Área: CV ( ) CHSA ( ) ECET ( )

UMA FERRAMENTA PARA GERAÇÃO AUTOMÁTICA DE RELATÓRIO DE MUDANÇAS

Antonio Gabriel Di Atlanta Valente (bolsista do PIBIC/CNPq), Pedro de Alcântara dos Santos Neto (Orientador, Depto. de Informática e Estatística – UFPI)

Introdução

O usuário de um software necessita ser informado sobre como operá-lo. Existem documentos que podem auxiliar nessa atividade, como a documentação de usuário, que é um material impresso ou eletrônico que provê informações de como realizar determinada tarefa passa a passo (SANTOS et. al., 2010). Outro documento que pode auxiliar no uso de um software é o relatório de mudanças, que relata todas as alterações realizadas em um software.

Atualmente, os softwares são alterados constantemente, seja de maneira corretiva ou no intuito de acrescentar funções. Por conta disso percebemos que é necessário orientar o usuário quanto às mudanças realizadas e seus impactos na realização das atividades. A utilização ou não de relatório de mudanças reflete diretamente na satisfação dos usuários, pois em caso de não uso, dificuldades na realização de tarefas irão aparecer, causando diminuição de produtividade, aumento de tempo gasto para execução das tarefas, e possivelmente desistência do uso.

A geração do relatório de mudanças pode ser bastante complicada quando existem diversas pessoas trabalhando na manutenção de um software. Imagine, por exemplo, termos 10 pessoas alterando telas, criando novas funcionalidades, alterando regras para cálculos. Ao se lançar uma nova versão de um software é necessário saber quais mudanças serão incorporadas à nova versão, visto que algumas podem não ter sido finalizadas ou podem estar feitas mas planejadas para serem incorporar apenas em versões futuras do produto.

Este trabalho propõe uma nova alternativa para a geração do relatório de mudanças, baseada no uso de testes funcionais. Testes funcionais têm como objetivo verificar se o comportamento do software está de acordo com sua especificação. Como eles possuem muitas informações sobre um software é possível reutilizá-los para gerar outros artefatos do desenvolvimento, como por exemplo, o relatório de mudanças.

Metodologia

Este trabalho foi dividido em cinco fases: Fase Teórica, Fase de Análise, Fase de Desenvolvimento e Fase de Finalização. Na Fase Teórica, fizemos um levantamento bibliográfico da área de testes de softwares, da utilização de testes funcionais para geração de artefatos e dos trabalhos relacionados a relatório de mudanças. Na Fase de Análise foi feito um levantamento das principais ferramentas de testes funcionais open-source e os scripts de testes gerados por elas foram analisados para definir como deveria ser feita a extração das informações relevantes para se gerar o relatório de mudanças. Parte dessa fase foi realizada em paralelo com a Fase de Desenvolvimento, em que foi criado o método que permitiu a geração do relatório de mudanças a partir de testes funcionais. Na Fase de Experimentação, realizamos um estudo experimental com intuito de comparar o esforço necessário para criação dos relatórios de mudanças com e sem o uso da ferramenta desenvolvida, que implementa o método proposto neste trabalho. Por fim, na última etapa, a Fase de

Finalização, escrevemos um artigo descrevendo o método desenvolvido e a ferramenta implementada, além do estudo experimental realizado para sua avaliação.

Resultados e Discussão

Como resultado da análise das principais ferramentas de testes funcionais open-source, escolhemos utilizar os scripts de testes funcionais gerado pelas ferramentas Selenium IDE (SELENIUM, 2011) e Canoo WebTest (CANOO, 2011). E a partir da análise desses scripts, definimos um método para utilização das informações extraídas na geração do relatório de mudanças.

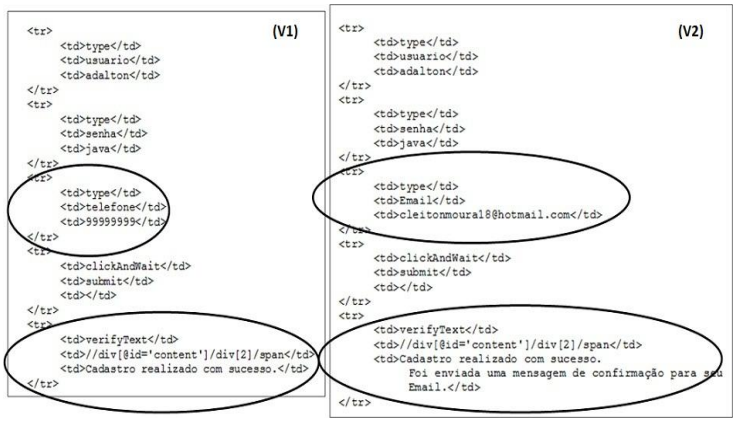

Figura 1: Comparação entre duas versões de um script de teste

As informações extraídas referem-se aos procedimentos e casos de teste. Um caso de teste especifica como deve ser testada uma parte do software. Essa especificação inclui entradas, saídas esperadas e condições sob as quais os testes devem ocorrer. Um procedimento de teste detalha as ações a serem executadas em um determinado caso de teste (IEEE, 1998).

