



UNIVERSIDADE
FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Desenvolvendo Um Índice De Manutenibilidade De Modelos De *Features* Utilizando Técnicas De *Machine Learning* Para Apoiar O Processo De Evolução De Linhas De Produto De Software

Publio Blenilio Tavares Silva

Orientadora:
Profa. Dra. Carla Ilane Moreira Bezerra

Trabalho de Conclusão de Curso I
Julho 2020



Motivação

- Linhas de Produtos de Software (LPS)
 - Modelo de *Features* (MF)
 - As mudanças sofridas pela LPS geralmente implicam em mudanças no modelo de *features* (BÜRDEK *et al.*, 2016)
- Estratégias de Avaliação da Manutenibilidade de MFs
 - Métricas de Manutenibilidade
 - É necessário analisar diversos valores
 - Em geral, as faixas de valores das métricas são amplas pode não haver um indicativo claro de quais valores podem ser considerados adequados ou inadequados (OLIVEIRA BEZERRA, 2019)
 - Índice de Manutenibilidade



Motivação

- Melhoria da Qualidade de Modelos de *Features*
 - Refatoração de Inconsistências de MFs
 - Refatoração de *Smells* de Variabilidade



Objetivos

Objetivo Geral

Desenvolver um índice de manutenibilidade de modelos de *features* utilizando técnicas de *machine learning* aplicadas a um *dataset* de métricas de manutenibilidade de modelos de *features* e fornecer aos engenheiros de domínio sugestões para melhoria da qualidade de modelos de *features*.



Objetivos

Objetivos Específicos

- Estimar o valor do índice de manutenibilidade utilizando técnicas de *machine learning*
- Identificar na literatura refatorações para inconsistências de MFs e *smells* de variabilidade
- Sugerir aos engenheiros de domínio melhorias de MFs baseadas na refatoração de inconsistências e *smells*
- Adicionar a ferramenta DyMMer o cálculo do índice de manutenibilidade e as sugestões de melhoria de MFs



