

Projeto 6.18

Automação de Testes Sistêmicos Funcionais

Paula Luciana F. Cunha, Rosanne M. R. Carneiro, Carlo Giovano S. Pires, Liane R. P. Bandeira, Paula M. Donegan, Camila Maia, Ana Cristina Matos

1. Objetivos e Justificativa

No âmbito de desenvolvimento de software, as organizações têm buscado qualidade dos produtos que constroem, a fim de satisfazer clientes e se manterem competitivas no mercado. Grandes esforços têm sido empenhados em melhorar as atividades de teste, uma vez que estas avaliam a qualidade do produto final.

Em virtude da crescente complexidade das regras de negócios que regem as aplicações e as novas tecnologias, o esforço necessário para testar aplicações tornou-se maior. A multiplicidade de caminhos a serem percorridos e a diversidade de combinações de dados de entrada e de saída de uma aplicação inviabiliza a execução manual dos testes. Não obstante, testes de desempenho (performance, carga, volume) são exemplos de testes difíceis de serem realizados sem o auxílio de uma ferramenta de automação. Enquanto estes aspectos - complexidade, diversidade de caminhos, escassez de recursos e tempo, por exemplo - têm contribuído para que a prática de testar software seja pouco disseminada nas organizações, uma metodologia bem concebida, um planejamento adequado e o uso de ferramentas podem melhorar a eficácia dos testes de software.

Este projeto objetiva difundir a automação de testes, com o estabelecimento de uma estratégia para automatizar os testes sistêmicos funcionais, controlando seus riscos e aumentando o reuso dos scripts gerados. A automação de testes, considerada o principal recurso para melhorar a eficiência de um processo de testes [Rios, 2003], consiste em utilizar-se de uma ferramenta para o acompanhamento do processo desde o planejamento, passando pela execução e chegando ao controle dos testes.

2. Metodologia de Execução

Esse trabalho foi elaborado no Instituto Atlântico, situado em Fortaleza-Ceará, uma organização que desenvolve e difunde tecnologias de alto valor agregado para os seus clientes. Seus valores são baseados em inovação e melhoria, esta última refletida principalmente no aperfeiçoamento da qualidade de seu processo de software, visando aumento de produtividade, cumprimento de prazos estabelecidos, redução de custos e, sobretudo, atendimento satisfatório às necessidades de seus

clientes. A organização foi avaliada oficialmente como CMMi nível 3 em fevereiro de 2006.

Ao longo do ano de 2005, foi desenvolvida uma metodologia para automação de testes sistêmicos funcionais, englobando as fases “Definição de critérios para automação”, “Seleção de ferramenta”, “Planejamento dos testes”, “Elaboração de scripts”, “Execução e gerenciamento de testes automatizados” e “Avaliação e melhoria do processo”.

Durante a fase “Definição de critérios de automação”, foi identificado que os projetos que realizam automação de testes são comumente de médio e grande porte, implementam o ciclo de vida iterativo-evolutivo e possuem analistas de testes dedicados ao projeto. Os critérios para seleção dos casos de uso a serem automatizados levam em consideração a quantidade de ciclos de testes a que será submetido, o nível de criticidade para o negócio do cliente – sua complexidade, bem como a quantidade de dependências com outros casos de uso desenvolvidos.

Na fase “Seleção de ferramenta”, detectou-se a necessidade de aquisição de ferramentas para apoiar o processo de testes de regressão¹.

Assim, a instituição adquiriu ferramentas da *IBM Rational Corporation*® [Rational, 2005] (*TestManager*®, para planejamento e execução testes e geração de relatórios; *Rational Robot*® e *Rational Funcional Tester*®, para gravação e manutenção de scripts de testes) e utiliza-se de uma ferramenta para registro e acompanhamento de problemas. Além de tais investimentos, diversos cursos foram ministrados a fim de capacitar os colaboradores nas mesmas [Bandeira, 2005].

