

Medidas para Avaliação da Manutenibilidade do Modelo de *Features* de Linhas de Produto de Software Tradicionais e Dinâmicas

Carla I. M. Bezerra^{1,2,3}, Rossana M. C. Andrade^{1,2*}, José Maria Monteiro¹

¹MDCC – Mestrado e Doutorado em Ciências da Computação
Universidade Federal do Ceará (UFC) Fortaleza – CE - Brasil

²GREat - Grupo de Redes de Computadores e Engenharia de Software

³Campus Quixadá - Universidade Federal do Ceará (UFC)

carlailane@ufc.br, rossana@ufc.br, monteiro@dc.ufc.br

Abstract. *This research aims to investigate the feature models maintainability and to propose solutions to evaluate using measures. To do that, it is necessary to built a quality measures catalog and, to support the catalog usage, to develop a tool, which allows the automatic collection of measurements belonging to this catalog. To validation was realized an exploratory study that investigates the impact of the feature models evolution in the maintainability of these models; an exploratory case study that explores the relationships among the maintainability measures; and a study for aggregating measures, especially related to DSPLs, using fuzzy logic. The results of this thesis suggest the quality measures can be effectively used to support the feature models maintainability.*

Resumo. *Este trabalho tem por objetivo investigar a manutenibilidade e propor soluções para avaliar o modelo de features utilizando medidas. Para atingir esse objetivo, foi elaborado um catálogo de medidas de qualidade de manutenibilidade, e para apoiar o uso do catálogo, foi desenvolvida uma ferramenta, que permite a coleta automática de medições pertencentes a este catálogo. Para validação foi realizado um estudo exploratório que investiga o impacto da evolução dos modelos de features na manutenibilidade; um estudo de caso exploratório efetuado com o intuito de explorar os relacionamentos entre as medidas de manutenibilidade; e um estudo com o propósito de agregar medidas, relacionadas à LPSDs, por meio da utilização de lógica fuzzy. Os resultados desta tese sugerem que as medidas de qualidade podem ser efetivamente utilizadas para apoiar a avaliação da manutenibilidade de modelos de features.*

1. Introdução

O paradigma de Linha de Produtos de Software (LPS) tem sido considerado, tanto pela indústria quanto pela academia, uma estratégia eficiente para lidar com as diferentes necessidades dos usuários, uma vez que torna possível personalizar um produto de software por meio da seleção das variabilidades antes mesmo da sua implantação. Todavia, as aplicações sensíveis ao contexto e/ou baseadas em computação

*Bolsa de pesquisa - DT Level 2, financiada pelo CNPq

ubíqua/pervasiva requerem variabilidades dinâmicas, ou seja, resolvidas em tempo de execução [Hallsteinsen et al. 2008]. Neste cenário, surgiu o paradigma denominado Linhas de produtos de software dinâmicas (LPSD) [Hallsteinsen et al. 2008]. Uma LPSD tem por finalidade produzir software capaz de se adaptar de acordo com as necessidades de usuários e restrições de recursos em tempo de execução. As LPSDs podem vincular pontos de variação quando o software é iniciado para se adaptar ao ambiente (contexto) atual, bem como durante sua operação para se adaptar às mudanças no ambiente [Capilla et al. 2013]. Desta forma, LPSDs podem apresentar características de sistemas autônomos, autoadaptáveis ou sensíveis ao contexto.

Um dos importantes ativos de uma linha de produto é o modelo de *features*¹. Este modelo representa as *features* do domínio e a variabilidade da linha. Já as *features* descrevem as funcionalidades e as características de qualidade de um produto de software. Desta forma, Um modelo de *features* modela todos os produtos possíveis de uma LPS [Benavides et al. 2010]. Por outro lado, uma LPSD deve ser ciente do estado de um ambiente (contexto), percebendo suas mudanças. Por este motivo, os modelos de *features* de LPSDs devem representar também a variabilidade dinâmica. Neste cenário, o modelo de *features* de LPSDs envolvem a definição das *features* de contexto e das restrições de contexto [Capilla et al. 2014].