Fundamentação Teórica

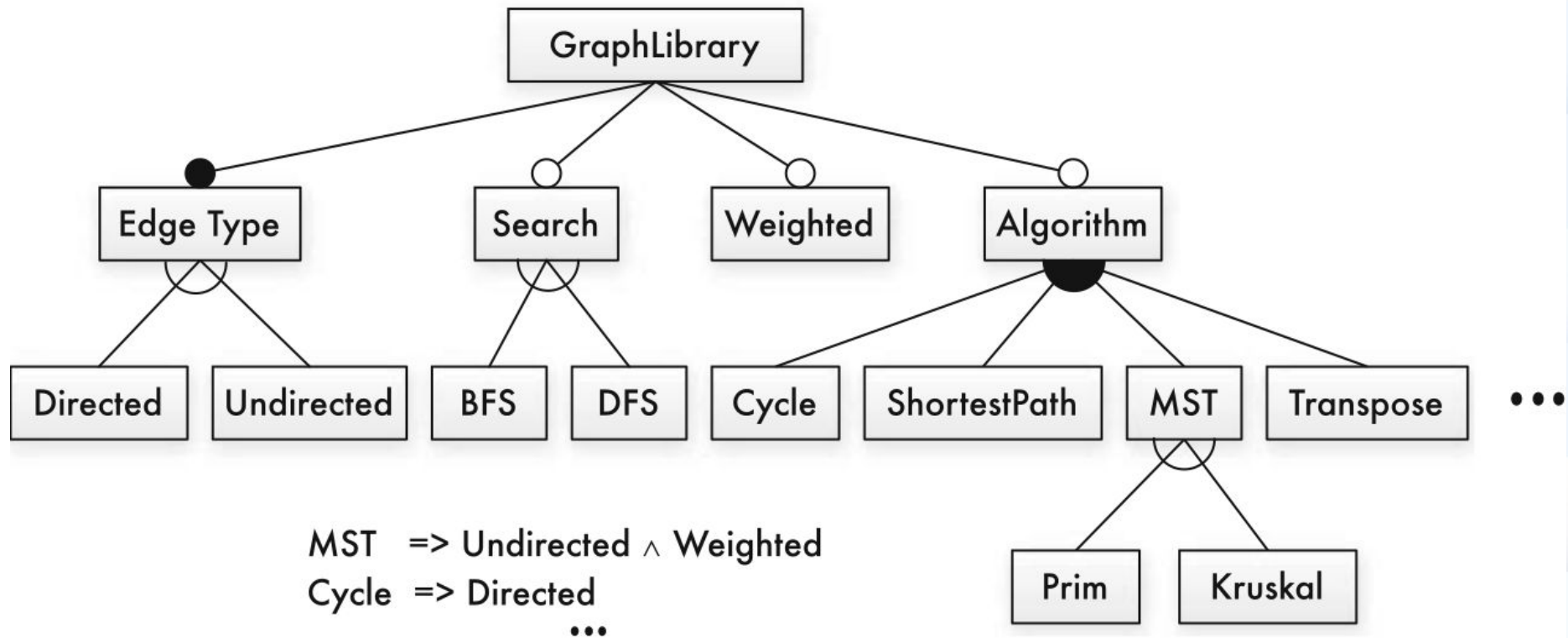


Linhas de Produtos de Software

- Uma linha de produtos de software (LPS) é uma família de sistemas criada a partir de um conjunto de *features* em comum (SOARES *et al.*, 2018)
- O modelo de *features* especifica todas as *features* de uma LPS, as restrições existentes entre elas e o conjunto de todas as combinações válidas entre as *features* (BÜRDEK *et al.*, 2016)



Modelo de *Features*



Métricas de Manutenibilidade

- Atributos de Qualidade
- Manutenibilidade
 - Aproximadamente 70% do tempo de desenvolvimento de um produto de software envolve atividades de manutenção (JHA *et al.*, 2019)
 - LPS evoluem durante todo o seu ciclo de vida, enquanto que nas metodologias tradicionais a evolução acontece predominantemente na fase de manutenção (BEZERRA *et al.*, 2016)



Métricas de Manutenibilidade

MiniCOfFEE (BEZERRA *et al.*, 2017)

- N° de *Features*
- N° de *Features* Obrigatórias
- N° de *Features* Folha
- N° de *Features* do Topo
- Complexidade Cognitiva
- Extensibilidade das *Features*
- Flexibilidade de Configuração
- Profundidade Máxima da Árvore
- Número de Configurações Válidas
- Razão de Variabilidade
- Coeficiente de Densidade de Conectividade
- Número de Grupos XOr
- Número de Grupos Or
- Número de *Features* Dependentes de Ciclos Únicos
- Ciclos Múltiplos Dependentes de *Features*



Método Vale

- É um método de derivação de *thresholds* (VALE *et al.*, 2019)
- Mesmo com o uso de métricas ainda é difícil avaliar a qualidade de um artefato sem o auxílio de limiares ou *thresholds* para indicar quais faixas de valores podem ser consideradas adequadas ou inadequadas (VALE *et al.*, 2019)
- Faixas de Valores:
 - Muito Baixo
 - Baixo
 - Moderado
 - Alto
 - Muito Alto



Machine Learning

- “Um conjunto de métodos que permitem aos computadores aprender com os dados para fazer e melhorar previsões” (ALPAYDIN, 2020)
- Aprendizado Supervisionado
 - Exige que o valor do dado alvo seja conhecido
 - Problemas de Regressão: um problema de aprendizado supervisionado no qual o dado alvo é numérico (MARSLAND, 2015)
 - Métricas:
 - R^2 (*Coefficient of Determination*)
 - MAE (*Mean Absolute Error*)
 - RMSE (*Root Mean Squared Error*)



Inconsistências, *Smells* e Refatorações em MF

- Inconsistências são informações conflitantes em um modelo de *features* e são consideradas um problema grave, pois impedem a derivação de instâncias de produto válidas (JAVED *et al.*, 2016)
- Abordagem para detecção de inconsistências proposta por Bhushan *et al.* (2018)
- *Smells* são indicadores de uma propriedade indesejada em uma LPS (VALE *et al.*, 2014)
 - *Smells* de Variabilidade
 - No presente trabalho, serão considerados apenas *smells* de variabilidade que podem ser detectados através da análise do modelo de *features* e configurações de produtos
- Refatorações



