

Avaliação da Qualidade do Modelo de *Features* em Linhas de Produtos de Software Utilizando Medidas

Carla I. M. Bezerra^{1,2}, José Maria Monteiro¹, Rossana M. C. Andrade^{1,*}

¹ Universidade Federal do Ceará (UFC), ¹ Mestrado e Doutorado em Ciência da Computação (MDCC), Departamento de Computação (DC), Grupo de Redes, Engenharia de Software e Sistemas (GREat), Fortaleza, CE – Brasil

² Campus Quixadá, CE – Brasil

carlailane@ufc.br, {monteiro, rossana}@lia.ufc.br

Abstract. *In Software Product Lines (SPL), the quality evaluation is a critical factor, once an error in the line can spread to various other products. While a features model is one of the main artifacts in an SPL, one of the current strategies to assess the quality consists in using some measures to this evaluation. So, this work focuses on proposing a set of measures to evaluate the quality of the features model of an SPL. First, the measures are identified from a broad bibliographic review and, then, their detailed specification is proposed. These measures are then used to evaluate the quality of the features model of Mobline, a SPL for mobile and context aware applications.*

Resumo. *Em Linhas de Produto de Software (LPS), a avaliação da qualidade é um fator crítico, pois um erro na linha pode se propagar para diversos produtos. Enquanto o modelo de features é um dos principais artefatos de uma LPS, uma das estratégias existentes para avaliar a qualidade consiste em utilizar medidas para essa avaliação. Sendo assim, este trabalho tem como objetivo propor um conjunto de medidas para avaliação da qualidade do modelo de features de uma LPS. Inicialmente, as medidas são identificadas a partir de uma ampla revisão bibliográfica e, em seguida, uma especificação detalhada das mesmas é proposta. Essas medidas são então utilizadas para avaliar a qualidade do modelo de features do Mobline, uma LPS voltada para aplicações móveis e sensíveis ao contexto.*

1. Introdução

A avaliação da qualidade em Linhas de Produtos de Software (LPS) apresenta uma complexidade maior do que no desenvolvimento de software tradicional. Este fato decorre de dois aspectos principais: i) produtos diferentes, em uma mesma LPS, podem requerer níveis de qualidade distintos, e ii) diversos produtos podem ser derivados de uma mesma LPS. Neste contexto, a avaliação da qualidade de todos os produtos de software de uma determinada linha mostra-se impraticável, tanto por razões econômicas quanto pelo esforço envolvido. Desta forma, novas estratégias capazes de reduzir o custo e o esforço despendidos na avaliação da qualidade em LPSs são necessárias [Etzeberria e Sagardui 2008a].

* Bolsista do CNPq de Produtividade DT-2 com o número de processo 314021/2009-4

Uma estratégia que pode ser utilizada para reduzir os custos associados à garantia da qualidade das linhas de produtos de software consiste em realizar avaliações de qualidade nas fases iniciais da LPS, com o intuito de evitar que os erros se propaguem para outros artefatos da linha ou para fases posteriores.

Um dos principais artefatos elaborados nas fases iniciais de uma LPS é o modelo de *features*, que é um artefato elaborado na fase de engenharia de domínio de uma LPS, sendo utilizado para modelar as partes comuns e variáveis entre os produtos que compõem uma família de produtos [Kang et al. 1990]. Segundo Tessier et al. (2005), *feature* é um conceito ou propriedade relacionada com um requisito de uso visível, funcional ou não funcional. Assim, as *features* podem capturar aspectos comuns ou, eventualmente, variabilidades entre aplicações de um mesmo domínio.

Como os modelos de *features* podem ser utilizados como uma plataforma para derivar muitas aplicações a partir de uma linha de produto de software, sua qualidade influencia as propriedades finais das aplicações desenvolvidas. Portanto, é importante assegurar a qualidade dos modelos de *features* já a partir dos estágios iniciais da linha. Por exemplo, Mello et al. (2012) utilizam técnicas de inspeção para a identificação de erros no modelo de *features*.

Montagud e Abrahão (2009) realizaram uma revisão sistemática com o objetivo de explorar modelos de avaliação de qualidade em LPS. Nesta revisão, diversos trabalhos foram identificados. No entanto, a maioria desses trabalhos concentra-se na avaliação dos atributos de qualidade no nível de arquitetura [Kim et al. 2008; Matinlassi et al. 2002; Olumofin e Mišic 2007; Bosch 2000, Benavides et al. 2007, Thiel 2002], já outros tratam da avaliação dos atributos do domínio relevante [Etxeberria e Sagardui 2008a; 2008b]. Entretanto, nenhum trabalho voltado para a qualidade do modelo de *features* foi encontrado.

Na revisão bibliográfica apresentada por este artigo, foram encontrados trabalhos que discutem medidas direcionadas para a avaliação da qualidade do modelo de *features* em Linhas de Produtos de Software [Mendonça et al. 2009; Bagheri e Gasevic 2011]. No entanto, esses trabalhos não consolidam um conjunto de medidas para avaliação do modelo de *features*, não são baseados em características e atributos de qualidade relevantes no contexto da avaliação da qualidade do modelo de *features* em LPSs, e não fornecem uma especificação das medidas de forma a guiar a coleta e análise.

Sendo assim, este trabalho identifica e especifica um conjunto de medidas importantes para avaliação da qualidade do modelo de *features* de uma linha de produto de software. Essas medidas foram identificadas a partir da ampla revisão bibliográfica citada anteriormente. Adicionalmente, todas as medidas foram especificadas detalhadamente. Em seguida, para cada medida, identificou-se a característica de qualidade e o atributo de qualidade aos quais a medida estava relacionada. Por exemplo, a medida Número de *Features* está relacionada à característica de qualidade Manutenibilidade e ao atributo de qualidade Tamanho do modelo de *features*. Para validar a solução proposta, as medidas identificadas são utilizadas para avaliar a qualidade do modelo de *features* do Mobiline, uma LPS voltada para aplicações móveis e sensíveis ao contexto [Mobiline 2012].

O restante deste artigo está organizado da seguinte forma: na Seção 2 são apresentados conceitos sobre avaliação de qualidade em LPS; na Seção 3 são apresentadas as medidas identificadas em um estudo para avaliar o modelo de *features* de uma LPS; na Seção 4 é apresentada a aplicação e análise das medidas em uma LPS de Guias de Visitas Móveis; na Seção 5 são apresentados os trabalhos relacionados; e na Seção 6 são apresentadas as conclusões e direções para trabalhos futuros.

