



Universidade Federal do Amazonas – UFAM
Instituto de Ciências Exatas – ICE
Programa de Pós-Graduação em Informática



TÉCNICAS DE INSPEÇÃO DE MODELOS DE LINHA DE PRODUTO PARA MELHORIA EM QUALIDADE DE SOFTWARE

Rafael Normando Cunha
Orientadora: Tayana Conte

Linha de Produto de Software

- É uma implementação de um conjunto de produtos de software que apresentam características comuns entre si.
- Permite às organizações explorar semelhanças entre seus produtos, aumentando, assim, a reutilização de artefatos e, como consequência, tem-se uma diminuição dos custos e do tempo no desenvolvimento [HEYMANS e TRIGAUX, 2003].

Linha de Produto de Software

- Diferente versões
- Variabilidade
- Gerenciamento de configuração



Linha de Produto de Software

“uma linha de produto de *software* é um conjunto de sistemas que usam *software* intensivamente, compartilhando um conjunto de características comuns e gerenciadas, que satisfazem as necessidades de um segmento particular de mercado ou missão, e que são desenvolvidos a partir de um conjunto comum de ativos principais e de uma forma preestabelecida”

[CLEMENTS e NORTHROP, 2001]

Modelo de LPS

- *Feature Model*

- *Feature* é todo aspecto visual proeminente ou distintivo para o usuário, qualidade ou característica de um sistema
- Definir as *features* e suas dependências

Relacionamento	Tipo	Semântica	Notação
Relacionamento de Domínio	Mandatório	Se a <i>feature</i> pai é selecionada, o filho também devem ser selecionado	
	Opcional	Se a <i>feature</i> pai é selecionada, a funcionalidade filha pode ser selecionada	
	Alternativa	Se a <i>feature</i> pai é selecionada, exatamente uma <i>feature</i> filha deve ser selecionada	
	Ou	Se a <i>feature</i> pai é selecionada, pelos uma da <i>features</i> filha deve ser selecionada.	

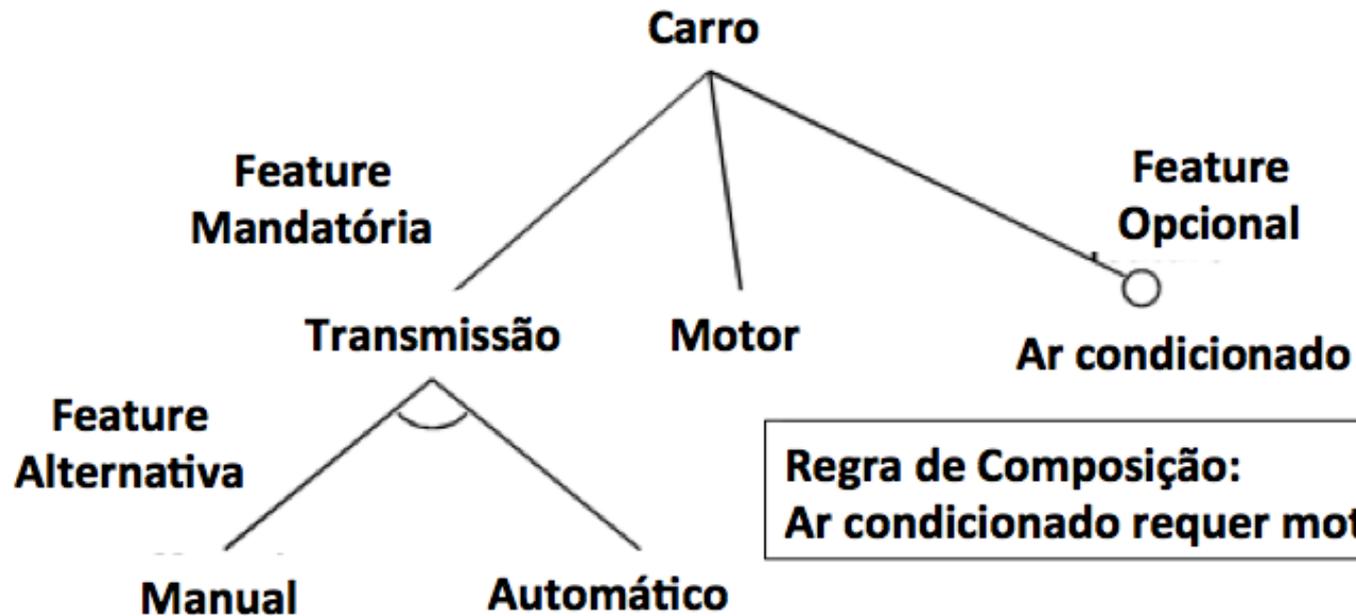
Modelo de LPS

- *Feature Model*

Relacionamento	Tipo	Semântica	Notação
Dependência	Implicação	Se uma <i>feature</i> é selecionada, a <i>feature</i> implicada deve ser selecionada, ignorando a sua posição na árvore de funcionalidade	
	Exclusão	Indica que ambas as <i>feature</i> que não podem ser selecionadas na mesma configuração de produto e que são mutuamente exclusivas	

