

Higor Amario de Souza, 5619750 – 15 de abril de 2013.

MAC 5779 – Engenharia de Software Experimental

**Projeto de Pesquisa (Doutorado): Desenho de estudo qualitativo baseado em entrevistas – questão de pesquisa, objetivos, roteiro e critérios de seleção de participantes.**

**Tema:** Depuração automatizada em programas contendo múltiplos defeitos.

**Problema e contextualização**

Diversas técnicas de localização de defeitos baseadas em cobertura de código têm sido propostas para reduzir o esforço manual de localização dispendido por programadores (Naish et al., 2010; Wong et al., 2010; Mariani et al., 2011). A partir da cobertura obtida dos testes executados, tais técnicas indicam comandos mais suspeitos de conter defeitos.

No entanto, os experimentos realizados são baseados em métodos quantitativos, medindo o esforço de localização a partir da lista de contendo os comandos mais suspeitos. Tal esforço é medido pela quantidade de código que precisa ser verificado até que o defeito seja localizado. Uma limitação com o uso da métrica de esforço de localização é a chamada situação de detecção perfeita. Isso significa que o defeito é considerado como localizado assim que o comando defeituoso é examinado. Na prática, um programador inspecionando o código pode examinar o comando defeituoso e não identificar o defeito, fato que pode levar a medida de esforço de localização a obter valores diferentes daqueles observados durante os experimentos realizados atualmente.

Além disso, os experimentos são realizados em programas contendo um único defeito por versão. O projeto de doutorado a ser desenvolvido irá propor e avaliar uma técnica para a localização de em programas contendo múltiplos defeitos. O desenho experimental serão realizados tanto com o uso de experimento controlado em diferentes programas contendo múltiplos defeitos quanto com o uso de programadores utilizando a técnica proposta para localizar defeitos.

**Questão de pesquisa**

A técnica proposta é eficaz para ser usada por desenvolvedores para localizar defeitos em programas reais?

O esforço de localização de defeitos da técnica proposta é equivalente quando utilizado por programadores?

**Objetivo geral**

Avaliar se a técnica proposta é utilizada pelos desenvolvedores para localizar o defeito.

**Objetivos específicos**

- Avaliar o esforço de localização para o uso prático da técnica realizando experimentos com desenvolvedores.

- Avaliar a opinião dos desenvolvedores a respeito da técnica proposta.

## **Seleção dos participantes**

Os participantes devem ser programadores com conhecimento da linguagem Java, contando com estudantes de graduação, pós-graduação e profissionais. Os participantes serão divididos pelo seu nível de experiência em programação. Um grupo será formado por estudantes sem experiência profissional e programadores com menos de 5 anos de experiência e outro grupo será formado por programadores profissionais (estudantes ou não) com experiência superior a 5 anos.

## **Roteiro**

Será feito um treinamento prévio para explicar o experimento e mostrar aos programadores como utilizar a técnica proposta.

Serão utilizados dois programas com tamanho e características semelhantes (domínio, quantidade de defeitos (2) e tipo de defeito) para o experimento. Cada grupo de programadores usará a técnica proposta em um dos programas e não usará a técnica no outro programa, alternadamente entre os grupos. Para o programa cuja localização será manual será executado um conjunto de testes que apresenta falhas para os programadores.

Para cada programa será estabelecido um limite de 40 minutos para localizar os defeitos. Após essa etapa os participantes irão responder a um questionário, para identificar as opiniões dos participantes a respeito do uso da técnica de localização proposta. O questionário é composto por questões abertas. Além do questionário, os cliques efetuados pelos programadores serão contados para medir o esforço de localização usando a técnica proposta, assim como o tempo de localização de cada defeito.

## **Questões**

1. Você conseguiu localizar os defeitos no programa em que usou a técnica?
2. Você conseguiu localizar os defeitos no programa em que não usou a técnica?
  - 2a. Se sim, como você localizou o(s) defeito(s) no programa em que não usou a técnica?
3. O uso da técnica facilitou a localização dos defeitos?
4. Você seguiu a ordem sugerida pela técnica?
  - 4a. Se não, qual foi o motivo?
  - 4b. Se você localizou o(s) defeito(s) sem seguir a ordem sugerida, como você localizou o(s) defeito(s)?
5. Você tem sugestões para melhorar a técnica?

## Referências

Collofello JS, Woodfield S. Evaluating the effectiveness of reliability-assurance techniques. *Journal of Systems and Software* 9 (3), 191–195. 1989.

Jones, J. A.; Bowring, J. F.; Harrold, M. J. Debugging in parallel. In: *Proceedings of the 2007 International Symposium on Software Testing and Analysis*. New York, NY, USA: ACM, 2007. (ISSTA '07), p. 16–26.

Mariani, L.; Pastore, F.; Pezze, M. Dynamic analysis for diagnosing integration faults. *Software Engineering, IEEE Transactions on*, v. 37, n. 4, p. 486 –508. 2011.

Naish, L.; Lee, H. J.; Ramamohanarao, K. Statements versus predicates in spectral bug localization. In: *Software Engineering Conference (APSEC), 2010 17th Asia Pacific*. [s.n.], 2010. p. 375 –384.

Parnin, C. Orso, A. Are automated debugging techniques actually helping programmers? In: *Proceedings of the 2011 International Symposium on Software Testing and Analysis*. Toronto, ON, Canada: ACM, 2011. (ISSTA '11), p. 199–209.

Wong, W. E.; Debroy, V.; Choi, B. A family of code coverage-based heuristics for effective fault localization. *Journal of Systems and Software*, v. 83, n. 2, p. 188–208, 2010.