2. Avaliação de Qualidade em Linhas de Produtos de Software

A avaliação da qualidade no contexto de linhas de produto de software é essencial, uma vez que um erro ou uma incompatibilidade em um ativo reutilizável pode ser propagado para um lote de produtos. Neste sentido, a avaliação da qualidade em LPSs possui alguns desafios que não estão presentes no desenvolvimento de software tradicional [Etxeberria e Sagardui 2008a].

Neste sentido, tanto no desenvolvimento de software tradicional, quanto no desenvolvimento em linhas de produtos de software, a qualidade tem sido um fator crítico. Diversos modelos e normas de qualidade têm focado na qualidade dos produtos de software e pode-se encontrar na literatura algumas definições de qualidade. A NBR ISO 8402 (1994) define qualidade em termos da habilidade em satisfazer necessidades explícitas e implícitas. Já a norma do Instituto de Engenharia Elétrica e Eletrônica [IEEE 1998] define a qualidade do software qualidade de software como o grau em que o software possui uma combinação desejada de atributos. Os atributos são características físicas ou mensuráveis de uma entidade.

A avaliação de produtos de software com o objetivo de satisfazer as necessidades de qualidade de software é uma das etapas do ciclo de vida de desenvolvimento de software. A qualidade do produto de software pode ser avaliada medindo-se os atributos internos (tipicamente medidas estáticas de produtos intermediários), os atributos externos (tipicamente pela medição do comportamento do código quando executado) ou os atributos de qualidade em uso. O objetivo é que o produto tenha o efeito requerido num contexto de uso particular [ISO/IEC 9126 2003].

É recomendado que, para a avaliação de qualidade de um produto de software, seja definido um modelo de qualidade e que este modelo de qualidade seja usado na definição das metas de qualidade para os produtos de software final e intermediários [ISO/IEC 9126 2003].

Em relação ao estabelecimento de medidas para a avaliação da qualidade, o padrão IEEE (1998) define uma metodologia para estabelecer os requisitos de qualidade, além de identificar, implementar, analisar e validar processos e medidas de qualidade do produto de software. As medidas são essenciais na avaliação da qualidade de um produto, e estas podem compor um modelo de qualidade. Segundo a norma SQuaRE [ISO/IEC 25000 2005], um modelo de qualidade é composto de características de qualidade, subcaracterísticas e atributos. Característica de qualidade é um conjunto de propriedades de um produto de software pela qual sua qualidade pode ser definida e avaliada [ISO/IEC 9126 2001]. Uma característica pode ser dividida em várias subcaracterísticas. Atributos são propriedades mensuráveis, físicas ou abstratas, de uma entidade. As medidas de qualidade são utilizadas para indicar as características e/ou subcaracterísticas de qualidade em um produto de software. Como por exemplo, a característica de qualidade Manutenibilidade, é dividida em duas subcaracterísticas que

são Modularidade e Reusabilidade. O atributo de qualidade relacionado à subcaracterística Modularidade é a Modularidade de uma LPS. A medida relacionada à Modularidade é a Taxa de ativos por *features*.

Em relação a técnicas de avaliação de qualidade para Linhas de Produto de Software, Etxeberria e Sagardui (2005) classificam essas técnicas em dois grupos:

- Técnicas de Questionários para Avaliação Qualitativa: incluem cenários, questionários e *checklists*. Podem ser usadas para avaliar a qualidade de desenvolvimento ou operacional;
- Técnicas de Medição para Avaliação Quantitativa: incluem simulação, prototipagem, modelos matemáticos e experimentais. Podem ser usadas para medir a aplicação de técnicas que englobam qualidades específicas, normalmente operacionais.

Segundo Etxeberria e Sagardui (2008a), a avaliação de qualidade pode ser realizada em diferentes fases: nas fases iniciais, relacionadas à engenharia de domínio, como por exemplo, a fase de *design* (métodos de avaliação de arquitetura de software), ou em fases relacionadas à engenharia de aplicação (métodos de medição de qualidade). Para reduzir custos e aumentar a qualidade dos outros artefatos e dos produtos da linha, a avaliação de qualidade deveria ser aplicada em fases iniciais da linha, na Engenharia de Domínio. Muitos trabalhos têm investigado a avaliação de qualidade da arquitetura da LPS, mas pouco se tem pesquisado acerca da avaliação da qualidade do modelo de *features*.

No contexto da revisão bibliográfica realizada neste trabalho, não foram encontrados modelos de qualidade que focassem na avaliação da qualidade do modelo de *features*. Apenas iniciativas individuais de elaboração de medidas voltadas para a avaliação de alguns aspectos do modelo de *features*, sem ter um modelo formal de apoio [Montagud and Abrahão 2011; Bagheri e Gasevic 2011], foram observadas.

Nesse sentido, o presente trabalho tem por objetivo fornecer um conjunto de medidas voltadas para a avaliação do modelo de *features* em LPSs. Essas medidas foram extraídas a partir de diversos trabalhos encontrados na literatura. Adicionalmente, essas medidas foram enriquecidas, adaptadas e detalhadas. Posteriormente, cada medida é associada às características de qualidade e a um ou mais atributos de qualidade. Assim, este trabalho constitui um guia sistemático para a utilização de medidas na avaliação da qualidade do modelo de *features* em LPSs. Esse guia envolve não somente a identificação das medidas que devem ser utilizadas, mas também uma descrição detalhada das medidas, orientações para a coleta e instruções para a análise e interpretação dos resultados.

3. Avaliação da Qualidade do Modelo de *Features* em LPS

Avaliar a qualidade de LPSs em estágios iniciais, evita que os erros sejam propagados para os artefatos da linha de forma a evitar defeitos nos produtos gerados. O modelo de *features* é um dos principais artefatos elaborados na fase de engenharia de domínio e sua qualidade influencia diretamente a qualidade dos produtos derivados da LPS. Um dos mecanismos de avaliação que podem ser utilizados para avaliar a qualidade de um determinado artefato consiste no uso de medidas. Contudo, as medidas devem ser definidas de acordo com o artefato que será avaliado. Neste caso, para avaliar a

qualidade do artefato modelo de *features* um conjunto particular de medidas deve ser utilizado.