Trabalhos Relacionados

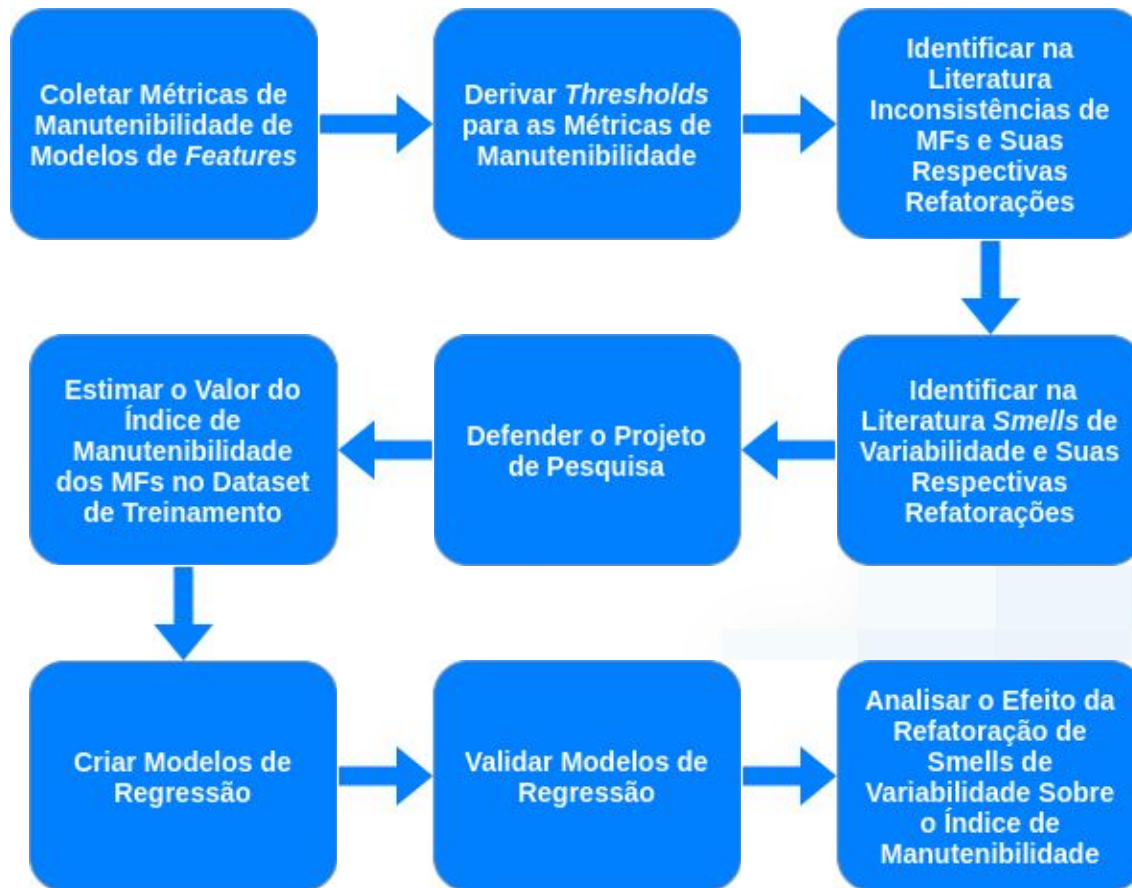
Trabalho	Avalia a Qualidade de MFs	Usa Métrica Única	Lida com Inconsistências em MFs	Lida com <i>Smells</i> em MFs	Tem Suporte Automatizado
Bezerra <i>et al.</i> (2015)	✓				
Oliveira e Bezerra (2019)	✓	✓			✓
Souza <i>et al.</i> (2019)				✓	
Trabalho Proposto	✓	✓	✓	✓	✓