A fase “Planejamento dos testes” é de extrema importância, pois, a partir dela, serão traçadas as diretrizes para o processo de automação. Antes de iniciar a automação, elaboram-se as especificações de teste com seus respectivos Casos de Teste [Myers, 2004], que correspondem ao projeto dos testes funcionais necessários. O projeto é composto de procedimentos de execução, ações de verificação e idéias de dados de teste. As atividades de automação correspondem à implementação dos casos de teste anteriormente citados. Finalizando as especificações de testes, realiza-se a estruturação dos casos de teste na ferramenta de planejamento, agrupando-os por funcionalidades, cenários e dados válidos e inválidos, respectivamente, seguindo uma padronização de nomenclaturas. Em seguida, analisam-se quais devem ser executados automaticamente, levando em consideração os riscos envolvidos, a

¹ 1 Teste de Regressão: Após a correção de um erro, é realizada a repetição de todos os testes, para garantir que a mesma não impactou negativamente na aplicação [Rakitin, 2001].

priorização dos requisitos do cliente e a estabilidade dos casos de uso. O resultado dessa análise é documentado no plano de teste do projeto.

Paralelamente, devem-se desenvolver dados de teste para validar as regras da aplicação, antes da geração dos scripts. A organização dos dados de teste é baseada na técnica Data-Driven [Mosley, 2003], na qual os mesmos são separados dos scripts, como forma de reduzir o esforço necessário para manutenção. Ademais, com o intuito de diminuir o retrabalho, estabiliza-se a nomeação de campos e demais elementos da interface ao longo do desenvolvimento, uma vez que eles, posteriormente, direcionam o preenchimento dos campos de entrada na execução dos scripts.

Na fase “Elaboração de scripts”, gera-se um script para cada caso de teste a ser automatizado, garantindo assim a independência de scripts.

Isso faz com que uma falha de um deles não interfira na execução de outros. Além disso, os scripts automatizados devem ter o ponto de partida em comum e serem autocontidos. Os scripts desenvolvidos contêm as seguintes ações: navegação da interface, funções específicas de negócio e pontos de verificação. A geração de scripts pode ser realizada de duas formas. Na primeira delas, utiliza-se o protótipo da aplicação, deixando a gravação de pontos de verificação para o final da fase de implementação.

Embora a geração sobre o protótipo ocasione uma antecipação da gravação, visto que os scripts podem ser iniciados concorrentemente à fase de implementação, constatou-se que há uma grande manutenção nos scripts, em virtude de haver instabilidade da aplicação e mudanças na interface. A segunda forma de gravação, mais utilizada, diz que a automação ocorre somente após a execução manual de ciclo de testes, pois a aplicação apresenta-se mais estável, depois de ter passado por bateria manual de teste, e assim o esforço de gravação dos scripts é diminuído. Após a gravação dos scripts, eles são executados novamente, a fim de assegurar sua corretude. Caso seja detectada alguma falha, são efetuadas as devidas correções até que cumpram adequadamente o propósito do teste. Concluída a geração dos scripts, os mesmos são associados aos casos de teste estruturados na ferramenta de planejamento, anteriormente citada.

Quando da fase “Execução e gerenciamento de testes automatizados”, agrupam-se os scripts de teste por caso de uso, após sua geração, visando organizar e agilizar o processo de execução, além de suportar de forma direta o mecanismo de rastreabilidade com requisitos. Esse agrupamento, denominado suíte, facilita o controle do acompanhamento dos testes, uma vez que fornece status dos casos de teste, possibilitando assim uma melhor visibilidade de cada caso de uso. A cada nova execução de uma suíte de teste, faz-se necessário restaurar a

configuração inicial do ambiente de automação de teste, para garantir a consistência dos resultados esperados. Uma falha detectada durante a execução dos testes é cadastrada na ferramenta de registro de problemas, sendo vinculada ao seu caso de teste, a uma baseline de teste e classificada mediante sua severidade. Além disso, os passos utilizados para reprodução da falha são descritos detalhadamente com o intuito de facilitar sua correção pela equipe de desenvolvimento. Após a execução da suíte, a ferramenta fornece o resultado de cada script que a compõe. Esse resultado pode ser salvo para posterior geração de relatórios. A ferramenta de planejamento permite gerar vários relatórios, quer sejam gráficos ou textuais, podendo também ser customizados pela versão do build, plano de teste e status dos casos de teste (implementados, falhos ou com sucesso).

Caso existam mudanças na aplicação, deve-se realizar manutenção nas suítes de testes afetadas, para prosseguirem válidas em testes posteriores.