Neste sentido, este trabalho propõe um conjunto de medidas voltado para a avaliação da qualidade do modelo de *features* em LPSs. Para se chegar a essas medidas, foi realizada uma ampla revisão na literatura, de forma não sistemática. A busca dos artigos foi realizada na biblioteca IEEE Explorer², Springer³ (apenas periódicos) e ACM DL Library⁴. Adicionalmente, foram realizadas buscas no Google Scholar⁵ visando encontrar outros artigos ou relatórios técnicos relacionados ao tema.

Nas buscas realizadas foram encontrados diversos trabalhos relacionados a avaliação de arquitetura, onde apenas alguns possuíam medidas relacionadas à avaliação da qualidade deste artefato [Oliveira et al. 2008; Zhang et al. 2008]. Diferentes artigos voltados para a avaliação das características de qualidade relacionadas à configuração de produtos [Villela et al. 2012; Tawhid e Petriu 2011; Bartholdt et al. 2009] também foram observados. Contudo, apenas três trabalhos envolvendo medidas relacionadas à avaliação do modelo de *features* foram identificados [Montagud e Abrahão 2009b; Bagheri e Gasevic 2011; Mendonça et al. 2009].

Na próxima subseção são apresentadas as medidas, extraídas da literatura, para a avaliação do modelo de *features* em Linhas de Produtos de Software, bem como as outras contribuições deste artigo, resumidas a seguir: (i) é importante ressaltar que além da extração das medidas, a partir dos artigos encontrados na revisão bibliográfica, este trabalho propõe melhorias nas medidas identificadas, tais como a identificação das características e dos atributos de qualidade de cada medida; (ii) além disso, uma especificação detalhada das medidas é proposta, a qual envolve, para cada medida, a descrição, fórmula de cálculo, unidade de medida e metas a serem alcançadas; e (iii) adicionalmente, instruções para a coleta das medidas e análise dos resultados são fornecidas, o que resulta em um guia para a utilização de medidas na avaliação da qualidade do modelo de *features* em LPSs.

3.1. Especificação de Medidas para Avaliação do Modelo de *Features*

A partir da revisão bibliográfica realizada foram extraídas 15 medidas voltadas para a avaliação da qualidade do modelo de *features*. Das 15 medidas identificadas, 3 possuíam apenas uma breve descrição (MED_2, MED_14 e MED_15), 10 possuíam descrição e a característica de qualidade a qual estavam associadas (MED_4 a MED_13) e 2 possuíam além de sua descrição a característica e os atributos de qualidade à qual estavam relacionadas (MED_1 e MED_3).

Desta forma, inicialmente, para cada medida selecionada, identificamos as características de qualidade e os atributos de qualidade a ela relacionados. A Tabela 1 ilustra a associação entre cada medida e suas características e atributos de qualidade. Vale destacar que as medidas identificadas na Tabela 1 podem ser utilizadas para capturar a complexidade da LPS, o seu tamanho, sua variabilidade, além da consistência e da correteza do modelo de *features*.

² <http://http://ieeexplore.ieee.org>

³ <http://link.springer.com/>

⁴ <http://http://dl.acm.org>

⁵ <http://scholar.google.com/>

Tabela 1: Características e atributos de qualidade das medidas identificadas para avaliação da qualidade do modelo de *features*.

Referência	ID da Medida	Medida	Característica de qualidade	Atributo de Qualidade
[Montagud e Abrahão 2009b]	MED_01	Número de variabilidades em LPS	Manutenibilidade	Modularidade da LPS
[Mendonça et al. 2009]	MED_02	<i>Features</i> Comuns	Manutenibilidade	Consistência da LPS
[Montagud e Abrahão 2009b] (adaptada)	MED_03	Número de pontos de variação incluídos no âmbito da LPS	Manutenibilidade	Reusabilidade da LPS
[Mendonça et al. 2009; Bagheri e Gasevic 2011]	MED_04	Número de <i>Features</i> (NF)	Manutenibilidade	Tamanho do modelo de <i>features</i>
[Bagheri e Gasevic 2011]	MED_05	Taxa de Variabilidade	Manutenibilidade	Tamanho do modelo de <i>features</i>
[Bagheri e Gasevic 2011]	MED_06	Número de <i>Features</i> Top	Manutenibilidade	Tamanho do modelo de <i>features</i>
[Bagheri e Gasevic 2011]	MED_07	Número de <i>Features</i> Leaf	Manutenibilidade	Tamanho do modelo de <i>features</i>
[Bagheri e Gasevic 2011; Mendonça et al. 2009]	MED_08	Número de Configurações Válidas	Manutenibilidade	Validade das configurações
[Bagheri e Gasevic 2011]	MED_09	Flexibilidade da Configuração	Manutenibilidade	Flexibilidade da Configuração
[Bagheri e Gasevic 2011; Mendonça et al. 2009]	MED_10	Profundidade da árvore	Manutenibilidade	Profundidade da árvore
[Bagheri e Gasevic 2011; Mendonça et al. 2009]	MED_11	Complexidade Ciclomática	Manutenibilidade	Complexidade Ciclomática
[Bagheri e Gasevic 2011]	MED_12	Coefficiente de Conectividade	Manutenibilidade	Conectividade entre os componentes do modelo de feature
[Bagheri e Gasevic 2011; Mendonça et al. 2009]	MED_13	Cross-tree Constraints	Manutenibilidade	Grau de envolvimento das <i>features</i> na definição das restrições de integridade
[Mendonça et al. 2009]	MED_14	Número de <i>Features</i> Mortas	Manutenibilidade	Corretude da LPS
[Mendonça et al. 2009]	MED_15	Densidade CTC	Manutenibilidade	Corretude da LPS

Posteriormente, para cada medida identificada, foram detalhadas a sua descrição e a sua fórmula de cálculo (Tabela 2). Para isso, as medidas selecionadas foram estendidas e adaptadas. A seguir, destacamos algumas das adaptações realizadas. A medida Número de Pontos de Variação incluídos no âmbito da LPS (MED_03) foi adaptada da medida original. A medida original calculava os pontos de variação dos casos de uso, e não apenas no modelo de *features*. Essa medida foi adaptada para que fosse relacionada apenas ao Modelo de *Features*. As medidas (MED_02, MED_14 e MED_15) não apresentavam nenhuma especificação das medidas. As medidas MED_01 e MED_03 possuíam uma especificação bem definida, no entanto, poucas medidas foram identificadas para avaliação do modelo de *features*. O restante das medidas extraídas apresenta nos trabalhos apenas as descrições das medidas, sendo necessário definir as especificações de todas as medidas identificadas. A Tabela 2 ilustra o trabalho de especificações da fórmula de cálculo de cada medida.

