimento piloto pode ser visto na Figura 2. O grupo de quatro alunos selecionados foi dividido em duas duplas chamadas grupos 1 e 2. O grupo 1 desenvolveu os testes de aceitação para o sistema sem utilizar o método de automação, ou seja, de forma usual, considerada aqui como forma manual. O grupo 2 desenvolveu os testes de forma automatizada por meio do método proposto neste trabalho.

A partir da análise do gráfico pode-se notar a eficiência do método proposto, pois o grupo 2 (representado pela cor vermelha no gráfico) mostrou considerável ganho de tempo na construção dos testes de aceitação com relação ao grupo 1 na segunda etapa do processo, em que o grupo 1 utilizou o método manual e o grupo 2 usou a automação com o método desenvolvido neste trabalho.

Em seguida desenvolveu-se outro experimento com nove voluntários da turma de Engenharia de Software I. A divisão dos grupos se deu da mesma forma do experimento piloto. Os resultados desse experimento são mostrados na Figura 3. O gráfico do segundo experimento não apresentou resultados tão animadores quanto o do primeiro. O fato de que para o segundo foi utilizado apenas um caso em cada funcionalidade enquanto que no primeiro foram utilizados dois casos para cada funcionalidade pode ter contribuído para a diferença dos resultados. Fora isso, foram utilizados apenas casos de uso simples, que não são encontrados tão facilmente em sistemas reais e não favorecem nossa abordagem.

Conclusão

A utilização do método mostrou resultados satisfatórios para o experimento piloto e não muito animadores para o segundo experimento realizado. No entanto acredita-se ainda que a aplicação do método desenvolvido durante esse trabalho possa contribuir de modo satisfatório para a aceleração da atividade de testes, de forma a reduzir custos e favorecer a qualidade dos produtos desenvolvidos. Espera-se ainda a realização de novos experimentos com maior quantidade de alunos e também no meio industrial. Os novos experimentos poderão oferecer maior confiabilidade dos resultados e a sua avaliação em um ambiente real.

Apoio

CNPQ – Conselho Nacional de Desenvolvimento Científico e Tecnológico

Figura 1 - Resultados do experimento piloto.

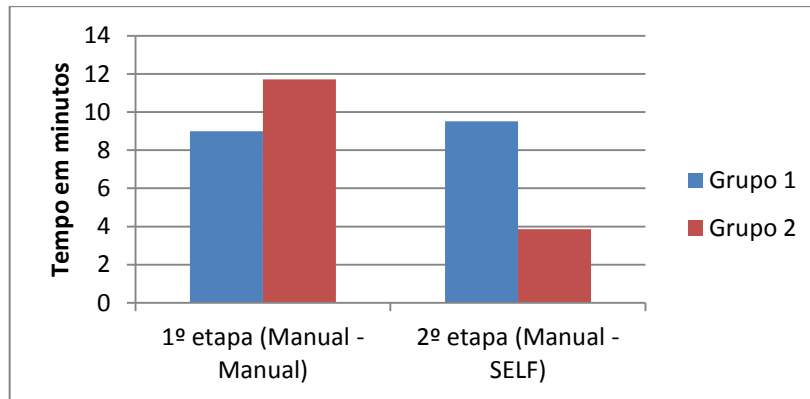
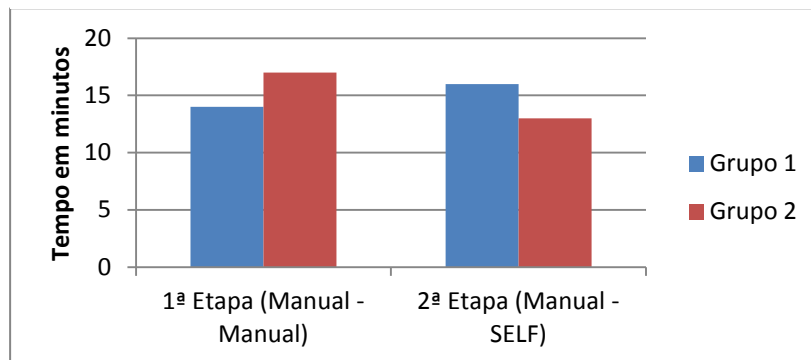


Figura 2 - Resultados do segundo experimento, realizado com alunos da turma de Engenharia de Software I.



Referências

- BOURQUE, P. e DUPUIS, R. **Guide to the Software Engineering Body of Knowledge: Versão 2004 – SWEBOK**. Los Alamitos, California: IEEE Computer Society Press, 2005.
- CANOO. **Free testing of web apps**. <http://webtest.canoo.com>, Julho, 2003.
- FITNESSE. **The Fully integrated standalone wiki, and acceptance testing framework**. <http://fitnesse.org>, Julho, 2011.
- MYERS, G., SANDLER, T. T. **The Art of Software Testing**. [S. I.]. Editora John Wiley & Sons, 2004.
- SELENIUM **Web application testing system**, <http://seleniumhq.org>, Julho, 2004.
- WOHLIN, C., RUNESON, P., HOST, M., OHLSSON, M., REGNELL, B. e WESSLEN, A. **Experimentation in Software Engineering: An Introduction**. Kluwer Academic Publishers. [S. I.: s.n.] 2000.

Palavras-chave: Engenharia de Software. Testes de Aceitação. Testes Funcionais.