Geração de casos de teste aleatória e baseada em modelo para aplicações Android

Henrique Neves da Silva
hen123neves@gmail.com
Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Paraná, Brasil

André Takeshi Endo
andreendo@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Paraná, Brasil

Com o crescente uso de dispositivos móveis e de suas aplicações, pesquisadores têm investigado formas para melhor testar o software, para que se possa garantir o comportamento esperado durante seu uso. Diferentes técnicas para testar as aplicações móveis e gerar entradas de teste automáticas estão presentes na literatura. Neste trabalho serão abordados duas delas, o Teste Baseado em Modelo juntamente com a ferramenta FourMA e o Teste Aleatório com auxílio da ferramenta de estresse Monkey.

PALAVRAS-CHAVE: Teste de aplicações móveis. Teste automatizado. Teste baseado em modelo. FourMA. Teste aleatório. Monkey.

INTRODUÇÃO

Nos últimos anos houve um incrível aumento de aplicações móveis. De acordo com relatórios atuais, **Google Play** o mais famoso mercado de *app* para a plataforma Android, oferece milhões de aplicações. A onipresença dos dispositivos móveis fez com que as aplicações fossem cada vez mais visadas e melhor testadas, antes da disponibilização no mercado [Zhauniarovitch et al. 2015].

Aplicações móveis como qualquer outro software, devem ser testadas adequadamente para que se tenha confiança que a *app* se comportará da maneira esperada. O aumento na utilização de diferentes aplicações implica na demanda por ferramentas que possam testar automaticamente as aplicações móveis [Choudhary et al. 2015].

Uma interessante forma de se testar as aplicações móveis é com a utilização da abordagem de Teste Baseado em Modelo (TBM). Nela cria-se um modelo de software, com auxílio de uma ferramenta como a FourMA (https://github.com/guilhermefarto/MBTS4MA), que representa o comportamento esperado de uma série de eventos dispostos pela aplicação móvel para a interação com o usuário [Farto and Endo 2017].

Além do TBM, é possível utilizar a técnica de Teste Aleatório, que por sua vez, considera a inferface de cada componente do software e faz uso delas para elaborar os cenários de teste. A seleção dos casos de teste é feita de forma aleatória de acordo com o domínio de entrada [Gao et. al 2003]. Em um contexto de teste de aplicações móveis, a técnica consiste em gerar entradas de *UI events* (eventos de usuários) e mediar as respostas da aplicação. A ferramenta utilizada no estudo foi a ferramenta Monkey (https://developer.android.cm/studio/test/monkey.html), pois já pertence ao conjunto de desenvolvimento Android e não requisita nenhuma instalação adicional.

Desta forma o presente estudo trabalha com duas abordagens de teste (Teste Baseado em Modelo e Teste Aleatório) e evidencia os resultados que se pode ter as aplicando em um contexto de aplicações Android.

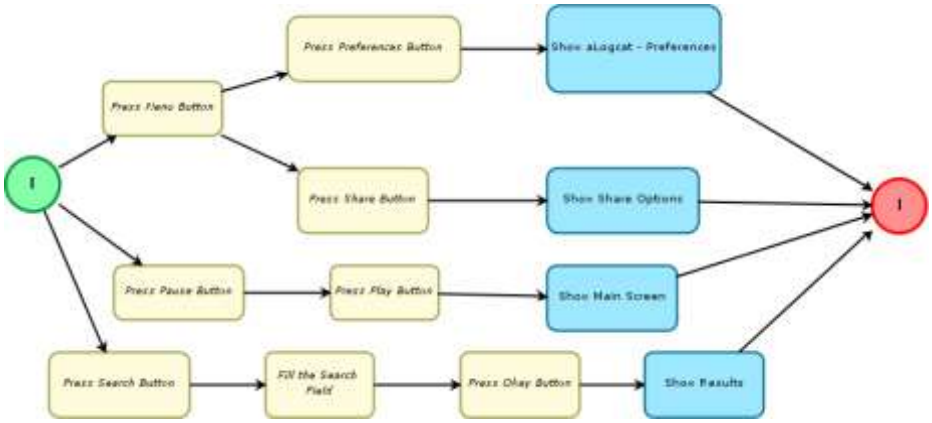
MÉTODOS

Para a execução do experimento, se utilizou como objeto de estudo três diferentes aplicações móveis *open source* disponíveis no **Google Play** (*arXiv mobile*, *aLogcat* e *whohasmystuff*).