Tabela 2: Conjunto de Medidas identificadas para avaliar a qualidade do modelo de *features* de uma LPS.

ID da Medida	Medida	Descrição	Fórmula de Cálculo
MED_01	Número de variabilidades em LPS	Taxa do Número de <i>features</i> alternativas ou opcionais frente ao número total de <i>features</i> .	$V = A + B/N$ A - Número de <i>features</i> alternativas B - Número de <i>features</i> opcionais N - Número de <i>features</i> totais
MED_02	<i>Features</i> Comuns	<i>Features</i> que são incluídas em todas as configurações válidas, são as <i>features</i> comuns a todas as configurações.	$CF = \sum$ (Número de <i>features</i> que são incluídas em todas as configurações válidas)
MED_03	Número de pontos de variação incluídos	Medir quantos pontos de variação são incluídos no âmbito da LPS.	$PV = \sum$ (Número de pontos de variação incluídos no âmbito da LPS)
MED_04	Número de <i>features</i> (NF)	O número total de <i>features</i> (NF) que estão presentes em modelo de <i>features</i> . Isto inclui tanto as folhas do modelo e as <i>features</i> principais. O número de <i>features</i> conta todos os nós na árvore de modelo de <i>features</i> .	$NF = \sum$ (Número de <i>features</i> do modelo de <i>features</i>)
MED_05	Taxa de variabilidade	O fator médio de ramificação que a <i>feature</i> pai apresenta no modelo de <i>features</i> . Em outras palavras, o número médio de filhos dos nós na árvore do	$TV = \sum$ (Número médio de filhos dos nós no modelo de <i>features</i>)
MED_06	Número de <i>features</i> Top	O número de <i>features</i> que são descendentes diretos da primeira raiz do modelo de <i>features</i> . Em outras palavras, o número de nós em profundidade de uma árvore.	$NTop = \sum$ (Número de nós descendentes da raiz do modelo de <i>features</i>)
MED_07	Número de <i>features</i> Leaf	O número de <i>features</i> sem filhos ou especializações adicionais. Estas correspondem com as folhas da árvore no modelo de <i>features</i> .	$NLeaf = \sum$ (Número de <i>features</i> sem filhos do modelo de <i>features</i>)
MED_08	Número de configurações válidas	O número de todas as configurações possíveis e válidas que podem ser derivadas a partir do modelo de <i>feature</i> em face de restrições a sua integridade e estrutura de árvore.	$NCV = \sum$ (Número de configurações possíveis e válidas do modelo de <i>feature</i>)
MED_09	Flexibilidade da configuração	Esta é a razão entre o número de funcionalidades opcionais ao longo de todos as <i>features</i> disponíveis no modelo de <i>feature</i> . A lógica por trás disso é que as <i>features</i> mais opcionais existem no modelo de <i>feature</i> , as escolhas mais estão disponíveis para os designers para escolher durante a configuração do modelo de <i>feature</i> .	$FoC = NFO/NF$ NFO - Número de <i>features</i> opcionais NF - Número de <i>features</i>
MED_10	Profundidade da árvore	O comprimento do caminho mais longo a partir da raiz do modelo de <i>features</i> até a <i>feature</i> folha do modelo de <i>features</i> .	$PA = \sum$ (Número de <i>features</i> do maior caminho a partir da raiz do modelo de <i>features</i> até a <i>feature</i> folha do modelo de <i>features</i>)
MED_11	Complexidade ciclomática	O número de ciclos diferentes que podem ser encontrados no modelo de <i>features</i> . Uma vez que os modelos de <i>features</i> são sob a forma de árvores, sem ciclos podem existir em um modelo de <i>features</i> , no entanto, as restrições de integridade entre os <i>features</i> disponíveis podem causar ciclos. É simples mostrar que o número de ciclos distintos de complexidade e, portanto, de um modelo de ciclomática funcionalidade é equivalente ao número de restrições de integridade de um modelo de <i>features</i> .	$CC = \sum$ (número de restrições de integridade de um modelo de <i>features</i>)

Continuação da Tabela 2: Conjunto de Medidas identificadas para avaliar a qualidade do modelo de *features* de uma LPS.

ID da Medida	Medida	Descrição	Fórmula de Cálculo
MED_12	Coefficiente de conectividade	A razão entre o número de arestas mais o número de <i>features</i> , em um modelo de <i>features</i> . Em teoria, o gráfico, o coeficiente de conectividade representa o quão bem os componentes estão ligados graficamente.	$CoC = NA/NF$ NA - Número de arestas NF - Número de <i>features</i>
MED_13	Cross-tree Constraints	A razão entre o número de <i>features</i> únicas envolvidas na restrição de integridade do modelo de <i>features</i> sobre o total do número de <i>features</i> do modelo de <i>features</i> . Essa medida representa o grau de	$CTC = NFRI / NF$ NFRI - número de <i>features</i> únicas envolvidas na restrição de integridade do modelo de <i>features</i> NF - Número de <i>Features</i>
MED_14	Número de <i>Features</i> Mortas	Somatório de <i>features</i> mortas, aquelas que não são passíveis de utilização no modelo de <i>features</i> .	$FM = \sum (\text{número de } features \text{ mortas})$
MED_15	Densidade CTC	Razão entre o número de restrições e o número de variáveis nas restrições.	$DCTC = NR / NRV$ NR - Número de restrições CC - Número de restrições variáveis (complexidade ciclomática)

4. Aplicação das Medidas para Avaliação do Modelo de *Features*

Para validar a proposta deste artigo, as medidas identificadas foram utilizadas para avaliar a qualidade do modelo de *features* do Mobiline, uma LPS voltada para aplicações móveis e sensíveis ao contexto [Mobiline 2012]. Para isso, apresentaremos, inicialmente, nesta seção, a LPS Mobiline, destacando o seu modelo de *features* (Figura 1). Em seguida, discutiremos a coleta e a análise das medidas utilizadas para avaliar a qualidade deste artefato. Por fim, são discutidas as vantagens e desvantagens da utilização da solução proposta.