A avaliação da qualidade é uma atividade essencial em uma linha de produtos, uma vez que um erro ou inconsistência em um artefato pode ser propagado para todos os seus produtos [Montagud and Abrahão 2009]. No entanto, a avaliação da qualidade de todos os artefatos e produtos de software de uma determinada linha revela-se impraticável, tanto por razões econômicas quanto pelo esforço necessário [Etxeberria and Sagardui 2008a]. Desta forma, uma alternativa eficiente e econômica consiste em avaliar somente os artefatos mais relevantes. Neste sentido, como o modelo de *features* é um dos principais artefatos de uma linha de produto, avaliar a sua qualidade torna-se de fundamental importância. Por outro lado, uma estratégia bastante popular para avaliar a qualidade de um produto (i.e., software) consiste na utilização de medidas. Uma medida é o mapeamento de uma entidade para um número ou um símbolo com o objetivo de caracterizar uma propriedade da entidade [ISO/IEC 2014]. Vale destacar que a avaliação da qualidade do modelo de *features* em LPSDs envolvem uma maior complexidade, por ser necessário modelar e representar as diversas adaptações de contexto, as *features* de contexto e as restrições relacionadas a esses contextos.

Recentemente, diversos trabalhos têm discutido e apresentado medidas para a avaliação da qualidade do modelo de *features* em LPSs [Thörn 2010, Bagheri 2011, Montagud et al. 2012, Berger 2014]. Contudo, na maioria desses trabalhos, as definições das medidas apresentam diferentes problemas, tais como: ausência da fórmula de cálculo, falta de uma semântica clara, entre outros. Além disso, essas medidas foram avaliadas a partir de sua aplicação em um número pequeno de modelos de *features*. Já no contexto de LPSDs, poucos trabalhos têm sido propostos com o objetivo de indicar medidas para a avaliação da qualidade do modelo de *features* [Bagheri 2011, Montagud et al. 2012, Berger 2014]. Porém, as poucas medidas propostas para LPSDs apresentam os mesmos

¹Neste trabalho a palavra “*feature model*” foi traduzida para “modelo de *features*”, e não “modelo de características”, pois como é tratado nesta tese o conceito de características de qualidade, a tradução para modelo de características poderia confundir os conceitos.

problemas encontrados na literatura para LPSs. A maior parte dos trabalhos relacionados à avaliação da qualidade de LPSDs concentra-se na verificação e validação da variabilidade dinâmica do modelo de *features* [Capilla and Bosch 2011, Marinho et al. 2012, Alférez et al. 2014]. Além disso, a manutenibilidade do modelo de *features* é fator crítico. Isso ocorre devido o domínio das LPSs e LPSDs se modificarem constantemente e a mudança da estrutura do modelo de *features* impacta fortemente na sua manutenibilidade [Bagheri 2011, Bezerra et al. 2013, Berger 2014]. Desta forma, foi identificada, então, uma lacuna na área de avaliação da manutenibilidade do modelo de *features* em LPSs e LPSDs.

Neste cenário, este trabalho propõe um suporte para avaliação da qualidade do modelo de *features* em LPSs e LPSDs, composto por: (i) COFEE, um catálogo de características, subcaracterísticas e medidas para avaliação da qualidade do modelo de *features* de LPSs e LPSDs; (ii) DyMMer, uma ferramenta para suportar a avaliação da qualidade do modelo de *features*, possibilitando a coleta automática das medidas do catálogo e a modelagem dos modelos de *features* de LPSDs [Bezerra et al. 2016b]; e (iii) três *datasets* (AFFOgaTO, MACchiATO, ESPREssO), com coleta das medidas em modelos de *features* extraídos do S.P.L.O.T. e da literatura, para auxiliar na investigação da pesquisa.

Além disso, foram realizados três estudos para análise da utilização das medidas na avaliação da qualidade do modelo de *features* de LPSs e LPSDs, que são: (i) um estudo exploratório utilizando medidas de manutenibilidade para avaliar a qualidade da evolução dos modelos de *features* de LPSs [Bezerra et al. 2016c]; (ii) um estudo de caso para investigar a aplicação das medidas de LPSs, analisando a correlação entre essas medidas, agrupando as medidas e definindo *thresholds* para as medidas agrupadas [Bezerra et al. 2016a]; e (iii) a agregação de medidas utilizando lógica *fuzzy*, gerando novas medidas relacionadas ao tamanho, estabilidade, flexibilidade e dinamicidade de modelos de *features* de LPSDs. Devido a restrições de tamanho do artigo, será apresentado apenas o estudo (ii).

2. Fundamentacao Teórica

A avaliação da qualidade no contexto de Linha 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 and Sagardui 2008b]. Neste contexto, muitas abordagens de avaliação da qualidade em LPSs têm sido propostas ao longo dos últimos anos [Olumofin and Mišić 2007, Benavides Cuevas et al. 2007, Kim et al. 2008, Mendonca et al. 2009, Bagheri 2011].

