Aplicação de Dados Históricos para Seleção de Casos de Teste de Regressão

Camila Socolowski, Mario Jino (Orientador), Marcelo Fantinato (Co-orientador)

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, CEP 13083-970 – Campinas, SP, Brasil

{camila.socolowski@gmail.com, jino@dca.fee.unicamp.br, m.fantinato@usp.br}

Abstract – Regression testing is applied to ensure the software quality across its multiple versions. To minimize the cost of this activity, test cases with higher capability of detecting faults should be selected, keeping the efficiency of the test suite resulting from the selection. In this paper, we propose an approach in which historical data are used as a selection technique in regression testing. The approach feasibility and usefulness are validated through its application to test a real world software. In the experiment, the proposed approach is compared with other two black box testing selection techniques, in terms of both the efficiency in the fault detection and the software reliability achieved with their application.

Keywords – regression testing, test case selection techniques, historical data, software reliability.

1. Introdução

Após seu desenvolvimento, um produto de software usado na indústria precisa evoluir devido à inclusão de novas funcionalidades requeridas pelo cliente à medida que seus negócios mudam. Além disso, defeitos são encontrados durante o uso do software em produção, provocando a necessidade de sua manutenção. Diante dessa evolução e manutenção, manter a qualidade do software após diversas versões é um desafio. Algumas vezes, a qualidade se deteriora devido às alterações realizadas [1].

O teste de regressão é usado para garantir a qualidade do software após diversas versões terem sido geradas. No entanto, ele é muito custoso, por requerer muitas execuções de Casos de Teste (CTs), cuja quantidade aumenta consideravelmente conforme o software evolui [2]. Esse alto custo leva à aplicação de técnicas de Seleção de Testes de Regressão (STR), que direcionam a seleção de apenas um subconjunto de CTs para re-execução. Em geral, a seleção ótima de testes, ou seja, aquela que seleciona exatamente os CTs que revelam todos os defeitos existentes, é impossível. Portanto, a análise de custo-benefício aplicada às diversas técnicas STR é um interesse central da pesquisa e da prática em teste de regressão [3].

Em geral, as técnicas STR são estudadas a partir de versões-base criadas ou identificadas de um sistema que acompanham os conjuntos de teste. Então, alguns algoritmos STR são executados e comparados em termos de tamanho e eficiência com o conjunto de testes original. No entanto, esses estudos empíricos consideram o teste de regressão em uma única sessão de testes, sem considerar restrições de tempo e de recursos do mundo real. De acordo com Kim e Porter [4], uma forma mais proveitosa de estudar as técnicas STR seria por meio de sessões de teste sujeitas a restrições que existem na prática. Além disso, dados históricos de desempenho dos CTs podem ser usados para melhorar o desempenho do teste de regressão a longo prazo, uma vez que as técnicas STR existentes são *memoryless*, ou seja, consideram apenas as informações obtidas a partir das versões atuais ou das imediatamente precedentes do software em teste.

Park *et al.*[5] usam dados históricos para uma tarefa similar à realizada pelas técnicas STR: priorizar CTs, definindo uma ordem em que todos os CTs do conjunto devem ser re-executados. Neste artigo, uma abordagem é proposta em que a técnica de priorização de Park chamada aqui de *Retest Using Historical Data*, é aplicada como uma técnica STR. Essa abordagem é comparada com duas outras técnicas STR, em termos tanto da eficiência na detecção de defeitos quanto da confiabilidade de software atingida com sua aplicação. A comparação é realizada por meio de um experimento em que os dados históricos são coletados em função do teste de um software do mundo real.

2. Proposta

Esse trabalho apresenta o projeto de mestrado que propõe o estudo e a implementação de uma

técnica de seleção de casos de teste caixa preta baseada em dados históricos, coletados de aplicações do mundo real. Um experimento será conduzido para validar essa técnica, comparando-a com outras técnicas de STR "caixa preta", como o Reteste de Casos de Uso de Maior Risco [6] e o Reteste por Perfil Operacional [7].

Por fim, será especificada uma ferramenta para registro de casos de teste e de defeitos encontrados que facilite o armazenamento e a recuperação dos dados históricos na execução de baterias de teste e que calcule automaticamente os valores históricos das execuções de teste para uso posterior nos testes de regressão. A especificação dessa ferramenta será composta pelos requisitos necessários para a construção da ferramenta, diagramas da UML (do inglês, *Unified Modelling Language*), alguns protótipos da interface gráfica e principais algoritmos utilizados.

2.1 Trabalhos relacionados

Outros trabalhos abordam técnicas para teste de regressão englobando seleção de testes de regressão, priorização de testes de regressão e redução da suíte de teste.

No trabalho de Graves et al. [8] é apresentado um experimento que demonstra os custos e benefícios de diversas técnicas de teste de regressão enfatizando a redução da suíte de teste e a detecção de defeitos. As técnicas consideradas pelos autores são as de minimização da suíte de testes, segura e fluxo de dados. Os autores consideram que as técnicas segura e de fluxo de dados são eficientes em detectar falhas, mas selecionam números amplamente variados de casos de teste. Em um ambiente restrito, tal abordagem pode ser simplesmente inviável [8].

