UNIVERSIDADE FEDERAL DE PERNAMBUCO

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA

2006.1

FRAMEWORK PARA AUTOMAÇÃO DE TESTES FUNCIONAIS UTILIZANDO O *RATIONAL FUNCTIONAL TESTER*

PROPOSTA DE TRABALHO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE

Aluno – Rafael Oliveira Nóbrega, ron@cin.ufpe.br.
Orientador – Alexandre Marcos Lins de Vasconcelos, amlv@cin.ufpe.br.

20 de junho de 2006

1. Contexto

Atualmente, as organizações de desenvolvimento de software enfrentam um difícil desafio: devem otimizar a qualidade de aplicações de complexidade crescente a fim de entregar soluções que rendam um elevado retorno de investimento e levam a vantagens competitivas para seus clientes.

Neste contexto, qualidade pode ser entendida como um conjunto de características a serem satisfeitas em um determinado grau, de modo que o produto de software atenda às necessidades explícitas e implícitas de seus usuários [2].

Garantir que o software faz o que o cliente deseja – ou validação de software – é um dos quatro processos fundamentais comuns ao desenvolvimento de software [10], e pode ser atingido de maneira mais eficiente a partir de testes de sistemas funcionais, que garantem que o comportamento do sistema está de acordo com a especificação de requisitos [1].

Para se obter uma maior eficiência na atividade de testes, uma grande aposta atual é automatizar o processo de teste manual existente. Desta forma, os testes manuais seriam executados automaticamente por uma ferramenta de automação de testes.

Os benefícios da automação podem parecer óbvios. Os testes podem ser executados mais rapidamente, são consistentes, e os testes podem ser executados repetidamente com menos despesas gerais. Quanto mais testes automatizados são adicionados ao projeto, mais testes podem ser executados. Já os testes manuais ficam focalizados em testes mais rigorosos [12].

Não há dúvidas que testes automatizados podem permitir que as organizações produzam um software de mais alta qualidade ao alavancarem recursos existentes, porém uma ferramenta de automação de testes é apenas parte da resposta.

A princípio, automatizar testes pode parecer fácil: basta comprar uma ferramenta de automação, gravar os seus testes manuais e executá-los quando desejar. Infelizmente, isto não funciona na prática. Da mesma forma que há mais do que uma linguagem de programação para desenvolver software, há mais do que uma ferramenta de automação para automatizar testes [3].

2. Objetivos

A proposta apresentada neste documento é o desenvolvimento de um *framework* para automação de testes funcionais que promova o reuso e a generalização.

Este framework foi inicialmente construído durante a automação de testes de uma empresa de desenvolvimento de software.

Este artigo irá documentar este *framework*, juntamente com as metodologias e boas práticas utilizadas para a sua construção e demonstrar os resultados obtidos na utilização deste em uma empresa desenvolvedora de software.

Este artigo é limitado a descrever uma abordagem para extensão da ferramenta *Rational Functional Tester* para o seu uso otimizado em aplicações Web. Porém, os conceitos aqui apresentados podem ser aplicados a outras ferramentas de automação de testes.

3. Cronograma

O cronograma (Figura 1) apresenta algumas datas para as atividades chaves do processo de desenvolvimento do trabalho de graduação.

	Mês															
	Junho				Julho				Agosto				Setembro			
Atividade	s1	s2	s3	s4	s1	s2	s3	s4	s1	s2	s3	s4	s1	s2	s3	s4
1. Levantamento do escopo do TG																
2.Levantamento bibliográfico																
3. Análise do material bibliográfico																
4.Definição macro dos capítulos do TG																
5.Desenvolvimento da metodologia para automação de testes funcionais																
6.Documentação do framework e do case																
8.Escrita do relatório de graduação (TG)																
9.Preparação para apresentação Oral																

4. Referências

[1] BURNSTEIN, Ilene. Pratical Software Testing, Springer, 2002

[2] DUARTE K. C., FALBO R. A., Uma Ontologia de Qualidade de Software. Disponível em: http://www.inf.ufes.br/~falbo/download/pub/Wqs2000.pdf. Acessado em: 25/06/2006

[3] FEWSTER, M., GRAHAM D., Software Test Automation, ACM Press , 1999

[4] IBM Rational Functional Tester, IBM Corporation. Disponível em: ftp://ftp.software.ibm.com/software/rational/web/datasheets/rft.pdf. Acessado em: 25/06/2006

[5] KANER, Cem. Architectures of Test Automation, 2000. Disponível em: http://www.kaner.com/testarch.html. Acessado em: 25/06/2006

[6] KANER, Cem. Improving the Maintainability of Automated Test Suítes, 1997. Disponível em: http://www.kaner.com/lawst1.htm. Acessado em: 25/06/2006

[7] KELLY, Michael. Framework automation with IBM Rational Functional Tester: Data-driven. Disponível em: http://www-128.ibm.com/developerworks/rational/library/05/1108_kelly/. Acessado em: 25/06/2006

[8] KELLY, Michael. Introduction to IBM Rational Functional Tester 6.1. Disponível em: http://www-128.ibm.com/developerworks/rational/library/04/r-3228/index.html. Acessado em: 25/06/2006

[9] PETTICHORD , Bret. Success with Test Automation, 2001. Disponível em: http://www.io.com/%7Ewazmo/succpap.htm. Acessado em: 25/06/2006

[10] SOMMERVILLE, Ian. Software Engeneering, 1996

[11] SWEBOK®, Guide to the Software Engineering Body of Knowledge. Versão 2004. Disponível em:

http://www.computer.org/portal/cms_docs_ieeecs/ieeecs/education/certification/Swebok_2004.pdf. Acessado em: 25/06/2006

[12] ZALLAR, Kerry. Automated Software Testing - A Perspective. Disponível em: http://www.testingstuff.com/autotest.html. Acessado em: 25/06/2006

[13] ZAMBELICH, Keith. Totally Data-Driven Automated Testing. Disponível em: http://www.sqa-test.com/w_paper1.html. Acessado em: 25/06/2006

Rafael Oliveira Nóbrega
Orientando

Alexandre Marcos Lins de Vasconcelos
Orientador

Recife, 20 de junho de 2006