4.1. Linha de Produto de Software Mobiline

O Mobiline é uma Linha de Produto de Software Sensível ao Contexto (LPSSC) para guias de visita móveis e sensíveis ao contexto [Mobiline 2012]. O projeto MobiLine⁶ evoluiu dentro de três vertentes de pesquisa tecnológica: (i) investigação de características particulares do domínio de software móvel sensível ao contexto, (ii) análise de requisitos comuns a aplicações de guias de visita móveis existentes, e (iii) construção de uma linha de produto de software para guias de visita móveis e sensíveis ao contexto.

Para construir a LPSSC Mobiline, a Engenharia de Domínio foi decomposta em dois níveis de análise, sendo que o primeiro trata das características de uma LPS móvel e sensível ao contexto, e o segundo foca no domínio de guias de visitas móveis [Marinho et al. 2010]. Desta forma, a LPSSC Mobiline possui dois modelos de *features*. Para mais detalhes sobre essa LPSSC é possível consultar o site do projeto de pesquisa Mobiline⁷, que deu origem a LPSSC com o mesmo nome. Para a validação das medidas propostas neste trabalho iremos utilizar o Modelo de *Features* do Mobiline voltado ao domínio de guias de visitas móveis, parte do qual é apresentado na Figura 1. O modelo completo está disponível no site do projeto Mobiline.

4.2. Aplicação e Análise das Medidas

⁶ O projeto MobiLine foi fomentado pelo CNPq (MCT/CNPq 15/2007 Universal - processo número 484523/2007-4) e desenvolvido pelo Grupo de Redes de Computadores, Engenharia de Software e Sistemas (GREat) da Universidade Federal do Ceará em parceria com o Grupo de Reuso da COPPE - UFRJ.

⁷ <http://mobiline.great.ufc.br>

A coleta e análise das medidas foram realizadas por um especialista de medição e linhas de produtos de software. Esta coleta foi realizada de forma mista, tendo uma parte das medidas foi coletada manualmente (MED_1, MED_3, MED_5, MED_6, MED_7, MED_9 e MED_12), enquanto outra parte foi coletada automaticamente (MED_2, MED_4, MED_8, MED_10, MED_11, MED_13, MED_14, MED_15) utilizando-se a ferramenta SPLOT (Mendonça et al. 2009). Para isso, o modelo de *features* do domínio de guias de visitas móveis da Linha Mobiline foi incluído no repositório de modelos de *features* da SPLOT. Inicialmente, a ferramenta SPLOT verifica a consistência, a correte e a configuração de produtos que podem ser gerados a partir do modelo de *features*. Após esta etapa, a SPLOT coleta as medidas. As medidas que não são coletadas pela ferramenta SPLOT foram coletadas manualmente. O resultado das atividades de coleta e análise das medidas utilizadas para a avaliação da qualidade do modelo de *features* selecionado (domínio de guias de visitas móveis da LPS Mobiline) pode ser visualizado na Tabela 3.

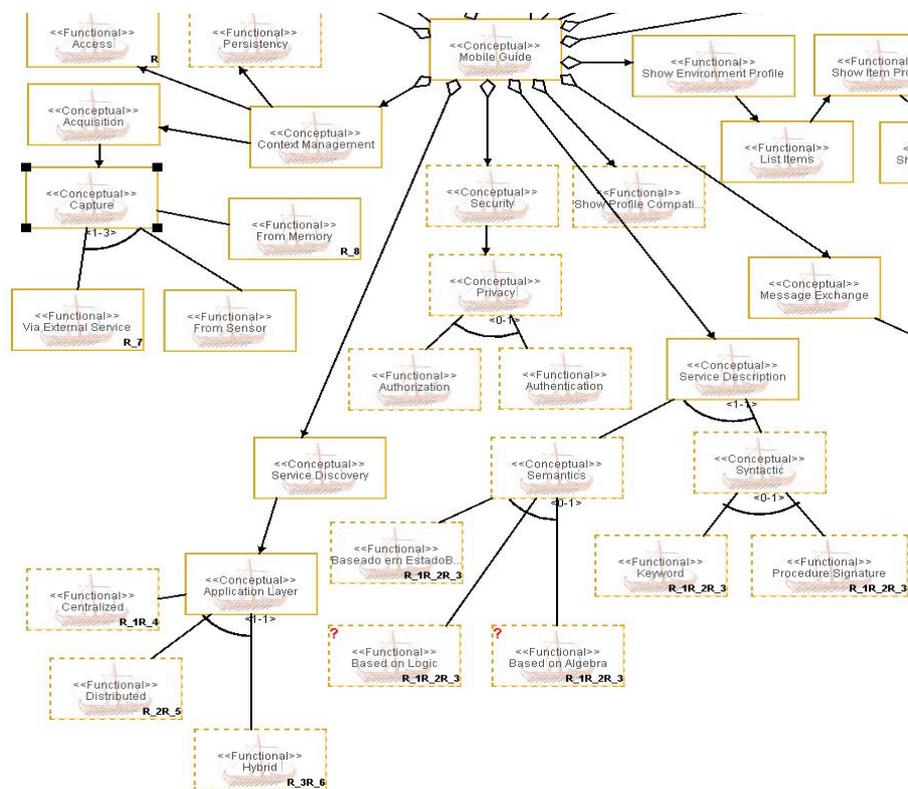


Figura 1: Modelo de *Feature* para aplicações de Guias de Visita Móveis e Sensíveis ao Contexto.

A Tabela 3 apresenta também informações complementares que constituem as especificações das medidas propostas neste trabalho, tais como: unidade de medida e meta. Além disso, a tabela apresenta a coleta e análise das medidas de acordo com o modelo de *features* de Guias de Visitas Móveis na Figura 1. O modelo de *features* de Guias de Visitas Móveis na Figura 1 segue a notação Odyssey-FEX [Oliveira 2006].

Tabela 3: Coleta e Análise das Medidas no Modelo de *Features* da LPSSC de Guias de Visitas Móveis.