Procedimientos metodológicos



Procedimentos Metodológicos



Cronograma

ATIVIDADES	2020											
	Jun	Jul	Ago	Set	Out	Nov	Dez					
Coletar Métricas de Manutenibilidade de Modelos de Features	X											
Derivar Thresholds para as Métricas de Manutenibilidade		X										
Defender o Projeto de Pesquisa		X										
Identificar Smells de Variabilidade e Suas Respectivas Refatorações		X	X	X	X							
Identificar Inconsistências de MFs e Suas Respectivas Refatorações		X	X	X	X							
Estimar o Valor do Índice de Manutenibilidade dos MFs no Dataset de Treinamento		X	X	X	X							
Criar Modelos de Regressão					X	X						
Validar Modelos de Regressão					X	X						
Analisar o Efeito da Refatoração de Smells de Variabilidade Sobre o Índice de Manutenibilidade							X	X				
Revisão final da monografia										X		
Defesa da monografia											X	-



Resultados Preliminares

- Coleta das Métricas de Manutenibilidade de 272 modelos de *Features*
- Derivação de *thresholds* com método Vale
 - Criação de um script para automatizar a tarefa
- Identificação na literatura de inconsistências de MF e refatorações para removê-las
 - Criação de um script para detecção de inconsistências
- Identificação de *smells* na literatura



Inconsistências

Inconsistência	Refatoração
Implicação e exclusão simultâneas	Remover uma das restrições (implicação ou exclusão)
Exclusão entre <i>features</i> obrigatórias	Remover restrição de exclusão
Exclusão entre uma <i>feature</i> obrigatória e uma <i>feature</i> opcional	Remover restrição de exclusão
Implicação entre <i>features</i> alternativas	Remover restrição de implicação



Smells

<i>Smell</i>	Refatoração
<i>Unused feature</i>	Remoção da <i>feature</i>
<i>Unused variability</i>	Remoção da variabilidade (tornar a <i>feature</i> obrigatória)



Referências

ALPAYDIN, E. Introduction to machine learning. [S. l.]: MIT press, 2020.

BEZERRA, C. I. BARBOSA, J. FREIRES, J. H. ANDRADE, R. MONTEIRO, J. M. Dymmer: a measurement-based tool to support quality evaluation of dspl feature models. In: ACM. Proceedings of the 20th International Systems and Software Product Line Conference. [S. l.], 2016. p. 314–317.

BÜRDEK, J. KEHRER, T. LOCHAU, M. REULING, D. KELTER, U. SCHÜRR, A. Reasoning about product-line evolution using complex feature model differences. Automated Software Engineering, Springer, v. 23, n. 4, p. 687–733, 2016.

JAVED, M. NAEEM, M. UMAR, A. I. BAHADUR, F. Automated inconsistency detection in feature models: A generative programming based approach. Selforganizology, v. 3, n. 2, p. 59–74, 2016.

JHA, S. KUMAR, R. ABDEL-BASSET, M. PRIYADARSHINI, I. SHARMA, R. LONG, H. V. et al. Deep learning approach for software maintainability metrics prediction. Ieee Access, IEEE, v. 7, p. 61840–61855, 2019.

MARSLAND, S. Machine learning: an algorithmic perspective. [S. l.]: CRC press, 2015.



Referências

OLIVEIRA, D. C. S. de BEZERRA, C. I. Development of the maintainability index for spls feature models using fuzzy logic. In: Proceedings of the XXXIII Brazilian Symposium on Software Engineering. [S. l.: s. n.], 2019. p. 357–366.

SOARES, L. R. SCHOBENS, P.-Y. MACHADO, I. do C. ALMEIDA, E. S. de. Feature interaction in software product line engineering: A systematic mapping study. Information and Software Technology, Elsevier, v. 98, p. 44–58, 2018.

VALE, G. FERNANDES, E. FIGUEIREDO, E. On the proposal and evaluation of a benchmark-based threshold derivation method. Software Quality Journal, Springer, v. 27, n. 1, p. 275–306, 2019.

VALE, G. FIGUEIREDO, E. ABÍLIO, R. COSTA, H. Bad smells in software product lines: A systematic review. In: IEEE. 2014 Eighth Brazilian Symposium on Software Components, Architectures and Reuse. [S. l.], 2014. p. 84–94.