Modelo de LPS

- *Feature Model*



Lógica:
Transmissão manual mais eficiente

Regra de Composição:
Ar condicionado requer motor > 1.0

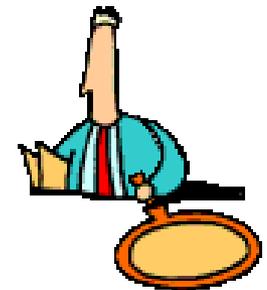
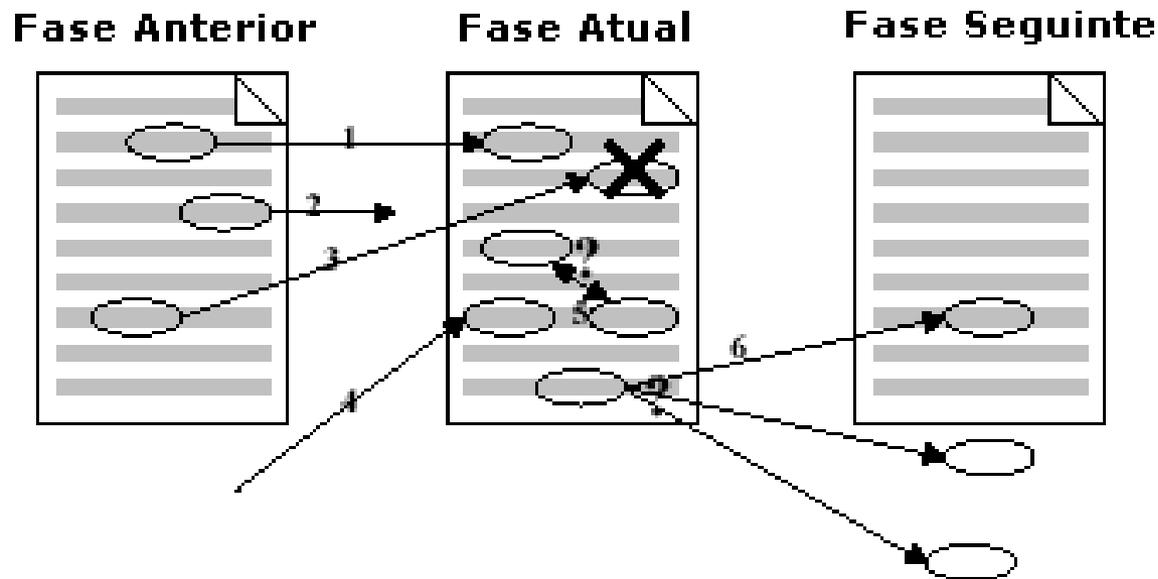
Modelo de LPS

- Matriz de Produtos

	Modelo 1	Modelo 2	Modelo 3	Modelo 4
Transmissão Manual		X		X
Transmissão Automática	X		X	
Motor	X	X	X	X
Ar condicionado			X	X

Objetivos

- Inspeção para:
 - Assegurar a **qualidade de modelos** de linha de produto de software



Objetivo Geral da Pesquisa

Criação de uma técnica de inspeção em modelos de linha de produto para melhoria de qualidade de software