ID da Medida	Medida	Unidade de Medida	Meta	Coleta	Análise
MED_01	Número de variabilidades em LPS	Número (0-2)	NV > 0,5	A = 26 B = 30 N = 51 V = 1,10	Devido a existência de <i>features</i> que são alternativas e opcionais ao mesmo tempo o valor pode ser maior que 1. O modelo de <i>features</i> de guias de visitas móveis atende a meta, tendo uma variabilidade boa.
MED_02	<i>Features</i> Comuns	Número	CF < 10	CF = 21	O modelo possui 21 <i>features</i> comuns a todas as configurações. Isso indica um número muito alto de <i>features</i> comuns nos produtos.
MED_03	Número de pontos de variação incluídos	Número	PV < 20	PV = 11	Poucos pontos de variação em relação ao número de <i>features</i> , está dentro da meta esperada.
MED_04	Número de <i>features</i> (NF)	Número	NF < 60	NF = 51	Modelo não muito complexo, pois quanto maior o número de <i>features</i> mais complexo o modelo.
MED_05	Taxa de variabilidade	Número	TV < 3,17	Número de filhos = 22 TV = 1,77	Em média cada <i>features</i> possui 1,77 filhos, sendo menor que 2 filhos por <i>feature</i> . Podemos concluir que a taxa de variabilidade é baixa.
MED_06	Número de <i>features</i> Top	Número	Ntop < 5	NTop = 12	O número de <i>features</i> top é muito alto para LPS.
MED_07	Número de <i>features</i> Leaf	Número	Nleaf < 40	NLeaf = 28	Quanto maior o o número de nós folhas em relação aos nós leaf, maior a variabilidade. O número de <i>features</i> Leaf está dentro da meta.
MED_08	Número de configurações válidas	Número	NCV < 1,614E + 48	NCV = 3,38E + 7	O número de configurações do modelo de <i>features</i> é acima da meta esperada. No entanto, esse valor está relacionado ao número de <i>features</i> e número de restrições. Podemos verificar que temos um grande número de configurações da LPS, indicando muitos produtos que podem ser derivados da LPS.
MED_09	Flexibilidade da configuração	Número (0-1)	FoC > 0,5	NFO=30 NF = 51 FoC = 0,6	A flexibilidade da configuração do modelo é alta, atingindo a meta. Podemos verificar que devido a essa medida temos uma relação também de um número de configurações alta na LPS.
MED_10	Profundidade da árvore	Número	PA < 7	PA = 6	A maior profundidade da árvore é 6, podemos concluir que a profundidade da árvore é baixa.
MED_11	Complexidade ciclomática	Número	CC < 14,84	17	O número de restrições de integridade no modelo de <i>features</i> , indicando uma alta complexidade ciclomática e não atingindo a meta.
MED_12	Coefficiente de conectividade	Número (0-1)	CoC > 1	NA = 51 NF = 51 CoC = 1,00	Número e arestas é igual ao número de <i>features</i> , no modelo não há <i>features</i> sem conexão. Isso indica a consistência do modelo de <i>features</i> .
MED_13	Cross-tree Constraints	Número (0-1)	CTC < 0,22	NFRI = 17 NF = 51 CTC = 0,33	A razão do número de <i>features</i> envolvidas nas restrições de integridade é aproximadamente 1/3 do total de <i>features</i> , sendo considerado um valor alto em relação a meta.
MED_14	Número de <i>Features</i> Mortas	Número	FM = 0	FM = 0	O modelo é correto, pois não possui <i>features</i> que não são passíveis de utilização.
MED_15	Densidade CTC	Número (0-1)	Densidade CTC < 0,84	NR = 9 CC = 17 DCTC = 0,53	A densidade CTC atinge a meta indicando que o número de restrições está dentro dos limites aceitáveis em relação a complexidade ciclomática.

Para cada uma das medidas propostas neste trabalho (Tabela 2), foi definida uma meta (Tabela 3). A meta é um valor tangível de ser alcançado e analisado de acordo com a coleta da medida. Para as medidas MED_01 e MED_02 foram utilizadas as metas definidas em Montagud e Abrahão (2009b). Para definir as metas das medidas MED_04

a MED_13 utilizou-se a média dos valores coletados para os 14 modelos de *features* analisados em Bagheri e Gasevic (2011). Porém, como as medidas MED_02, MED_14 e MED_15 não foram coletadas e nem analisadas em Bagheri e Gasevic (2011), a ferramenta SPLOT (Mendonça et al. 2009) foi utilizada para recuperar o valor dessas medidas para os mesmos 14 modelos de *features* discutidos em Bagheri e Gasevic (2011). Em seguida, a média dos valores coletados para MED_02 foi utilizada como meta para esta medida. O mesmo procedimento foi utilizado para definir as metas de MED_14 e MED_15. A meta é apenas uma sugestão inicial do que pode ser uma medição ideal para cada medida.

Para cada medida foi realizada uma análise com base na meta definida, conforme a Tabela 3. Na próxima seção são discutidos resultados da análise do modelo de *features* de Guias de Visitas Móveis e da abordagem realizada para medição.

4.3. Discussão dos Resultados

Na Tabela 3, pode-se verificar que dez medidas atingem a meta do total de quinze medidas. As cinco medidas que não atenderam as metas são: *Features* Comuns (MED_02), Número de *Features* Top (MED_06), Número de configurações válidas (MED_08), Complexidade Ciclomática (MED_11) e *Cross-tree Constraints* (MED_13). O grande número de *features* top indica que o modelo de *features* analisado apresenta elevada complexidade. Este fato é confirmado pelos altos valores obtidos para o número de *features* comuns, para o número de configurações do modelo, para a quantidade de restrições e para a Complexidade Ciclomática do modelo. Todas essas medidas estão relacionadas à complexidade do modelo. A complexidade pode influenciar a qualidade do modelo.

Como conclusões de análise de todas as medidas, pode-se interpretar que o Modelo de *Features* da linha Mobiline, mostrou-se consistente e correto. O modelo não apresenta *features* mortas, e não possui *features* sem conexão. A variabilidade do modelo encontra-se dentro das metas estabelecidas, atestando a qualidade do modelo da linha. No entanto, o modelo possui uma complexidade alta, apesar das medições de variabilidade estar dentro dos limites esperados.

A principal vantagem de se utilizar a abordagem de medição proposta neste artigo é a facilidade de coleta e análise das medidas, já que foi definida uma especificação. O conjunto de medidas definido para o modelo de *features* se mostrou eficaz na sua aplicação. No entanto, uma das desvantagens na coleta das medidas foi realizar a coleta de parte das medidas de forma manual, uma vez que não foram encontradas ferramentas para calcular essas medidas. O cálculo de forma manual pode induzir a erros humanos, e as coletas podem se tornar não confiáveis. O fato das medidas terem sido definidas para o modelo de *features* faz com que a qualidade da linha seja atestada logo no início, evitando que erros sejam propagados para o restante dos artefatos e produtos da linha. Todas as medidas foram relacionadas à característica de qualidade Manutenibilidade, isso indica que a qualidade da linha é garantida para que o modelo de *features* seja de fácil manutenção.

