



DEFINING METRIC THRESHOLDS FOR SOFTWARE PRODUCT LINES: A COMPARATIVE STUDY

APRESENTADO POR: BRUNO LUAN DE SOUSA

DADOS TÉCNICOS

- **Título**

- Defining Metric Thresholds for Software Product Lines: A Comparative Study.

- **Autores**

- Gustavo Vale; Danyllo Albuquerque; Eduardo Figueiredo; Alessandro Garcia

- **Ano de Publicação**

- 2015

- **Publicação**

- Proceedings of the 19th International Conference on Software Product Line

ORGANIZAÇÃO DO TRABALHO

- Introdução
- Trabalhos Relacionados
- Configuração do Estudo
 - Métodos para Derivação de Valores Referências para Métricas
 - Métricas de Softwares Seleccionadas
 - *Benchmarks* de Linhas de Produto de Software
 - Oráculo de *Code Smells*

ORGANIZAÇÃO DO TRABALHO

- Avaliação
 - Correlação com LOC
 - Distribuição das Métricas de Softwares
 - Valores Referências derivados
 - Identificação de *Code Smells*
- Lições Aprendidas
- Ameaças à Validade
- Conclusão e Trabalhos Futuros
- Agradecimentos

MOTIVAÇÃO

- Alto crescimento do tamanho e complexidade dos softwares.
- Dificuldade de controle da qualidade em Linhas de Produtos de Softwares (LPSs).
- Alta demanda das LPSs na indústria.
- Eficiência da medição é fundamental para a qualidade das LPSs.
- Métricas são essenciais.
 - Altamente dependente de Valores Referências.
- Dificuldade uma derivação pontual desses limiares.
- Existência de vários métodos para derivação de Thresholds.

OBJETIVOS

- Comparar métodos para derivação de valores referências para LPSs.
 - Foco na distribuição assimétrica das métricas de software.
- Investigar o contexto em que cada método é bem sucedido.
- Etapas:
 - Construção de um referencial teórico.
 - Escolha métricas.
 - Agrupamento de softwares LPSs.
 - Derivação de Valores Referências.
 - Análise dos Valores Referências.

TRABALHOS RELACIONADOS

- Revisão da Literatura baseada em uma Revisão Sistemática.
- Bases Pesquisas:
 - *IEEEExplore* ;
 - *Science Direct*;
 - *ACM Digital Library*;
 - *El Compendex*;
- Uso da técnica *snowballing*.
- Extração de 50 estudos primários.

TRABALHOS RELACIONADOS

- Sumarização dos métodos de derivação:
 - *Thresholds* derivados da experiência de programação;
 - *Thresholds* derivados de análise de medição;
 - Metodologias para caracterização da distribuição de métricas;

CONFIGURAÇÃO DO ESTUDO

- Organizado em quatro etapas:
 - Métodos para derivação de *Thresholds*.
 - Seleção das métricas de software.
 - *Benchmarks* de linha de produto de software.
 - Oráculo para *Code Smells*.

CONFIGURAÇÃO DO ESTUDO

- Métodos para derivação de *Thresholds*.
 - Comparação de três métodos nesse estudo.
 - Métodos:
 - Método proposto por Alves et. al. (2010).
 - Método proposto por Ferreira et. al. (2012).
 - Método proposto por Oliveira et. al. (2014).

CONFIGURAÇÃO DO ESTUDO

- Seleção das métricas de software.
 - Métricas selecionadas.
 - Linhas de Código (LOC);
 - Acoplamento entre classes de objetos (CBO);
 - Métodos ponderados por classe (WMC);
 - Número de refinamento de constantes (NCR);
 - Compõe uma estratégia de detecção.
 - Coleta automática por meio de uma ferramenta: *Variability Smell Detection (VSD)*.

CONFIGURAÇÃO DO ESTUDO

- *Benchmarks* de linha de produto de software.
 - Foco em SPLs desenvolvidas em FOP.
 - Repositórios utilizados na coleta:
 - *SPL2go*;
 - *FeatureIDE*;
 - 47 softwares extraídos dos dois repositórios.
 - 17 softwares extraídos de artigos.
 - Critérios de inclusão e exclusão:
 - Versões mais recentes;
 - Seleção de implementações *AHEAD* e *FeatureHouse*;
 - Total de 33 softwares selecionados.

CONFIGURAÇÃO DO ESTUDO

- Oráculo para *Code Smells*.

- Foi produzido um oráculo para avaliação dos valores referências.

- Passos seguidos:

- Escolha de uma LPS alvo;

- Definição do *bad smell*;

- Seleção de uma estratégia de detecção;

- Validação dos resultados com um especialista;

- LPS alvo: *MobileMedia*.

- *Bad Smell: God Class*.

- Estratégia de detecção: $((LOC > threshold) \text{ and } (CBO > threshold) \text{ and } ((WMC/LOC) > threshold)) \text{ or } (NCR > threshold)$

AVALIAÇÃO

- Planejamento:
 - Correlação com LOC.
 - Distribuição das métricas de softwares.
 - Valores Limites Derivados.
 - Identificação de *Code Smells*.

AVALIAÇÃO

Benchmark	Metrics		
	CBO	WMC	NCR
1	0.751621	0.976406	0.28995
2	0.753825	0.97711	0.295099
3	0.757247	0.97896	0.292746

Figura 1: Correlação das métricas com LOC.

