



Universidade Federal de Campina Grande
Programa de Pós-Graduação em Ciência da Computação

Disciplina: Fundamentos de Pesquisa em Ciência da Computação 1
Semestre: 2018.1
Aluna: Flávia Coelho

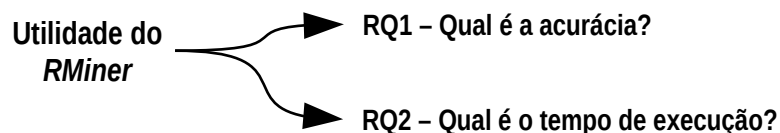
Proposta de *Design* de Experimento

REPLICAÇÃO

TSANTALIS, N., MANSOURI, M., ESHKEVARI, L. M., MAZINANIAN, D., DIG, D.
Accurate and Efficient Refactoring Detection in Commit History. In: 40th
International Conference on Software Engineering, May/June 2018, Gothenburg,
Sweden.

Informações Preliminares

O experimento realizado em [Tsantalis et al. 2018] compreende as perguntas de pesquisa (RQ1 e RQ2), a seguir esquematizadas, com o objetivo de comparar o *RMiner* ao estado-da-arte em detecção de refatoramento. *RMiner* é a ferramenta de detecção de automática de refatoramento desenvolvida pelos autores.



Para investigar a acurácia, considera-se precisão (p) e *recall* (r), conforme segue.

$$p = \frac{|VP|}{|VP| + |FP|}$$

$$r = \frac{|VP|}{|VP| + |FN|}$$

onde:

- $|VP|$ = número de refatoramentos detectados;
- $|FP|$ = número de instâncias sem refatoramento, detectadas como refatoramentos;
- $|FN|$ = número de refatoramentos não detectados.

Na investigação da RQ1, os autores (i) construíram um *dataset* validado a partir de projetos Java, disponíveis no *GitHub*¹, (ii) computaram a precisão e *recall* do *RMiner* e (iii) compararam a sua acurácia com o *RefDiff* (considerada estado-da-arte em detecção de refatoramento, pelos autores). Para a RQ2, efetuaram a computação e a comparação do tempo de execução das ferramentas.

Proposta de *Design* de Replicação do Experimento

Objetivo

Analisar as ferramentas de detecção de refatoramento *RMiner* e *RefDiff*, com a intenção de avaliar a acurácia (precisão e *recall*) e o desempenho, no contexto de códigos-fontes desenvolvidos em Java.

Contexto

O experimento será executado para um contexto específico, baseado em um problema real (dados coletados a partir do *GitHub*) e de modo *offline*, em um ambiente computacional.

Hipóteses

Este experimento é uma replicação, na qual as hipóteses propostas a seguir, consideram os resultados em função da tabela 1.

¹ *GitHub*, github.com, é um repositório de código-fonte, cujo gerenciamento de revisões de projetos é efetuado pelo sistema de controle de versões *Git*.

Classificação	Descrição
igual	Resultado total relativo entre a utilização do <i>RMiner</i> versus <i>RefDiff</i> proporcional ao obtido em [Tsantalis et al. 2018]
diferente	Resultado total relativo entre a utilização do <i>RMiner</i> versus <i>RefDiff</i> não-proporcional ao obtido em [Tsantalis et al. 2018]

Tabela 1 – Significado de igualdade e diferença para as hipóteses

- **H1₀**: A precisão das ferramentas de detecção de refatoramento *RMiner* e *RefDiff* é igual;
- **H1_a**: A precisão das ferramentas de detecção de refatoramento *RMiner* e *RefDiff* é diferente;
- **H2₀**: O *recall* das ferramentas de detecção de refatoramento *RMiner* e *RefDiff* é igual;
- **H2_a**: O *recall* das ferramentas de detecção de refatoramento *RMiner* e *RefDiff* é diferente;
- **H3₀**: O tempo de execução das ferramentas *RMiner* e *RefDiff* é igual;
- **H3_a**: O tempo de execução das ferramentas *RMiner* e *RefDiff* é diferente.

Variáveis

A variável de controle (**independente**) é exibida na tabela 2.

Nome	Classificação	Níveis
Ferramenta de detecção	Categórica	<i>RMiner</i>
		<i>RefDiff</i>

Tabela 2 – Variável de controle

As variáveis **dependentes** são descritas na tabela 3.

Nome	Classificação	Faixa de valores
Precisão	<i>Numérica</i>	0 a 100% (precisão perfeita)
Recall	<i>Numérica</i>	0 a 100% (<i>recall</i> perfeito)
Tempo de execução	<i>Numérica</i>	Não se aplica

Tabela 3 – Variáveis dependentes

Unidade Experimental

Serão considerados *datasets* de instâncias de refatoramento publicamente disponíveis.

Design Experimental

Serão considerados o princípio da **randomização** (na seleção de amostras de instâncias de refatoramento a partir do *dataset* original), sem **balanceamento** e o **design fatorial completo** sem repetição.

2 tratamentos serão aplicados, compreendendo os 2 tipos de ferramentas de detecção de refatoramento e 1 nível de *dataset*. Portanto, 2 * 1 tratamentos distintos, sem repetições.

Instrumentação

A tabela 4 exibe a especificação dos instrumentos a serem utilizados.

Categoria	Descrição	
Objetos	<i>Código-fonte/ documentação</i>	RMiner, disponíveis em github.com/tsantalís/RefactoringMiner
		RefDiff, disponíveis em github.com/aserg-ufmg/RefDiff
	<i>Computador</i>	Configuração Intel core i7-2620M CPU@2.70 GHz, 16 GB DDR3, 1TB SSD, Windows 10 OS, definida em [Tsantalís et al. 2018]

Categoria	Descrição	
Diretrizes	Não se aplica	
Instrumentos	<i>Desenvolvimento/ execução/análise</i>	Java 1.8.0 x64, conforme especificado em [Tsantalis et al. 2018] IDE Eclipse Oxygen.3a Release (4.7.3a) R 3.4.4 RStudio 1.1.423
	<i>Medição</i>	Método Java System.nanoTime, de acordo com [Tsantalis et al. 2018]

Tabela 4 – Instrumentação

Avaliação da Validade

Em princípio, serão consideradas as ameaças à:

- Validade estatística de conclusão, pois é relevante examinar a significância estatística para a relação fator-efeito investigada neste experimento, com base nos *datasets*;
- Validade interna, para evidenciar conclusões a respeito da causalidade da relação fator-efeito;
- Validade de constructo, a fim de identificar se é possível generalizar os resultados obtidos nesta replicação em função, por exemplo, dos níveis aplicados à variável de controle;
- Validade externa, para evidenciar a possibilidade ou não de generalizar os resultados obtidos além do escopo desta replicação.