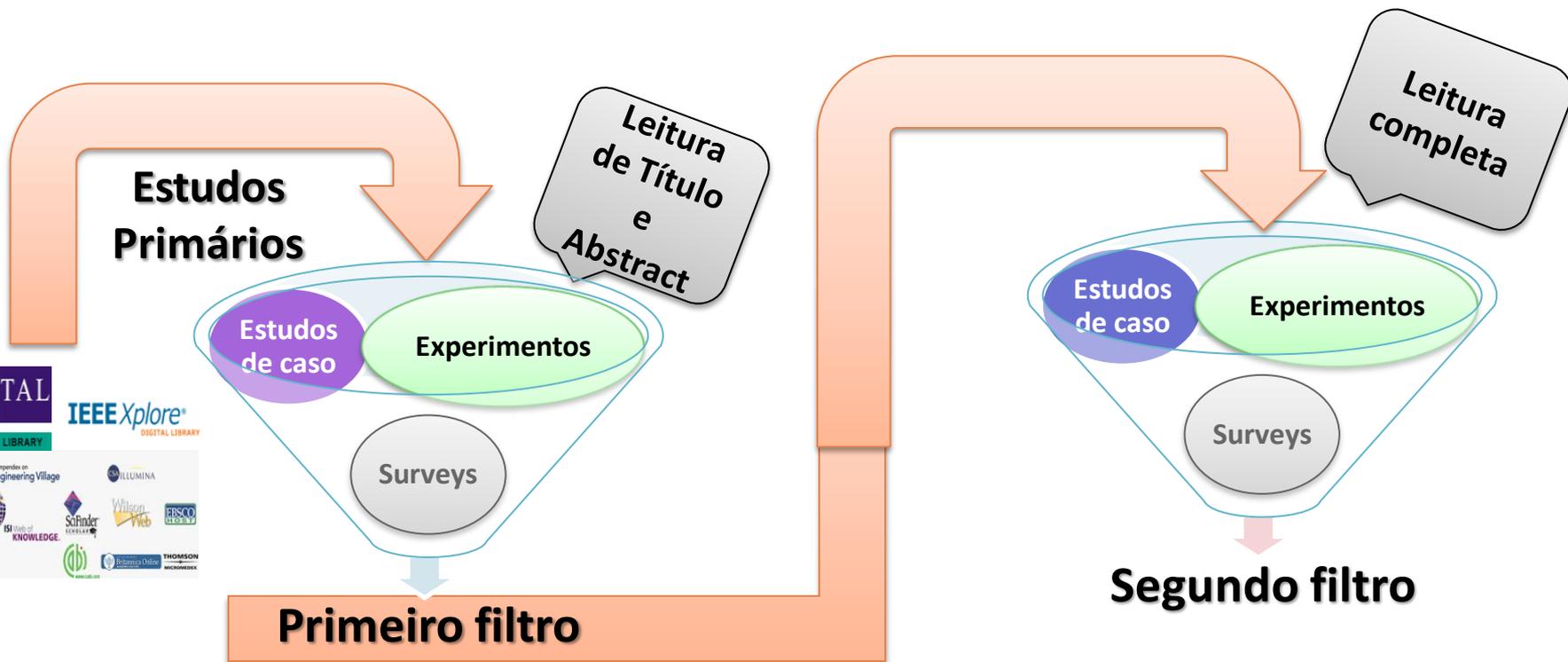


Revisão Sistemática

- Caracterização das técnicas de inspeção em modelos de linha de produto de software que foram apresentadas/propostas.

Analisar	Publicações científicas através de um estudo baseado em mapeamento sistemático
Com o propósito de	Caracterizar técnicas de inspeção de modelos de Linha de Produto de Software
Com relação a	Garantia da qualidade do produto da Linha de Produto de Software
Do ponto de vista dos	Pesquisadores
No Contexto	Acadêmico e industrial com foco em inspeção de modelos de Linha de Produto de Software

Revisão Sistemática



Bibliotecas selecionadas	Artigos Retornados	Artigos selecionados no 1º filtro
ACM DL	482	28
IEEE Xplorer	88	22
Scopus	271	40

Bibliotecas selecionadas	Artigos aceitos no 1º filtro	Artigos aceitos no 2º Filtro
ACM DL	28	10
IEEE Xplorer	22	05
Scopus	40	14

Artigos Selecionados

Técnicas de Inspeção em Modelos de Linha de Produto de Software

- Modelos a serem inspecionados:
 - **Matriz de Subproduto**
 - **Modelo de Features**
- Documento que deve ser comparado com os modelos:
 - **Requisitos**

Técnica 1

Requisitos x Matriz de Subproduto

- Defeitos
 - **Redundância**
 - Gera Produtos com as mesmas features
 - **Inconsistência**
 - Quando informações não correspondem ao Documento de Requisitos
- Saída
 - Formulário de Relato de Defeitos



Técnica 1

Requisitos x Matriz de Subproduto

Tipo de Defeito	Item
Redundância	1. Existem dois ou mais produtos na matriz de subproduto que apresentam o mesmo conjunto de features?
Inconsistência	2. Existem features encontradas no documento de requisitos que não estão contidas na matriz de subproduto?
	3. Existem features em um produto que pode ser gerado pela matriz de subprodutos que não estão descritos no documento de requisitos?
	4. Existem features em um produto que não pode ser gerado pela matriz de subproduto que estão descritos no documento de requisito?
	5. Existe alguma feature caracterizada como obrigatória no documento de requisito que não está em todos os produtos da matriz de subproduto?