AVALIAÇÃO

Metric	Benchmark	α	β
LOC	1	0.95201	26.858
	2	0.72892	28.37
	3	0.93964	27.232
CBO	1	1.1253	5.2798
	2	1.1693	5.4341
	3	1.2244	5.6196
WMC	1	0.7277	6.5979
	2	0.72748	6.6063
	3	0.72041	6.5979
NCR	1	0.98339	3.9359
	2	0.97109	3.9679
	3	0.96919	4.0734

Figura 2: Valores da Distribuição das métricas.

AVALIAÇÃO

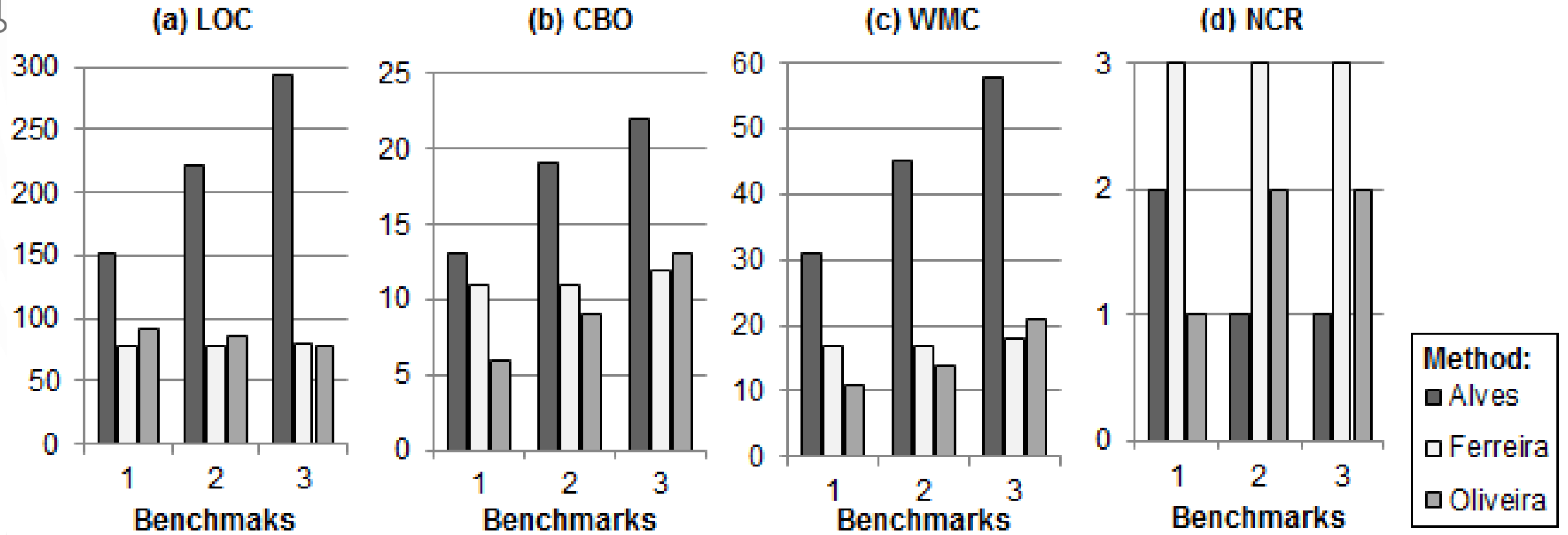


Figura 3: Comparação dos Valores Referências Derivados.

AVALIAÇÃO

Benchmark	Alves		Ferreira		Oliveira	
	P	R	P	R	P	R
1	62,5	71,4	85,7	85,7	46,6	100
2	38,5	71,4	85,7	85,7	63,6	100
3	38,5	71,4	85,7	85,7	62,5	71,4

Figura 4: Resultado de *Recall* e *Precision*.

CONCLUSÃO

- Este artigo mostra a importância das métricas em conjunto e dos valores referências para as mesmas.
- Comparação de três métodos de derivação de valores referências.
- Criação de um repositório com 64 LPSs.
- Método proposto por Alves et. al. (2010) mostrou-se mais sensível ao *benchmark* utilizado.
- LOC, WMC e NCR possuem distribuição de cauda pesada e CBO possui uma distribuição diferente.

CONCLUSÃO

- Método proposto por Alves et. al. (2010) mostrou-se melhor na avaliação individual.
- Método proposto por Ferreira et. al. (2012) apresentou uma maior precisão.
- Método proposto por Oliveira et. al. (2014) apresentou um maior *recall*.
- Trabalhos Futuros:
 - Explorar alguns pontos desejáveis em métodos de derivação de valores referências.
 - Construir uma ferramenta de medição para códigos desenvolvidos em *FeatureHouse*.
 - Explorar como construir *benchmarks* representativos.

REFERÊNCIAS

- Alves, T.L., Ypma, C. and Visser, J. Deriving Metric Thresholds From Benchmark Data. *In Proc. of 26th Int. Conf. on Software Maintenance (ICSM)*, pp. 1–10, 2010.
- Ferreira, K., Bigonha, M., Bigonha, R., Mendes, L. and Almeida, H. Identifying Thresholds for Object-Oriented Software Metrics. *Journal of Systems and Software*, vol. 85, Issue 2, pp. 244–257, February, 2012.
- Oliveira, P., Valente, M.T. and Lima, F.P. Extracting Relative Thresholds for Source Code Metrics. *In Proceedings of the Conference on Software Maintenance, and Reengineering (CSMR)*, pp.254-263, 2014.
- Vale, G., Albuquerque, D., Figueiredo, E., and Garcia, A. Defining metric thresholds for software product lines: a comparative study. *In Proceedings of the 19th International Conference on Software Product Line (SPLC '15)*, ACM, New York, NY, USA, 176-185, 2015.



OBRIGADO!