Os trabalhos de Wong e Rothermel [2, 3] baseiam-se nas técnicas de priorização sem memória e modelam o teste de regressão como uma atividade de uma única vez, ignorando possíveis efeitos sobre múltiplas releases de software. Finalmente, não levaram em consideração as restrições de tempo e as restrições de recursos [9].

Kim e Porter [4] apresentam resultados iniciais de um estudo empírico sobre o uso de dados históricos de execução de teste para priorizar a seleção de casos de teste em um processo restrito de teste de regressão. Os resultados experimentais relatados pelos autores apóiam fortemente o princípio de que o teste de regressão pode ter que ser feito de forma diferente em ambientes restritos e não restritos. Os autores também apóiam que a informação histórica pode ser útil em reduzir custos e aumentar a eficiência de processos de teste de regressão de longa duração.

Park et al. [5] conduzem um experimento para validar e provar a eficiência de sua abordagem baseada em valor histórico para priorização de casos de teste que considera o fator custo. No experimento realizado foi usada a métrica APFDc (*Average Percentage of Faults Detected per cost*) para comparar a abordagem em questão com outra técnica de priorização de casos de teste baseada em cobertura. Como resultado do experimento foi comprovado que a abordagem que considera o fator custo produz melhores resultados, em termos de APFDc, do que as técnicas de priorização de casos de teste baseadas em cobertura.

Uma análise mais específica dos trabalhos mencionados nesta seção mostrou que na área de testes a maior parte dos trabalhos relativos a regressão baseiam-se na aplicação de técnicas que dependem da análise de código fonte do software em teste. Isso pode acarretar dificuldades para selecionar estratégias de regressão em situações em que não é possível ter acesso ao código fonte para análise.

Outra questão importante mencionada por Kim e Porter [4] e Park et al. [5] é a necessidade de realizar experimentos para abranger uma variedade mais ampla de defeitos que ocorrem naturalmente em sistemas reais com restrições de custos, tempo e recursos e comparar os resultados com outros métodos não baseados em histórico descritos na literatura.

2.2. Experimento

A abordagem proposta foi aplicada em um software para Web do mundo real, desenvolvido para o Serviço da Receita Federal Brasileira (SRFB). Desse software, foram levados em conta apenas CTs caixa preta, ou seja, apenas aspectos funcionais foram considerados para a coleta de dados e para a aplicação da abordagem proposta. Por ser um software do mundo real, os defeitos encontrados no experimento são reais.

Para fins de comparação de resultados com a técnica *Retest using Historical Data*, ambas as técnicas Reteste de Casos de Uso de Maior Risco e Reteste por Perfil Operacional foram aplicadas no experimento.

Para o experimento, foram considerados inicialmente uma média 40 CTs (casos de teste)

executados em cinco baterias de teste. Na prática, o número de CTs variou entre as baterias, pois alguns CTs foram incluídos e outros, à medida que se tornaram obsoletos, foram excluídos de uma bateria para outra.

Para medir a eficiência das técnicas nas cinco baterias, o conjunto original de testes foi inteiramente executado seguindo a técnica *Retest All* para que seus resultados fossem usados como referência. O alto custo da criação desses resultados de referência limitou o tamanho inicial a 40 CTs.

Para coletar dados do experimento, um protótipo de software foi criado para:

1. Calcular o valor histórico a partir do custo dos CTs, do total de severidade dos defeitos e dos dados históricos armazenados;
2. Armazenar os dados históricos em um repositório de dados;
3. Selecionar os CTs com maior valor histórico, com base em um critério de corte definido pelos engenheiros de software;
4. Armazenar as informações obtidas a partir da aplicação das técnicas Reteste de Casos de Uso de Maior Risco e Reteste por Perfil Operacional usadas neste artigo;
5. Calcular a confiabilidade do programa após a execução de cada bateria de teste, para cada técnica abordada neste artigo.

A partir da execução dos CTs os custos derivados a partir dos tempos de execução de cada CT e as severidades das falhas de defeitos derivadas a partir das criticidades dos defeitos detectados pelos CTs foram armazenados em um repositório de dados históricos. Essas informações foram então usadas para o cálculo do valor histórico, calibrando a técnica *Retest using Historical Data* a cada bateria de teste.

Finalmente, os subconjuntos de teste selecionados para cada técnica puderam ser comparados em termos de capacidade de detecção de defeitos e em termos da confiabilidade alcançada para o software. Alguns resultados dessa comparação são apresentados a seguir.

3. Resultados

Entendemos como eficiência de um conjunto de testes T' selecionado a partir da aplicação de uma técnica STR sobre o conjunto de testes original T, a capacidade de detectar o maior número de defeitos com o menor número de CTs possível. A partir disso, as técnicas Reteste de Casos de Uso de Maior Risco (RUcR) e Reteste por Perfil Operacional (RpO) foram comparadas com a técnica Retest Using Historical Data (RUhD), em termos do total de defeitos detectados, conforme ilustra a Tabela 1, após as baterias de teste. A diferença entre os totais de defeitos detectados ao longo das baterias mostra que a técnica Retest Using Historical Data apresentou os melhores resultados. Por meio dessa técnica, um número maior de defeitos foi detectado, com o mesmo número de CTs.