Cada abordagem de teste exige diferentes ações. A técnica TBM se sumariza pelas etapas: 1) Modelagem da aplicação; 2) Geração dos casos de teste; 3) Concretização dos eventos com a plataforma *Robotium* na linguagem Java; 4) Execução dos testes e 5) Análise dos resultados (cobertura de código). Na etapa 1) gera-se um modelo com a ferramenta FourMA (Figura 1) com o uso da técnica *Event Sequence Graph* (ESG): utilizada para modelar interações de eventos do software, onde cada nó representa um evento e as arestas sequências válidas entre esses eventos. Etapa 2) cria-se casos de teste de acordo com as sequências lineares de eventos presente no modelo, etapa 3) transforma os casos de teste abstratos em teste executáveis no software a

ser testado, duas classes são criadas, uma contendo as sequências do modelo que se dá pelo início do Nó "[" finalizando no Nó "]" e a outra contém os métodos que cada Nó da sequência representa na aplicação (clique na tela, preenchimento de informações em campos de formulário, etc.).

Figura 1 – Modelo ESG para a aplicação *aLogcat*



Fonte: Elaborado pelo autor

O processo de experimento seguindo a abordagem TA foi realizado a partir do ambiente de trabalho utilizado na pesquisa de Choudhary et al. [2015]. A abordagem consistia em adicionar uma nova aplicação no ambiente e executar o *script* "run_monkey.sh", reponsável por iniciar a ferramenta de estresse Monkey e identitificar quais linhas de código foram cobertas ou não ao longo da execução do teste. O teste de estresse é realizado disparando 1.000.000 *UI events* na interface da aplicação onde se tem 200 milissegundos de espera entre cada evento, o tempo limite para cara teste era de uma hora. Para cada aplicação móvel testada, o *script* cria uma pasta e insere os arquivos de cobertura.

RESULTADOS

Após o estudo, obteve-se a cobertura correspondente às Classes, Métodos e Linhas de cada aplicação móvel testada (Tabela 1).

Tabela 1 – Cobertura de código atingida pelos testes TA e TBM.

Aplicações	Classes	Métodos	Linhas
aLogcat	TA: 72% (28/39) TBM: 68% (24/35)	TA: 55% (150/214) TBM: 67% (117/174)	TA: 62% (561/901) TBM: 60% (512/843)
arXiv mobile	TA: 58% (38/65) TBM: 53% (34/64)	TA: 53% (166/312) TBM: 46% (99/214)	TA: 43% (877/2057) TBM: 39% (826/2093)
whohasmystuff	TA: 93% (25/27) TBM: 68% (19/24)	TA: 76% (117/153) TBM: 48% (57/119)	TA: 67% (488/729) TBM: 43% (345/806)

Fonte: Elabora pelo autor.

DISCUSSÃO

Ao longo dos testes, notou-se que a abordagem TA mostrou uma cobertura de código superior em relação a técnica TBM, entretanto, esta informação não é suficiente para rotular uma técnica como superior. Alguns aspectos precisam ser considerados, na técnica TBM por exemplo a forma com que o modelo ESG é construído pelo testador implica diretamente na quantidade de código que o teste será capaz de cobrir. Desta forma, um modelo mais completo e complexo produzirá mais sequências (casos de testes).

