



UNIVERSIDADE FEDERAL DE
PERNAMBUCO
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA
2018.2

AutoTestCoverage^p: Uma ferramenta para cobertura de testes de integração no contexto Android sem uso de código-fonte

Proposta de Trabalho de Graduação

Aluno: Lucas Bezerra Perrusi (lbp@cin.ufpe.br)

Orientador: Alexandre Cabral Mota (acm@cin.ufpe.br)

Recife, 10 setembro de 2018

Sumário

1. Introdução	3
2. Objetivos	3
3. Cronograma	4
4. Possíveis avaliadores	4
5. Referências	5
6. Assinaturas	5

1. Introdução

Um sistema é construído em fases: requisitos, especificação, projeto, implementação e testes. Dentre estas fases, uma das principais é a fase de testes. Nesta fase, objetiva-se verificar se o sistema projetado está funcionando da maneira esperada. Mas esta fase não é isolada das demais. Ou seja, ao invés dela ocorrer em apenas um momento no tempo, ela ocorre durante o desenvolvimento, participando de todas as outras fases do desenvolvimento de um sistema.

O teste chamado de regressão objetiva verificar se o que funcionava em uma versão anterior do sistema, continua funcionando na versão atual. Para tal, pode-se simplesmente re-executar todos os testes existentes. Só que isto é inviável na prática e alguma forma de selecionar testes faz-se necessária. Como forma de selecionar testes para as áreas que foram modificadas, torna-se usual realizar alguma forma de análise de cobertura, quer seja de código, de requisitos, de mudanças, etc.

Quando é possível ter acesso ao código do sistema e realizar a instrumentação na parte modificada, ou seja, introduzir pontos de verificação em partes específicas do código com o objetivo de obter informações sobre a execução do programa, torna-se mais fácil e preciso medir o quanto um caso de teste exercitou a região desejada.

Neste trabalho iremos focar na análise da cobertura em relação às solicitações de modificação entre versões do sistema Android. Isto porque este trabalho está inserido no convênio de pesquisa entre o Centro de Informática e a Motorola Mobility, onde não é possível ter acesso irrestrito ao código-fonte do sistema. Além disto, iremos focar mais nas interações entre aplicativos do sistema que nos detalhes internos do sistema em si. O que se denomina por teste de regressão de plataforma.

2. Objetivos

Este trabalho tem como objetivo principal avaliar a qualidade dos testes de regressão a nível de sistema, ou seja, avaliar os testes que tem como foco a interação entre aplicações após alterações nas mesmas, usando cobertura de código. Para tal, iremos utilizar a `AutoTestCoverageP`, uma ferramenta usada para capturar as modificações realizadas entre duas versões do Android a partir de solicitações de mudanças (em Inglês, `change requests`) e capturar as ações realizadas durante a execução dos casos de teste, e com isso calcular a cobertura em relação a região modificada no código.

Este trabalho se divide em dois processos, o primeiro é baseado em recuperação de informação das solicitações de mudança (as quais são agrupadas em `release notes`) para se definir quais regiões do sistema deverão ser exercitadas durante a execução dos casos de teste. Neste trabalho chamaremos esta região de `ToBeCovered (TBC)`. O segundo processo consiste em capturar as informações do device Android durante a execução dos testes, o que

chamaremos neste trabalho de Trace. Para isto utilizaremos o Android Debug Bridge (adb) para coletar logs emitidos pelo código executado.

Para atingir o objetivo final deste trabalho é feito um cruzamento de informações entre o que foi coletado durante o primeiro processo, o TBC, e os dados coletados durante a segunda parte do processo, o Trace, e desta forma teremos o cálculo da cobertura para um plano de testes.

3. Cronograma

Nessa seção, a Tabela 1 descreve o cronograma de atividades previsto para desenvolvimento desse trabalho de graduação ao longo desse semestre.

Atividades	Março	Abril	Maió	Junho	Julho
Elaboração da proposta					
Pesquisa bibliográfica					
Testes e experimentos					
Análise dos resultados					
Elaboração do relatório					
Preparação para apresentação e defesa					

Tabela 1- Cronograma de Trabalho de Graduação

4. Possíveis avaliadores

Os possíveis avaliadores para o resultado a ser obtido no final de todas as etapas descritas nesse documento são:

- Juliano Manabu Iyoda;
- Leopoldo Teixeira

5. Referências

- [1] Y. Tsutano, S. Bachala, W. Srisa-an, G. Rothermel, Jackson Dinh Department of Computer Science and Engineering University of Nebraska–Lincon, “An Efficient, Robust, and Scalable Approach for Analyzing Interacting Android Apps”. In Proceedings 39th International Conference on Software Engineering, IEEE/ACM, 2017.
- [2] S. Arzt, S. Rasthofer, C. Fritz, E. Bodden, A. Bartel, J. Klein, Y. Le Traon, D. Oceau, and P. McDaniel, “FlowDroid: Precise con-text, flow, field, object-sensitive and lifecycle-aware taint analysis for Android apps,” in Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation, 2014, pp. 259–269.
- [3] W. Eric Wong, J. R. Horgan, S. London, H. Agrawal, “A Study of Effective Regression Testing in Practice”. Proceedings The Eighth International Symposium on Software Reliability Engineering, Software Reliability Engineering, 1997.

6. Assinaturas

Lucas Bezerra Perrusi
Orientando

Alexandre Cabral Mota
Orientador