5. Trabalhos Relacionados

A maior parte das pesquisas relacionadas ao modelo de *features* de linhas de produtos de software tem focado no desenvolvimento e validação de configurações de produtos

baseados em dados conjuntos de restrições dos *stakeholders*. Entretanto, poucas pesquisas têm sido desenvolvidas em qualidade de linhas de produtos de software, tais como a definição de indicadores de qualidade e avaliação interna e externa de atributos de qualidade [Bagheri and Gasevic 2011]. Poucos trabalhos foram encontrados de forma a definir medidas ou indicadores de qualidade para o modelo de *features* [Montagud and Abrahão 2009b; Bagheri and Gasevic 2011; Mendonça et al. 2009]. Outros trabalhos avaliam a qualidade do modelo de *features* utilizando técnicas de inspeção de software [Mello et al. 2012; Souza et al. 2012].

Montagud e Abrahão (2009b) apresentam um método de avaliação de qualidade baseado no SQuaRE [ISO/IEC 25000 2005]. Este trabalho define medidas para avaliação de todas as fases da LPS, mapeando-os com características de qualidade e atributos de qualidade de acordo com a norma de qualidade SQuaRE. Também é definido como esse método de avaliação deve ser aplicado. Neste trabalho foram identificadas apenas duas medidas relacionadas ao modelo de *features*, no qual uma foi adaptada a este trabalho.

Montagud e Abrahão (2011) publicaram uma revisão sistemática da literatura com o objetivo de identificar e interpretação de todos os estudos que apresentam atributos de qualidade e/ou medidas para SPL. Estes atributos e as medidas foram classificados utilizando um conjunto de critérios que inclui a fase de ciclo de vida em que as medidas são aplicadas, correspondentes às características de qualidade; seu apoio para as características SPL específicas, o procedimento usado para validar as medidas, etc. Foi elaborado ao final do estudo um catálogo com a identificação de todas as medidas identificadas para todas as fases da Linha de Produto de Software. No entanto, nesse catálogo as medidas referentes à fase de domínio da LPS são mapeadas para Requisitos, *Design* e Testes, não sendo encontradas medidas para avaliar o modelo de *features*, apenas para casos de uso.

Bagheri e Gasevic (2011) propõem um conjunto de medidas estruturais para modelos de *features* de linhas de produtos de software e, teoricamente valida as medidas utilizando princípios teóricos para validar medições. No trabalho é realizada uma investigação através da experimentação controlada de forma a analisar se estas medidas estruturais podem ser bons indicadores (indicadores precoces) das três subcaracterísticas principais de manutenção: analisabilidade, mutabilidade e compreensibilidade. No trabalho de Bagheri e Gasevic (2011) foi identificada a maioria das medidas para avaliar o modelo de *features* citadas neste trabalho, conforme ilustrado na Tabela 1 da Seção 3. No entanto, foi verificado que o trabalho não possui uma especificação dessas medidas, apenas contendo a descrição e o estudo da coleta baseada nas medidas propostas.

Em Mendonça et al. (2009) o foco do trabalho é a apresentação da ferramenta SPLOT, que é um sistema web com interfaces para configuração e recomendação de linhas de produtos de software a partir do modelo de *features*. A ferramenta apresenta também um repositório de modelos de *features* com mais de 300 modelos, que facilita o compartilhamento de modelos de *features* de forma a compartilhar esse conhecimento com pesquisadores da área. Dessa forma, a partir dos modelos de *features* que podem ser selecionados é possível obter dados de análises dos modelos calculadas de forma automática pela ferramenta, apenas analisando os modelos de *features*. Parte das medidas extraídas deste trabalho são apresentadas na Seção 3 do presente artigo, as

medidas foram extraídas da própria ferramenta e podem ser encontradas ao introduzir os modelos de *features* e ao selecionar a função de análise automática.

6. Conclusões e Trabalhos Futuros

Neste trabalho, foi apresentado um conjunto de medidas voltadas para a avaliação de qualidade do modelo de *features* em Linhas de Produtos de Software. Essas medidas foram identificadas a partir de uma ampla revisão bibliográfica. Adicionalmente, as medidas identificadas foram estendidas e adaptadas em diversos aspectos, como por exemplo: i) para cada medida foram definidas as características e os atributos de qualidade as quais ela está associada e ii) as medidas foram especificadas e detalhadas, a fim de tornar possível a coleta e a análise das medidas. Por fim, para validar a solução proposta, utilizamos o conjunto de medidas proposto neste trabalho para realizar uma avaliação da qualidade de um modelo de *features* real, pertencente a uma LPS móvel e sensível ao contexto denominada Moline. Este modelo de *features* foi avaliado após a coleta das medidas.

O conjunto de medidas proposto se mostrou eficaz na sua aplicação, e a especificação das medidas facilitou a coleta e análise no modelo de *features* pelo especialista. Parte da coleta foi realizada de forma automática, com o auxílio da ferramenta SPLOT [Mendonça et al. 2009], o que facilitou o trabalho do especialista. Do total de 15 medidas, 8 foram coletadas automaticamente pela ferramenta SPLOT. As demais 7 medidas foram coletadas manualmente. A partir da utilização do conjunto de medidas proposto neste trabalho foi possível atestar a qualidade e as limitações do modelo de *features* avaliado. No entanto, o conjunto de metas definidas para as medidas propostas poderia ser refinado, a partir de estudos estatísticos sobre o comportamento dos dados. Como trabalho futuro, pretende-se realizar um refinamento das especificações das medidas utilizando-se os dados do repositório de modelos de *features* da ferramenta SPLOT, visando a avaliação de maior quantidade de modelos de *features*. Assim, as metas e as diretrizes para a análise de cada uma das medidas propostas, por exemplo, podem ser refinados. Adicionalmente, pretende-se definir novas medidas que estejam relacionadas diretamente com o contexto no modelo de *features* de uma LPSSC, uma vez que os trabalhos encontrados na literatura limitam-se a medidas que não consideram o contexto da aplicação, fator fundamental em uma LPSSC.

Agradecimentos

Este artigo é um resultado parcial do projeto UbiStructure com apoio do CNPq (MCT/CNPq 14/2011 - Universal) com o número 481417/2011-7 e do projeto Maximum da FUNCAP/FAPs/INRIA/INS2i-CNRS com o número 0064-000120100/12.

