

Modelando o Problema da Próxima Release sob a Perspectiva da Análise de Pontos de Função



Aluno: Vitor Padilha Gonçalves

Orientador: Márcio Barros

Agenda

- Introdução
- Conceitos
 - Análise de Pontos de Função
 - *Next Release Problem* (NRP)
- Formalização do Problema
- Prova de Conceito
- Conclusão

Introdução

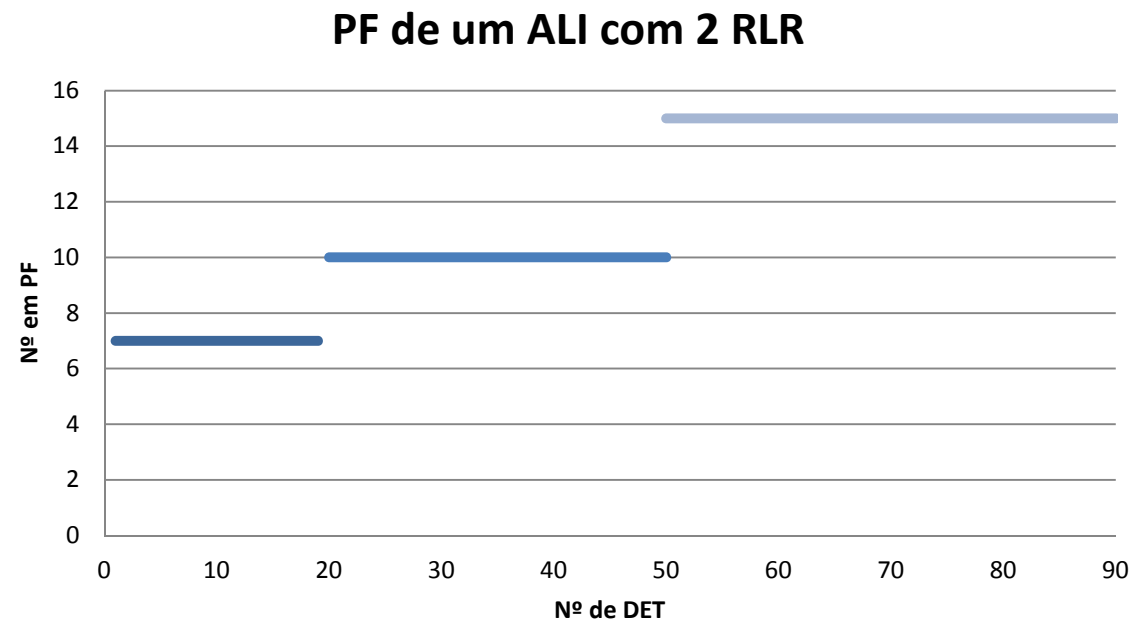
- A Análise de Pontos de Função (APF) é utilizada para medir software.
 - Prazo
 - Custo
 - Recursos
- O Problema da Próxima Release (Next Release Problem – NRP) consiste em escolher os requisitos mais importantes de maneira a maximizar a expectativa dos interessados nas primeiras entregas de software;
- Restrições do problema: prazo, custo, equipe, ... (esforço).

Análise de Pontos de Função

- IFPUG (2010) define o método de contagem de pontos de função.
- Conforme IFPUG(2010), consiste em:
 - Contar funções de dados: DERs x RETs.
 - Contar funções de transação: ALRs (funções de dados utilizadas) x DERs (DETs utilizados)
 - Aplicar fator de ajuste: Fator de ajuste a requisitos não funcionais. (Não nos interessa)
- **Utilizada em contratos de TI pelo Governo.**

Análise de Pontos de Função

- É expressa por uma função descontínua discreta.
- Ex: Número de PF de um Arquivo Lógico Interno



NRP

- Abordagem mono-objetiva: Maximizar a satisfação dos interessados a fim de atender uma restrição de custo.
- Abordagem bi-objetiva: Maximizar a expectativa dos interessados x Minimizar o esforço (valores do requisitos) (Zhang et al. 2007)
- Algoritmos: Genético, Hill Climbing, Simulated Annealing, NSGA-II, entre outros.

NRP – Formalização Básica

- Abordagem Mono-objetiva:

$$\sum_{i \in R'} c_i \leq B \quad (1)$$

$$\text{Maximizar } f(2) = \sum_{j=1}^m p_j \sum_{i \in R'} v_{ij} \quad (2)$$

B = Investimento Máximo


c_i = Custo de um requisito $i \in R$

p_j = Importância do patrocinador j no projeto

v_{ij} = Importância de um requisito i para um patrocinador j

NRP – Formalização Básica

- Abordagem Mono-objetiva:

APF 	$\sum_{i \in R'} c_i \leq B$	(1)
--	------------------------------	-----

	Maximizar $f(2) = \sum_{j=1}^m p_j \sum_{i \in R'} v_{ij}$	(2)
--	--	-----

B = Investimento Máximo

c_i = Custo de um requisito $i \in R$

p_j = Importância do patrocinador j no projeto

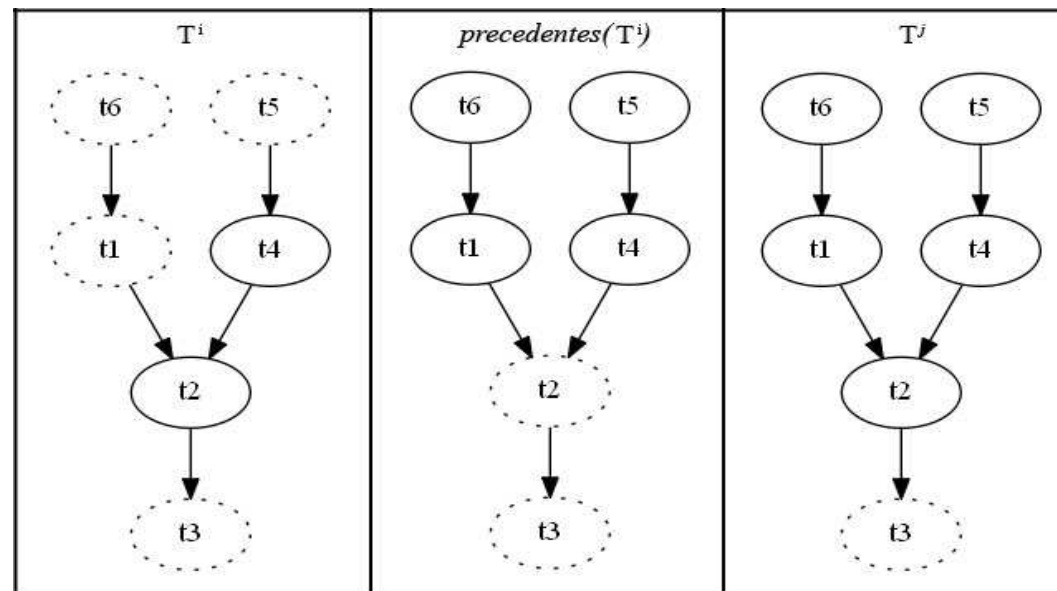
v_{ij} = Importância de um requisito i para um patrocinador j

Proposta de Solução

- Cada transação está associada a um grau de interesse dos *stakeholders*. Ex: Transação “Imprimir de Contra-cheque” tem importância de 100 para A e importância de 50 para B.
- Realizar uma busca que encontre um conjunto de transações, bem como, as funções de dados utilizadas por elas que maximize a expectativas dos stakeholders na primeiras entregas. Desta maneira, as últimas entregas serão as “menos importantes”.

Formalização do Problema

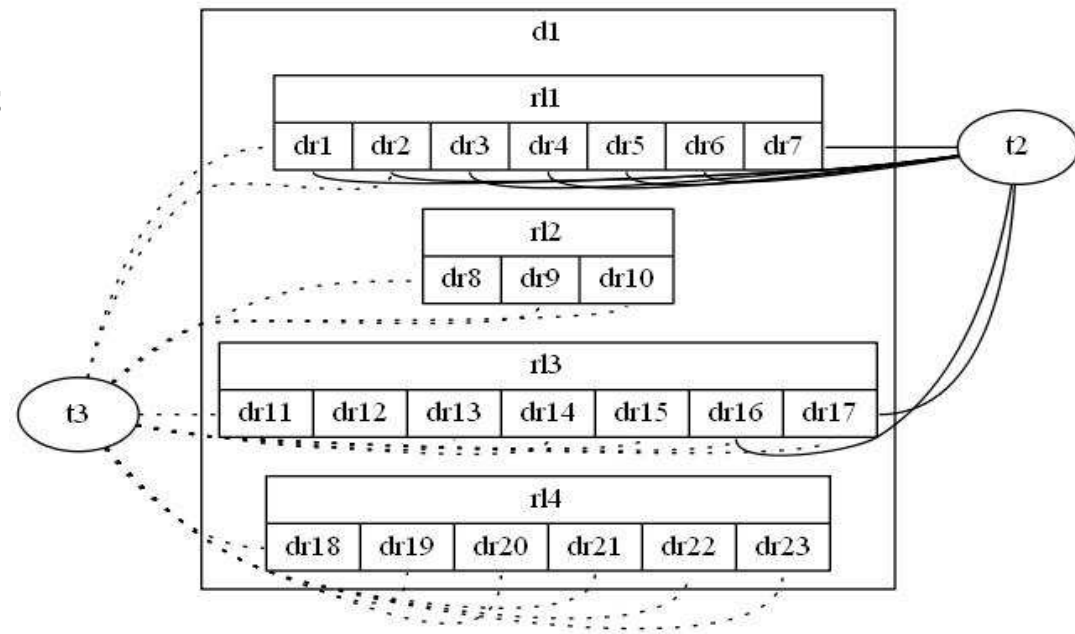
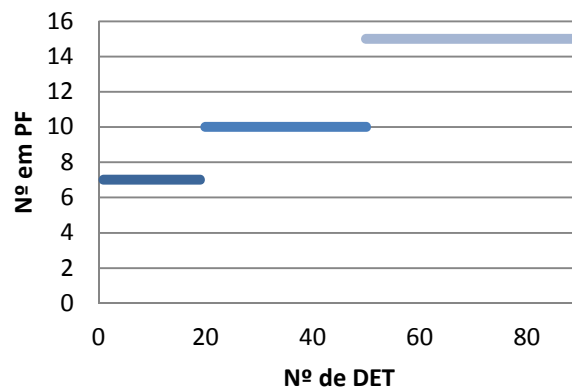
- Escolhido um subconjunto inicial de transações a partir de alguma heurística.
 - Análise de dependências: Escolhido um subconjunto inicial de funções de transação, adicionar outras funções de transações que devem ser desenvolvidas antes deste subconjunto;