Tabela 1. Defeitos por criticidade

Técnicas	*Def. CRI*	*Def. ALT*	*Def. MED*	*Def. BAI*
RUcR	9	8	5	14
RpO	9	8	8	14
RUhD	14	12	22	24

A partir dos dados das Tabelas 2 e 3, pode-se concluir que a capacidade de detecção de defeitos da técnica Retest Using Historical Data foi, para esse experimento, praticamente igual à da técnica de referência Retest All considerando cada uma das baterias de teste. Isto pode ser considerado uma grande evidência de sua eficiência.

Tabela 2. Aplicação da técnica Retest All

Bateria	*CT selecionados*	*Total CT*	*Def. Detectados*
0	35	35	68
1	39	39	35
2	41	41	22
3	43	43	14
4	44	33	4

Tabela 3. Aplicação da técnica RUhD

Bateria	*CT selecionados*	*Total CT*	*Def. Detectados*
0	-	35	68
1	19	39	34
2	12	41	21
3	8	43	13
4	5	33	4

Além disso, as técnicas foram comparadas também em termos da confiabilidade de software alcançada após cada bateria de teste. A confiabilidade do software do SRFB foi medida após cada bateria de teste, sendo que, na bateria 0, as três técnicas tiveram o mesmo percentual de confiabilidade, para uma comparação justa entre as técnicas.

Tabela 4. Percentuais de confiabilidade

Técnicas	*Bt. 0*	*Bt. 1*	*Bt.2*	*Bt.3*	*Bt.4*
RUcR	31,54	52,62	75,07	90,35	90,77
RpO	31,54	52,87	75,32	90,58	90,77
RUhD	31,54	56,46	78.34	90,58	91,09

4. Conclusões

Este trabalho apresenta uma abordagem em que a técnica *Retest Using Historical Data* originalmente proposta para a priorização de casos de teste de regressão foi usada para a seleção de casos de teste de regressão. Por meio de um experimento prático, executando testes em um software do mundo real, a eficiência e a confiabilidade dessa abordagem foi avaliada, por meio de uma comparação com outras duas técnicas STR caixa preta. Como resultado do experimento, a técnica *Retest Using Historical Data* mostrou-se mais eficiente na detecção de defeitos e produziu uma melhora maior na confiabilidade do software do SRFB, quando comparada com as técnicas Reteste de Casos de Uso de Maior Risco e Reteste por Perfil Operacional.

Alguns trabalhos futuros estão previstos para aperfeiçoar a abordagem proposta neste artigo: i) novos experimentos com mais dados devem ser realizados, visto que os 40 CTs usados no experimento atual constituem um número relativamente baixo; ii) a técnica Retest Using Historical Data poderá ser comparada com um número maior de técnicas STR; iii) pretende-se elaborar a especificação de uma ferramenta para registro de CTs e de defeitos encontrados que facilite o armazenamento de dados históricos e calcule automaticamente os valores históricos das execuções de teste para uso posterior nos testes de regressão.

Referências

[1] S. Elbaum, A. Malishevsky, and G. Rothermel. Test case prioritization: A family of empirical studies. IEEE Transactions on Software Engineering, 28(2):159–182, February 2002.

[2] G. Rothermel and M. J. Harrold. Analyzing regression test selection techniques. IEEE Transactions on Software Engineering, 22(8):529– 551, August 1996.

[3] G. Rothermel and M. J. Harrold. A safe, efficient regression test selection technique. ACM Transactions on Software Engineering. Methodology, 6(2):173–210, April 1997.

[4] J.-M. Kim and A. Porter. A history-based test prioritization technique for regression testing in resource constrained environments. In ICSE ’02: Proceedings of the 24th International Conference on Software Engineering, pages 119–129, New York, NY, USA, 2002. ACM.

[5] H. Park, H. Ryu, and J. Baik. Historical value-based approach for cost-cognizant test case prioritization to improve the effectiveness of regression testing. SSIRI, 0:39–46, 2008.

[6] R. V. Binder. Testing Object-Oriented Systems: Models, Patterns and Tools. Addison-Wesley Longman, 2000.

[7] J. D. Musa. Software-reliability-engineered testing practice (tutorial). In ICSE ’97: Proceedings of the 19th International Conference on Software Engineering, pages 628–629, New York, NY, USA, 1997. ACM.

[8] T. L. Graves, M. J. Harrold, J.-M. Kim, A. Porter and G. Rothermel. An empirical study of regression test selection techniques. ACM Transactions on Software Engineering Methodology, pages 184–208, New York, NY, USA, 2001. ACM.

[9] W. E. Wong, J.R. Horgan, S. London and H. A. Bellcore. A Study of Effective Regression Testing in Practice. In ICSE ’97: ISSRE '97: Proceedings of the Eighth International Symposium on Software Reliability Engineering, pages 264-273, Washington, DC, USA, 1997. IEEE Computer Society.