Periodicamente são realizadas reuniões dos analistas de testes da Instituição, a fim de realizar a “Avaliação e melhoria do processo”. Durante estes encontros, discutem-se melhores práticas, lições aprendidas, incorporando melhorias ao processo quando pertinentes e necessárias.

3. Resultados Relevantes

- E3 – Métodos e/ou algoritmos desenvolvidos

Desenvolvimento da Metodologia para Automação de Testes Sistemáticos Funcionais, descrita neste artigo.

- E4 – Artigos publicados

Donegan, P., Bandeira, L., Matos, Ana, Cunha, P., Maia, C. & Pires, G. (2005). Aplicabilidade da Automação de Testes Funcionais – A Experiência no Instituto Atlântico. Simpósio Brasileiro de Qualidade de Software (pp. 447 - 454). Porto Alegre: Pontifícia Universidade Católica do Rio Grande do Sul.

- E5 – Recursos humanos capacitados (especialistas, mestres, doutores, etc)

Treinamento na ferramenta *Rational Functional Tester®*.

- E6 – Dissertações e/ou teses geradas

Monografia de Conclusão de Curso - BANDEIRA, Liane R. P.. “Qualidade do Produto a partir de Automação de Testes: Um Estudo de Caso”.

Monografia de conclusão de curso de Ciência da Computação, Universidade Estadual do Ceará, 2005.

4. Características Inovadoras

O Instituto Atlântico possui um Processo de Testes formal baseado nas melhores práticas descritas pelo CMMi – *Capability Maturity Model Integration* e RUP – *Rational Unified Process*. Em 2005, realizou-se um trabalho de melhoria do processo de testes em conformidade com o CMMi nível 3. Com o amadurecimento das metodologias de desenvolvimento adotadas pelos projetos, as atividades de teste passaram a ter grande relevância para o Instituto, sendo importante a existência de um grupo de teste. Esse grupo é responsável por planejar, executar e acompanhar testes sistêmicos de projetos da organização.

5. Conclusão e Perspectivas Futuras

A metodologia descrita tenciona servir como guia para que outras organizações desenvolvam seus processos de testes, utilizando-se do benefício da automação de testes sistêmicos. No Instituto Atlântico, a automação de testes sistêmicos funcionais gerou bons resultados para a organização. Verificou-se a garantia da qualidade dos testes, pois eram repetidos sob os mesmos dados de teste e procedimentos de execução durante os ciclos de teste, sendo esse aspecto muito relevante para testes de regressão.

Levando em consideração a filosofia institucional de melhoria contínua dos processos, pretende-se evoluir constantemente com a metodologia acima descrita, a fim de que a organização possua um processo cada vez mais aderente aos padrões internacionais de qualidade, tais como o CMMi – *Capability Maturity Model Integration*.

6. Referências Bibliográficas

[Bandeira, 2005] BANDEIRA, Liane R. P.. “Qualidade do Produto a partir de Automação de Testes: Um Estudo de Caso”.

Monografia de conclusão de curso de Ciência da Computação, Universidade Estadual do Ceará, 2005.

[Matos, 2005] MATOS, Ana C.; PIRES, Giovano; BANDEIRA, Liane; FERREIRA, Paula; DONEGAN, Paula. “Uma Definição de Processo de Teste para SW-CMM Nível 3”. In: Anais SUCESU – Congresso Nacional de Tecnologia de Informação e Comunicação, 2005.

[Mosley, 2003] MOSLEY, Daniel J; POSEY, Bruce A.. “Just Enough Software Test Automation”. – Just Enough Series/Yourdon Press, 2003.

[Myers, 2004] MYERS, Glenford J.. “The Art of Software Testing”. – John Wiley & Sons, 2nd. Edition, 2004.

[Rakitin, 2001] RAKITIN, Steven R.. “Software Verification and Validation: A Practitioner's Guide”. – Artech Computer Science Library, 2001.

[Rational, 2005] IBM Rational Corporation. Acessado em: 08/03/2005.
Disponível em: <http://www-306.ibm.com/software/rational/>.

[Rios, 2003] RIOS, Emerson; MOREIRA, Trayahú R. F.. "Projeto & Engenharia de Software - Testes de Software". – Ed. Atlas Book, Rio de Janeiro, 2003.

[RUP, 2002] Rational Unified Process®, RUP® – Versão 2002.05.00.