Montagud e Abrahão (2009) executaram uma revisão sistemática compilando o conhecimento sobre avaliação da qualidade em LPSs [Montagud and Abrahão 2009]. Dentre os dados extraídos dos estudos primários identificados nesta revisão tem-se as abordagens de avaliação da qualidade, as fases do ciclo de vida, os artefatos avaliados, os mecanismos usados para capturar os atributos de qualidade, o tipo de atributo de qualidade avaliado, a conformidade com padrões de qualidade, o suporte a avaliação e o procedimento de avaliação. Nesse estudo foi verificado que a maior parte dos modelos de avaliação da qualidade se concentram na fase de arquitetura da engenharia de domínio da LPS [Olumofin and Mišić 2007, Kim et al. 2008], pou-

cos estudos tem avaliado o modelo de *features* [Trinidad et al. 2008, Bagheri 2011]. Poucas ferramentas de suporte a avaliação da qualidade foram identificadas na revisão (e.g., [Benavides Cuevas et al. 2007, Trinidad et al. 2008, Mendonca et al. 2009]), e algumas ferramentas identificadas apenas suportam a análise semi-automática da avaliação da arquitetura. A maior parte dos estudos também não utiliza padrões e normas de conformidade, como o SQuaRE [ISO/IEC 2014], para avaliação da qualidade [Olumofin and Mišić 2007, Trinidad et al. 2008, Mendonca et al. 2009].

Thüm et al. (2014) elaboraram um *survey* que abordam estratégias de análise de linhas de produto de software [Thüm et al. 2014]. Nesse estudo, algumas abordagens identificadas estão relacionadas à abordagens e estudos sobre testes em LPSs. Outras abordagens identificadas no *survey* e na literatura estão relacionadas à verificação e análise automática do modelo de *features* [Benavides et al. 2010, Marinho et al. 2012, Asadi et al. 2016]. Além disso, outros estudos identificados apresentam medidas para avaliar o modelo de *features* de LPSs [Bagheri 2011, Montagud et al. 2012, Berger 2014, Oliinyk et al. 2015, Bezerra et al. 2015].

A avaliação da qualidade em LPSDs ainda é um tema recente e possui maiores desafios que a avaliação da qualidade em LPSs, devido à variabilidade dinâmica das LPSDs. Foram identificados na literatura alguns estudos que tratam da verificação das LPSDs em tempo de execução [Pascual et al. 2015, Lochau et al. 2015]. Também foram identificados outros estudos que propuseram abordagens para o modelo de *features* em LPSDs e avaliaram essas abordagens com medidas específicas [Cetina et al. 2009, Saller et al. 2013, Alférez et al. 2014]. No entanto, não foram identificadas na literatura abordagens de avaliação da qualidade para o modelo de *features* de LPSDs.

3. Metodologia

A pesquisa realizada neste trabalho adota a metodologia ilustrada na Figura 1. As atividades realizadas durante a execução desta pesquisa são detalhadas a seguir:

- **Estudo do Referencial Teórico:** estudo dos principais conceitos relacionados a LPS tradicional e dinâmica, modelo de *features*, avaliação da qualidade de produto e medição de software.
- **Definição do catálogo COFEE:** realização de uma ampla revisão da literatura com o objetivo de identificar, classificar e catalogar medidas que têm sido utilizadas para avaliar a qualidade de modelos de *features*.
- **Construção da ferramenta DyMMer e *datasets*:** construir uma ferramenta para a coleta automática das medidas catalogadas e desenvolver *datasets* para apoiar a

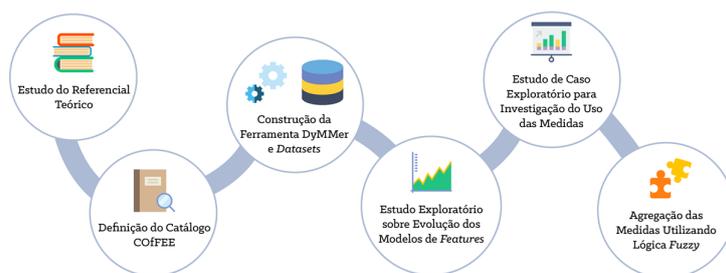


Figura 1. Metodologia da pesquisa

utilização e avaliação dessas medidas.

- **Estudo exploratório sobre o modelo de *features*:** realizar um estudo exploratório sobre o impacto da evolução dos modelos de *features* do ponto de vista de manutenibilidade.
- **Estudo de caso exploratório para investigação do uso das medidas:** realizar um estudo de caso exploratório com a finalidade de investigar a utilização de três técnicas de análise de dados para identificar correlações entre as medidas, agrupar as medidas e definir limites superiores e inferiores para estas medidas.
- **Agregação das medidas utilizando lógica *fuzzy*:** realizar um estudo exploratório que buscou utilizar Lógica *Fuzzy* com a finalidade de agregar medidas de qualidades voltadas para a avaliação da qualidade do modelo de *features* em LPSDs.