É necessário realçar que com a técnica TBM é possível diminuir a entrada de eventos redudantes, obter uma maior cobertura e possuir um oráculo de teste bem definido. Diferente da técnica TA, que não se tem um teste de oráculo específico, ou seja, não há como saber se o resultado gerado a partir do caso de teste é o resultado esperado. Mesmo TBM ter demonstrado uma melhoria nos resultados com um modelo ESG mais bem elaborado é incerto dizer que ele prevê todos os estados da aplicação.

CONSIDERAÇÕES FINAIS

Neste artigo, apresentou-se um estudo comparativo entre as abordagens TBM e TA em três diferentes aplicações móveis. Durante o estudo foram levantados diferentes aspectos que influenciam na qualidade do teste e ainda mostrou-se que é possível obter resultados interessantes fazendo uso da técnica TA com a ferramenta Monkey, mesmo a abordagem não possuindo critério de parada e gerando diversas entradas redudantes. Além disso, investigou-se a adoção da técnica TBM com modelagem ESG para o contexto de aplicações móveis.

Random and model-based test case generation for Android applications

ABSTRACT

With the increasing use of mobile devices and their applications, researchers have investigated ways to better test the software so that it can guarantee the expected behavior during its use. Different techniques for testing mobile applications and generating automatic inputs are present in the literature. In this work two of them will be evaluated, the Model-Based Testing together with the FourMA tool and the Random Testing with the aid of the Monkey stress tool.

KEYWORDS: Mobile application testing. Automated Testing. Test Based on Model. FourMA. Random Test. Monkey.

AGRADECIMENTOS

A. T. Endo é parcilamente financiado pelo CNPq/Brasil (processo 445958/2014-6). H. N. Silva é bolsista PIBIC da UTFPR/Brasil.

REFERÊNCIAS

Shauvik Roy Choudhary, Alessandra Gorla, and Alessandro Orso. 2015. Automated Test Input Generation for Android: Are We There Yet? (E). In Proceedings of the 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE) (ASE '15). IEEE Computer Society, Washington, DC, USA, 429-440.

Farto, G. C. and Endo, A. T. (2017). Reuse of model-based tests in mobile apps. InProceedings of the 31st Brazilian Symposium on Software Engineering (SBES).

Gao, J. Z., Tsao, J., Wu, Y., and Jacob, T. H.-S. (2003). Testing and Quality Assurance for Component-Based Software. Artech House, Inc., Norwood, MA, USA.

Gudmundsson, V., Lindvall, M., Aceto, L., Bergthorsson, J., and Ganesan, D. (2016). Model-based testing of mobile systems - an empirical study on quizup android app. In Proceedings First Workshop on Pre- and Post-Deployment Verification Techniques, PrePost@IFM 2016, Reykjavík, Iceland, 4th June 2016., pages 16–30.

Lecheta, R. (2015). Google Android: aprenda a criar aplicações com dispositivos móveis com o Android SDK. Novatec, São Paulo, 5 edition.

Zhauniarovich, Y., Philippov, A., Gadyatskaya, O., Crispo, B., and Massacci, F. (2015). Towards black box testing of android apps. In 2015 Tenth International Conference on Availability, Reliability and Security (ARES), pages 501–510.

Recebido: 31 ago. 2017.
Aprovado: 02 out. 2017.
Como citar:
H. N. Silva; A. T. Endo. Geração de casos de teste aleatória e baseada em modelo para aplicações Android. In: SEMINÁRIO DE INICIAÇÃO CIENTÍFICA E TECNOLÓGICA DA UTFPR, 22., 2017, Londrina. **Anais eletrônicos...** Londrina: UTFPR, 2017. Disponível em: <https://eventos.utfpr.edu.br//sicite/sicite2017/index. Acesso em: 2017.
Correspondência:
Henrique Neves da Silva
Rua dos Bandeirantes, número 120, Centro, Cornélio Procópio, Paraná, Brasil.
Direito autoral:
Este resumo expandido está licenciado sob os termos da Licença Creative Commons-Atribuição-NãoComercial 4.0 Internacional.