Formalização do Problema

- Escolhido o subconjunto de função de transação com seus precedentes.
 - Adicionar ao escopo apenas DER's e/ou RLR's que são referenciados pelas funções de transação já adicionadas, bem como, as funções de dados na qual estes elementos estão contidos.
 - DER \in RLR
 - RLR \in Função de Dados

PF de um ALI com 2 RLR



Formalização do Problema

- Sabendo que:

- PFD': Valor que representa o número de pontos de função das funções de dados escolhidas.
- PFT': Valor que representa o número de pontos de função das funções de transação escolhidas.
- P_j : Peso que representa a importância do patrocinador no projeto.
- V_{ij} : Valor de uma função de transação i para um patrocinador j .

- Objetivo:

$$\text{Minimizar } f(1) = PFD' + PDT' \quad (1)$$

$$\text{Maximizar } f(2) = \sum_{j=1}^k p_j \sum_{i \in T^f} v_{ij} \quad (2)$$

Prova de Conceito

- Sistema com 65 funções de transação (221 PF) e 9 funções de dados (67 PF).
- Algoritmo mono-objetivo: Máximo de PF por release fixado em 96 PF.
- 50 execuções.
- Algoritmo Genético: 16900 avaliações e população inicial de 130 indivíduos.

Prova de Conceito

Release	Informação	Media	Mínimo	Máximo
1 ^a	Pontos de função	94,72 ± 1,21	93,0	96,0
	Satisfação (%)	44,90 ± 0,60	44,6	45,9
2 ^a	Pontos de função	190,3 ± 1,34	187,0	192,0
	Satisfação (%)	75,51 ± 0,40	74,6	75,9
3 ^a	Pontos de função	285,38 ± 1,70	281,0	288,0
	Satisfação (%)	99,64 ± 0,22	99,09	100,0
4 ^a	Pontos de função	288,0 ± 0	288,0	288,0
	Satisfação (%)	100,0 ± 0	100,0	100,0
	Tempo de execução (seg)	44,4 ± 0,37	43,8	46,0

Conclusões e Trabalhos Futuros



- 1ª Release: Apesar de medir $1/3$ dos pontos de função inicial, atende em média 45% da satisfação dos patrocinadores.
- Proposta x Análise Sistemática: Tempo ganho
- Trabalhos Futuros
 - Explorar o problema bi-objetivo.
 - Novas Instâncias.
 - Avaliar outros algoritmos.

Referências

- Bagnall, A. J.; Rayward-Smith, V. J.; Whittle, I. M. (2001). The next release problem. *Information and Software Technology*, v. 43, n. 14, p. 883–890
- Durillo, J.J., Zhang, Y., Alba, E., Harman, M., and Nebro, A.J. (2011). A study of the bi-objective next release problem. *Empirical Softw. Engg.* 16, 29–60.
- IFPUG (2009). *Function Point Counting Practices Manual (Release 4.3)*.
- Li, C., van den Akker, M., Brinkkemper, S., and Diepen, G. (2010). An integrated approach for requirement selection and scheduling in software release planning. *Requirements Engineering* 15, 375–396.
- del Sagrado, J., Águila, I.M., and Orellana, F.J. (2011). Requirements interaction in the next release problem. In *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation*, (New York, NY, USA: ACM), pp. 241–242.
- Souza, J.T. de, Maia, C.L.B., Ferreira, T. do N., Carmo, R.A.F. do, and Brasil, M.M.A. (2011). An Ant Colony Optimization Approach to the Software Release Planning with Dependent Requirements. In *Search Based Software Engineering*, M.B. Cohen, and M.Ó. Cinnéide, eds. (Springer Berlin Heidelberg), pp. 142–157.
- Tonella, P., Susi, A., and Palma, F. (2013). Interactive requirements prioritization using a genetic algorithm. *Information and Software Technology* 55, 173–187.

Dúvidas?

Sugestões?