4. COFFEE: Um Catálogo de Características, Subcaracterísticas e Medidas para Avaliação da Manutenibilidade do Modelo de *Features*

Com o objetivo de identificar um conjunto de características, subcaracterísticas e medidas de qualidade que possam ser utilizadas para suportar a avaliação da qualidade de modelos de *features* em LPSs, foi realizado um mapeamento sistemático o qual foi publicado em [Bezerra et al. 2015, Bezerra et al. 2016a]. Além disso, foi realizada uma revisão da literatura complementar ao mapeamento sistemático com a finalidade de identificar características, subcaracterísticas e medidas de qualidade voltadas especificamente para modelos de *features* de LPSDs. A partir desses dois estudos, buscou-se catalogar as medidas de qualidade encontradas na literatura com a finalidade de suportar a avaliação do modelo de *features*, além de averiguar se essas medidas possuem ou não procedimentos operacionais de coleta e análise. Inicialmente foram identificadas medidas para várias características de qualidade compatíveis com a norma SQuaRE [ISO/IEC 2014], como: adequação funcional, manutenibilidade, usabilidade, eficiência de desempenho, portabilidade, confiabilidade e segurança. No entanto, a maior parte das medidas identificadas estavam relacionadas à característica de manutenibilidade.

Com isso, foi possível compilar um catálogo de medidas de qualidade que pode ser utilizado na avaliação da manutenibilidade de modelos de *features*, denominado COFFEE (*CatalOg of measures for Feature modEl quality Evaluation*), o qual é composto por 40 medidas de qualidade para característica de qualidade de manutenibilidade. Essas medidas foram ilustradas na Tabela 1. As medidas estão associadas à subcaracterísticas de qualidade. As subcaracterísticas em negrito são específicas do modelo de *features* e foram propostas no trabalho.

As fórmulas de cálculo das 40 medidas do catálogo COFFEE são apresentadas na Tabela 3. As fórmulas de cálculo das medidas estão descritas de acordo com as descrições dos artigos originais.

O catálogo de medidas COFFEE é uma contribuição desta tese, e suas medidas foram implementadas em uma ferramenta desenvolvida neste trabalho para que o Engenheiro de Domínio possa realizar de forma automática a coleta das medidas em modelos de *features* de LPSs e LPSDs para suportar a avaliação do modelo com base nas medidas e propor melhorias no modelo. As medidas do catálogo serão utilizadas como base para os estudos realizados nesta tese.

Tabela 1. Catálogo COfFEE - Um catálogo de medidas de qualidade para suportar a avaliação da manutenibilidade do modelo de *features*.

Característica	Subcaracterísticas	Medidas
Manutenibilidade	Analísabilidade	Número de <i>features</i> folhas (NLeaf)
	Complexidade Cognitiva / Entendabilidade	Complexidade Cognitiva do Modelo de <i>Features</i> (CogC)
	Extensibilidade	Extensibilidade da Feature (FEX)
	Flexibilidade	Flexibilidade da Configuração (FoC)
	Modularidade	<i>Features</i> Dependentes Cíclicas Únicas (SCDF); <i>Features</i> Dependentes Cíclicas Múltiplas (MCDF)
	Complexidade Estrutural	Número de <i>Features</i> (NF); Número de <i>Features</i> Mandatórias (NM); Número de <i>Features</i> Top (NTop); Profundidade da Árvore (DT Max, DT Mean and DT Median); Complexidade Ciclomática (CyC); Complexidade Composta (ComC); Restrições <i>Cross-tree</i> (CTC); Restrições Variáveis <i>Cross-tree</i> (CTCV); Taxa de Conectividade do Grafo (RCon); Densidade do Grafo (RDen); Coeficiente de Densidade da Conectividade (CoC); Número de <i>Features</i> Agrupadas (NGF); Número de Filhos por <i>Feature</i> (BF Max)
	Variabilidade Estática	Número de <i>Features</i> Opcionais (NO); <i>Single Hotspot Features</i> (SHoF); <i>Multiple Hotspot Features</i> (MHoF); <i>Rigid Nohotspot Features</i> (RNoF); Número de <i>Features</i> Variáveis (NVF); Número de Configurações Válidas (NVC); Taxa de Variabilidade (RoV); Número de Grupos Or (NGOr); Número de Grupos XOr (NGXOr); Taxa de <i>Features</i> Or (ROr); Taxa de <i>Features</i> XOr (RXOr)
Variabilidade Dinâmica	Número de Contextos (NC); Número de <i>Features</i> Desativadas (NDF); Número de <i>Features</i> Ativadas (NAF); <i>Features</i> de Contextos em Restrições (CFC); Número de Restrições de Contexto (NCC); Número de <i>Features</i> de Contextos (CF); Número de <i>Features</i> Ativadas por Contexto (AFCA); Número de <i>Features</i> Desativadas por Contexto (DFCA)	