Referências

- Bagheri, E.; Gasevic, D. (2011) “Assessing the maintainability of software product line feature models using structural metrics”. *Software Quality Journal*, 2011 – Springer.
- Bartholdt, J.; Medak, M. e Oberhauser, R. (2009) “Integrating Quality Modeling with Feature Modeling in Software Product Lines”. ICSEA '09. Fourth International Conference on Software Engineering Advances.
- Benavides, D.; Segura, S.; Trinidad, P.; Ruiz-Cortés, A. (2007) “FAMA: Tooling a Framework for the automated analysis of feature models”. In 1st International

- Workshop on Variability Modelling of Software Intensive Systems, 2007, pp.129-134.
- Bosch, J. (2000) "Design and use of software architectures: adopting and evolving a product line approach". ACM Press/Addison- Wesley Publishing Co., USA.
- Etxeberria, L. e Sagardui, G. (2005) "Product-line architecture: New issues for evaluation". In J. H. Obbink and K. Pohl, editors, 9th International Conference on Software Product Lines, SPLC, Proceedings, volume 3714 of Lectures Notes in Computer Science, pages 174–185. Springer.
- Etxeberria, L. e Sagardui, G. (2008a) "Variability Driven Quality Evaluation in Software Product Lines," in Proceedings of 12th International Software Product Line Conference, pp. 243-252.
- Etxeberria, L. e Sagardui, G. (2008b) "Evaluation of quality attribute variability in software product families." in Proceedings of the 15th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems, pp. 255-264.
- IEEE Standard for a Software Quality Metrics Methodology IEEE Std. 1061- 1998.
- ISO/IEC 9126-1 International Standard: Software engineering - Product quality. Part 1: Quality model, 2003.
- ISO / IEC 25000: 2005. Software Engineering. Software product Quality Requirements and Evaluation (SQuaRE).
- Kang, K. C.; Cohen, S. G.; Hess, J. A.; Novak, W. E. e Peterson, A. S. (1990) "Feature-Oriented Domain Analysis (FODA) Feasibility Study," Carnegie-Mellon University Software Engineering Institute, Tech. Rep.
- Kim, T.; Ko, I. Y.; Kang, S.W.; Lee, D. H. (2008) "Extending ATAM to assess product line architecture". In 8th IEEE International Conference on Computer and Information Technology, pp. 790-797.
- Marinho, F.; Costa, A.; Lima, F.; Neto, J.; Filho, J.; Rocha, L.; Dantas, V.; Andrade, R.; Teixeira, E.; Werner, C. (2010) "An architecture proposal for nested software product lines in the domain of mobile and context-aware applications". In: Proceedings of the 2010 Fourth Brazilian Symposium on Software Components, Architectures and Reuse, 2010. (SBCARS '10), p. 51–60. ISBN 978-0-7695-4259-1.
- Matinlassi, M.; Niemelä, E., Dobrica, L. (2002) "Quality-driven architecture design and quality analysis method: A revolutionary initiation approach to a product line architecture". Technical Report VTT-PUBS-456, VTT.
- Mello, R. M. ; Teixeira, E. N. ; Schots, M. ; Werner, C. M. L. ; Travassos, G. H. (2012) "Checklist-based Inspection Technique for Feature Models Review". In: VI Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização de Software, 2012, Natal, v. 3. p. 139-148.
- Mendonca, M., Branco, M. e Cowan, D. (2009) "S.P.L.O.T. - Software Product Lines Online Tools". In Companion to the 24th ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA, Orlando, Florida, USA

- Mobiline. Uma Linha de Produto de Software Móvel e Sensível ao Contexto. 2012. Disponível em <http://mobiline.great.ufc.br/>. Último acesso em março de 2013.
- Montagud, S. e Abrahão, S. (2009a) Gathering Current Knowledge about Quality Evaluation in Software Product Lines. Proceedings of the 13th International Software Product Line Conference (SPLC), San Francisco, USA, 2009.
- Montagud, S.; Abrahão, S. (2009b). A SQuaRE-based quality evaluation method for software product lines. Master's thesis, December 2009 (in Spanish).
- Montagud, S.; Abrahão, S. e Insfran, E. (2011) "A systematic review of quality attributes and measures for software product lines". *Software Quality Journal*, 2011 – Springer.
- Oliveira, R. F. (2006) *Formalização e Verificação de Consistência na Representação de Variabilidades*, Dissertação de M.Sc., COPPE Sistemas, UFRJ, Rio de Janeiro, Brasil.
- Oliveira, E. A. Jr., Gimenes, I. M. S., & Maldonado, J. C. (2008). A metric suite to support so product line architectures. In XXXIV Conferência Latinoamericana de Informática (CLEI 2008).
- Olumofin, F. G. e Mišic, V. B. (2007) "A holistic architecture assessment method for software product lines". *Information and Software Technology* 49, pp. 309-323.
- Souza, I; Oliveira, P. R.; Gomes, G; Almeida, E. S. (2012) "On the Relationship between Inspection and Evolution in Software Product Lines: An Exploratory Study." 26th Brazilian Symposium on Software Engineering (SBES), Natal, Brazil.
- Tawhid, R. e Petriu, D. C. (2011) "Automatic Derivation of a Product Performance Model from a Software Product Line Model". SPLC '11 Proceedings of the 15th International Software Product Line Conference.
- Tessier, P.; Gérard, S.; Terrier, F. e Geib, J. (2005) "Using variation propagation for model-driven management of a system family." In *Software Product Lines*, pp. 222–233.
- Thiel, S. (2002) "On the definition of a framework for an architecting process supporting product family development". In 4th International Workshop on Software Product-Family Engineering, Springer-Verlag, London, UK, 2002, pp.125–142.
- Villela, K.; Arif, T; Zanardini, D. (2012) "Towards product configuration taking into account quality concerns". SPLC '12 Proceedings of the 16th International Software Product Line Conference - Volume 2, Pages 82-90.
- Zhang, H.; Jarzabek, S. e Yang, B. (2003) "Quality Prediction and Assessment for Product Lines," in CAiSE 2003, LNCS 2681, pp. 681-695.
- Zhang, T.; Deng, L.; Wu, J.; Zhou, Q. e Ma, C. (2008) "Some metrics for accessing quality of product line architecture". In CSSE'08: Proceedings of the 2008 International Conference on Computer Science and Software Engineering (pp. 500–503). Washington, DC: IEEE Computer Society.