Técnica 2

Requisitos x Matriz de Subproduto x Modelo de Features

- Defeitos
 - Inconsistência
 - Artefatos incluem informações contraditórias entre si
- Saída
 - Formulário de Relato de Defeitos



Técnica 2

Requisitos x Matriz de Subproduto x Modelo de Features

Tipo de Defeito	Item
Inconsistência	1. Existe algum produto da matriz de subproduto que não pode ser gerado pelo modelo de <i>features</i> ?
	2. Existe algum relacionamento de implicação descrito no documento de requisitos que não está especificado no modelo de <i>features</i> ?
	3. Existe algum relacionamento de implicação que está especificado no modelo de features mas não está descrito no documento de requisitos ?
	4. Existe alguma relacionamento mutuamente exclusivo que não está especificado no modelo de <i>features</i> ?
	5. Existe algum relacionamento mutuamente exclusivo que está especificado no modelo de features mas não está descrito no documento de requisitos ?

Técnica 2

Requisitos x Matriz de Subproduto x Modelo de Features

Tipo de Defeito	Item
Inconsistência	6. Existe alguma feature obrigatória na matriz de subproduto e documento de requisitos que não está especificada como mandatória no modelo de features?
	7. Existe alguma feature opcional na matriz de subproduto e documento de requisitos que não está especificada como opcional no modelo de features?
	8. Existe algum conjunto de features alternativas na matriz de subproduto e documentos de requisitos que não está especificado como alternativa no modelo de features?
	9. Existe algum conjunto de features “ou” na matriz de subproduto e documentos de requisitos que não está especificada como “ou” no modelo de features?
	10. Existe algum relacionamento pai-filho entre features que descrito no documento de requisitos e que não está especificado no modelo de features?

Técnica 3

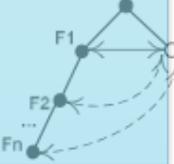
Modelo de Features

- Defeitos
 - **Redundância**
 - Qualquer informação semântica é modelada de mais de uma maneira
 - **Anomalia**
 - Perda de possível configuração devido a restrição da variabilidade ao utilizar features obrigatórias
 - **Inconsistência**
 - Inclui informações contraditórias
- Saída
 - Formulário de Relato de Defeitos



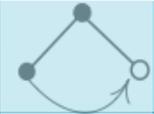
Técnica 3

Modelo de Features

Tipo de Defeito	Item	
Redundância	1. Existe uma feature completamente obrigatória que apresenta uma implicação ?	
	2. Existe uma feature F2 relativamente obrigatória para uma feature F1 e que apresenta uma implicação ?	
	3. Existem duas features filhas de uma mesma feature pai que são ao mesmo tempo alternativa e mutuamente exclusiva ?	
	4. Existe um feature com relacionamento de implicação por múltiplas features F1, ..., Fn , sendo que F1 é pai de F2 e que F2,...,FN são relativamente obrigatórias para F1?	
	5. Existe um feature com relacionamento mutuamente exclusivo por múltiplas features F1, ..., Fn, sendo que F1 é pai de F2 e que F2,...,FN são relativamente obrigatórias para F1?	
	6. Existe uma feature F1 que tem um relacionamento de implicação para F2 que tem uma implicação para F3?	

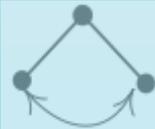
Técnica 3

Modelo de Features

Tipo de Defeito	Item	
Anomalia	7. Existe uma feature opcional que tem um relacionamento de implicação por uma feature completamente obrigatória ?	
	8. Existe uma feature que alternativa em relação a feature pai e que tem um relacionamento de implicação por uma feature completamente obrigatória ?	
	9. Existe uma feature que "ou" em relação a feature pai e que tem um relacionamento de implicação por uma feature completamente obrigatória ?	
	10. Existe uma feature opcional que tem um relacionamento mutualmente exclusivo com uma feature completamente obrigatória ?	
	11. Existe uma feature alternativa que tem um relacionamento mutualmente exclusivo com uma feature completamente obrigatória ?	
	12. Existe uma feature "ou" que tem um relacionamento mutuamente exclusivo com uma feature completamente obrigatória ?	

Técnica 3

Modelo de Features

Tipo de Defeito	Item	
Inconsistência	13. Existem duas features com relacionamento mutualmente exclusivo e que são completamente obrigatórias ?	
	14. Existe uma feature relativamente obrigatória para a feature F1 e estas features apresentam um relacionamento mutualmente exclusivo ?	
	15. Existem duas features filhas alternativas e que apresentam um relacionamento de implicação ?	
	16. Existem duas features que apresentam simultaneamente um relacionamento de implicação e mutualmente exclusivos ?	

