

6 Conclusão

O principal objetivo deste trabalho foi o de estudar como a geração semi-automática de testes a partir da utilização de tabelas de decisão e casos de uso pode auxiliar na redução do tempo gasto na criação e execução dos testes e na descoberta de falhas no comportamento do sistema alvo dos testes.

Para esse fim foi adaptada e aprimorada a ferramenta de edição de tabelas de decisão inicialmente desenvolvida por (Lachtermacher, 2010), para adequar-se à geração de testes funcionais para interfaces web. Também foi criada uma ferramenta para a geração automática de scripts de testes funcionais a partir do arquivo XML gerado pelo editor de tabelas de decisão. Essa ferramenta gera testes cuja execução é automatizada através de JUnit e a automação do navegador e da interação com os elementos da interface web é feita através da ferramenta Selenium (Selenium, 2009). Com essas duas ferramentas é possível gerar semi-automáticamente os testes automatizados para interfaces web. Para apoiar a elicitação e geração dos testes, propôs-se uma estrutura de descrição de casos de uso que pudesse servir de especificação para os testes (Díaz et al, 2004) (Heumann, 2001)(Staa, 2010a). A avaliação da eficácia do procedimento e das ferramentas deu-se através da aplicação destes na geração de testes funcionais para um sistema real e através de comparação com técnicas manuais e técnicas tradicionais de geração de testes automatizados aplicadas ao mesmo sistema.

De acordo com os resultados obtidos, os testes gerados semi-automáticamente com o auxílio de tabelas de decisão apresentaram as seguintes vantagens principais:

- Encontraram erros que os testes já existentes, gerados com Squish for Web a partir de codificação manual, não encontraram;
- Obtiveram maior cobertura de condições, ou seja, casos de teste, do que os testes já existentes gerados com o Squish for Web a partir de codificação manual.

- Redução do tempo gasto para gerar testes em comparação com a geração manual.
- Redução no tempo necessário para gerar os testes em comparação com a abordagem “*capture and replay*” utilizando-se Selenium IDE.
- Redução no tempo necessário para gerar os testes em comparação com a abordagem DDC a partir da utilização de JBehave.

Como desvantagens, temos que testes em telas complexas, com muitos elementos, podem gerar uma tabela de decisão com dezenas, senão centenas, de regras, dificultando sua criação e preenchimento.

Assim, acreditamos, perante os resultados observados, que a abordagem apresentada é uma boa alternativa às técnicas mais utilizadas para a geração de testes funcionais para sistemas web, conseguindo uma boa cobertura de condições e em um tempo menor que as demais técnicas testadas, principalmente para telas com formulários demasiados grandes, onde a visualização de todas as combinações de testes pode tornar-se difícil.

Por fim, trabalhos futuros sobre a ferramenta de edição das tabelas de decisão, como suporte a novas funcionalidades auxiliares para a edição, podem amenizar algumas dificuldades encontradas nesta abordagem e diminuir ainda mais o esforço necessário para a geração da massa de testes.