A Figura 1 mostra duas versões de um script de teste criado com Selenium IDE. Se fizermos a análise comparativa entre o script V1 e o script V2, podemos verificar que V2 se diferencia pela adição do campo “Email” e a ausência do campo “telefone”, além da diferença das mensagens de saídas entre as versões. Ou seja, alterações no procedimento de teste podem indicar alterações na interface do usuário, informando campos adicionados e removidos, além de indicar alterações nas regras do negócio. Sendo assim, essas alterações podem ser descritas na forma de um relatório de mudanças.

Tendo posse de um método para utilizar as informações extraídas para geração do relatório de mudanças, necessitamos de um processo para construção da ferramenta que implemente essa geração. Esse processo irá definir as atividades a serem realizadas, as informações de entrada necessárias, artefatos requeridos e produzidos, e procedimentos adotados.

Como indica na Figura 2, o usuário da ferramenta de geração de relatório de mudanças terá que fornecer como entrada os scripts de testes de duas versões diferentes do software, criados pelos desenvolvedores do software utilizando as ferramentas citadas a acima. Além disso, existem informações adicionais que necessitam ser inseridas pelo usuário, que não podem ser extraídas desses scripts.

Após isso, o relatório é construído a partir da comparação das informações extraídas dos scripts de testes funcionais. Por exemplo, temos duas versões do software, V1 (mais antiga) e V2

(mais recente). Se V1 possuir um script de teste para uma determinada funcionalidade e V2 não possuir, sabemos que temos uma funcionalidade que foi removida em V2. Se V1 não possuir um script de teste para uma determinada funcionalidade e V2 possuir, sabemos que temos uma funcionalidade que foi adicionada em V2. Se ambas as versões possuirem script de teste para a mesma funcionalidade, as informações extraídas desses scripts são comparadas, conforme a Figura 1. A partir disso, sabemos quais campos, mensagens, colunas e afins, foram adicionados, removidos ou alterados. Todas essas informações são registradas em um formato de relatório proposta neste trabalho.

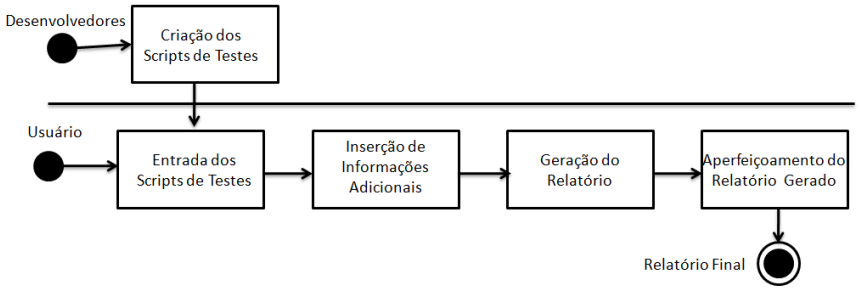

Figura 2: Processo de geração do relatório de mudanças

O relatório de mudanças pode ser alterado pelo responsável da sua geração, realizando ajustes pontuais. Esses ajustes devem ser aplicados, pois o relatório gerado inicialmente não relata todas as alterações da melhor forma para o entendimento pelo usuário final. Sendo assim, esse método possibilita a geração de um relatório final de mudanças a partir de poucas alterações no relatório gerado automaticamente.

Conclusões

Este trabalho apresentou um método para geração semi-automática de relatório de mudanças de software, que usa como insumos scripts de testes funcionais. Este método foi implementado com a criação de uma ferramenta denominada TChangeReport, sendo o trabalho de um projeto PIBITI 2010/11 com mesmo nome.

Com a ferramenta criada, um estudo experimental foi realizado com objetivo de avaliar a TChangeReport, consequentemente o método no qual ela implementa. O estudo mostrou que o uso da ferramenta pode reduzir o esforço para realização da geração do relatório de mudanças. Além do que a qualidade do relatório gerado pela ferramenta foi bastante superior ao trabalho manual. Isso nos sinaliza que abordagem de utilizar os testes como artefatos para construção de outros artefatos é algo promissor.

Referências Bibliográficas

CANOO. Canoo Web Test. Web Test tool. http://webtest.canoo.com/webtest/manual/WebTestHome.html. Último acesso em 08/08/2011.

IEEE Standard for Software Test Documentation – IEEE Std 829-1998. IEEE Computer Society, 1998.

SANTOS I., SANTOS NETO, P., MOURA, R. e CASTELO BRANCO, A. Documentação dirigida por testes, Simpósio Brasileiro de Qualidade de Software (SBQS), 15 pag, 2010.

SELENIUM. Web application testing system. http://seleniumhq.org/. Último acesso em 08/08/2011

Palavras-chave: Teste de Software. Script de Teste. Relatório de Mudanças.