Cronograma

Atividade	2011											2012		
	F e v	M a r	A b r	M a i	J u n	J u l	A g o	S e t	O t	N o v	D e z	J a n	F e v	M a r
Revisão Bibliográfica	■													
Revisão Sistemática	■	■	■											
Elaboração da Primeira versão da Técnica de Inspeção			■	■	■									
Estudo de Viabilidade					■	■								
Análise dos Resultados							■	■						
Elaboração de Artigos								■	■					
Elaboração da Segunda Versão da Técnica de Inspeção								■	■					
Estudo de Observação									■	■				
Análise de Resultados										■	■			
Relatório												■	■	



Perguntas

Referência Bibliográfica

- [ACKERMAN et al., 1989] ACKERMAN, A., BUCHWALD, L., LEWSKI, F., 1989, “Software Inspections: An Effective Verification Process”, IEEE Software, vol. 6, no. 3, pp.31-37.
- [ATKINSON et al., 2001] ATKINSON, C. BAYER, J., BUNSE, C., KAMSTIES, E., LAITENBERGER, O., LAQUA, R., MUTHING, D., PAECH, B., WÜST, J., ZETTEL, J. Component-Based Product-Line Engineering with UML. Boston: Addison-Wesley, 2001.
- [BAYER et al., 1999] BAYER, J.; FLEGE, O.; KNAUBER, P.; LAQUA, R.; SCHMID, K.; WIDEN, T.; DEBAUD, J. PuLSE: a methodology to develop software product lines. In: SYMPOSIUM ON SOFTWARE REUSABILITY, 5., 1999, Los Angeles. Proceedings...Los Angeles, 1999. p. 122-131.
- [BOSCH, 2000] BOSCH, J. Design & use of software architectures: adopting and evolving a product-line approach. Boston: Addison Wesley, 2000.
- [CLEMENS e NORTHROP, 2001] CLEMENS, P.; NORTHROP, L. Software product lines: practices and patterns. 1. ed. Boston: Addison-Wesley, 2001.
- [GOMAA, 2005] GOMAA, H. Designing software product lines with UML: from use cases to pattern-based software architectures. Boston: Addison-Wesley, 2005.
- [HEYMANS; TRIGAUX, 2003] HEYMANS, P.; TRIGAUX, J. C. Software product line: state of the art. Technical report for PLENTY project, Institut d’Informatique FUNDP, Namur, 2003.
- [INCT-SEC, 2010] Instituto Nacional de Ciência e Tecnologia em Sistemas Embarcados Críticos (INCT-SEC). Disponível em <http://www.inct-sec.org/> (22/10/2010)

Referência Bibliográfica

- [KANG, 1990] KANG, K. Feature-oriented domain analysis (FODA) - feasibility study. Technical Report CMU/SEI-90-TR-21, SEI/CMU, Pittsburgh, 1990.
- [MAFRA et al., 2006] MAFRA, S., BARCELOS, R., TRAVASSOS, G. H., 2006. "Aplicando uma Metodologia Baseada em Evidência na Definição de Novas Tecnologias de Software". In: Proceedings of the 20th Brazilian Symposium on Software Engineering (SBES 2006), v. 1, pp. 239 – 254, Florianopolis. October.
- [MCGREGOR, 2001] McGregor, J.D. Testing a Software Product Line. Technical Report CMU/SEI-2001-TR-022, Carnegie Mellon University, SEI (2001)
- [RAMAIAH et al., 2008] RAMAIAH, P. S.; SWARUP, M. B.; KUMAR, K. R.; Conceptual Modeling for Safety Critical Computer Systems, 9th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, p. 814-819, 2008.
- [SEI, 2007] Software Engineering Institute. A Framework for Software Product Line Practice, Version 5.0. Disponível em: <http://www.sei.cmu.edu/productlines/framework.html> (acessado em 19/09/2008).
- [SPC, 1993] SPC - SOFTWARE PRODUCTIVITY CONSORTIUM. Reuse-Driven Software Processes Guidebook. SPC-92019-CMC version 02.00.03 November 1993.
- [SHULL et al., 2001] Shull, F., Carver, J., Travassos, G. (2001) "An Empirical Methodology for Introducing Software Processes", In: Proceedings of the Joint 8th European Software Engineering Conference (ESEC) and 9th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE-9), pp. 288-296.
- [WEISS e CHI TAU, 1999] WEISS, D.; CHI TAU, R. L. Software product-line engineering: a family-based software development process. Boston: Addison-Wesley, 1999.