5. A Ferramenta DyMMer

Baseado no catálogo de medidas COfFEE, foi implementada uma ferramenta denominada DyMMer (*Dynamic feature Model tool based on Measures*). A ferramenta DyMMer suporta a edição de modelos de *features* e a extração automática das 40 medidas de qualidade que compõem o catálogo COfFEE.

A ferramenta DyMMer foi desenvolvida para extrair medidas de modelos de *features* representados segundo o formato XML proposto pelo repositório de modelo de *features* S.P.L.O.T. Desta forma, inicialmente, a DyMMer recebe como entrada um conjunto de modelos de *features*, onde cada modelo de *features* é representado por um arquivo XML. Em seguida, cada arquivo XML é processado individualmente e uma estrutura interna em memória principal (objeto Java) é criada para representá-lo. Assim, para cada modelo de *features* teremos um arquivo XML e uma representação interna. Utilizando essas estruturas internas, a DyMMer calcula, automaticamente, para cada modelo de *features*, os valores das 40 medidas de qualidade presentes no catálogo COfFEE. A ferramenta DyMMer e sua documentação estão disponíveis *on-line* ².

As principais funcionalidades da ferramenta DyMMer ilustradas na Figura 2, são:

- **Importação de um Modelo de *Features*:** A DyMMer, diferentemente de outras ferramentas, não foi concebida com o objetivo de ser um editor de modelos de *features*, mas uma ferramenta para extração automática de medidas de qualidade. Desta forma, a DyMMer importa modelos de *features* já existentes, descritos no

²<https://github.com/DyMMerProject/DyMMerV2>

Tabela 3. Catálogo COFEE: Fórmulas de cálculo das medidas.

Acronímico	Nome da Medida	Fórmula de Cálculo
NF	Número de <i>Features</i>	Quantidade de <i>features</i> do modelo
NO	Número de <i>Features</i> Opcionais	Quantidade de <i>features</i> opcionais do modelo
NM	Número de <i>Features</i> Mandatórias	Quantidade de <i>features</i> mandatórias do modelo
NTop	Número de <i>Features</i> Top	Quantidade de <i>features</i> descendentes da raiz da árvore
NLeaf	Número de <i>Features</i> Folhas	Quantidade de <i>features</i> sem filhos da árvore
DT Max	Profundidade Máxima	Número de <i>features</i> do caminho mais longo a partir da raiz do modelo de <i>features</i>
DT Mean	Profundidade Média	Número de <i>features</i> do caminho médio a partir da raiz do modelo de <i>features</i>
DT Median	Profundidade Mediana	Número de <i>features</i> do caminho mediano a partir da raiz do modelo de <i>features</i>
CogC	Complexidade Cognitiva	Número de Pontos de Variação
FEX	Extensibilidade da <i>Feature</i>	NLeaf + SCDF + MCDF
FoC	Flexibilidade da Configuração	NO/NF
SCDF	<i>Features</i> Dependentes Cíclicas Únicas	$\sum (\#Features$ participantes de restrições e que são filhas de pontos de variação com cardinalidade [1..1])
MCDF	<i>Features</i> Dependentes Cíclicas Múltiplas	$\sum (\#Features$ participantes de restrições e que são filhas de pontos de variação com cardinalidade [1..*])
CyC	Complexidade Ciclomática	Quantidade de restrições de integridade
ComC	Complexidade Composta	$NF^2 + (NM^2 + 2*NO^2 + 3*NXOr^2 + 3*NGF^2 + 3*R^2)/9$ R = NGF + CyC
NGF	Número de Agrupamento de <i>Features</i>	Número de agrupamentos de relações Or e XOr
CTCV	Restrições Variáveis de <i>Cross-Tree</i>	Número de variáveis distintas em restrições <i>cross-tree</i>
CTCR	Taxa de Restrições <i>Cross-Tree</i>	$NFRI*/NF$ *Número de <i>features</i> envolvidas em restrições de integridade
RCon	Taxa de Conectividade do Grafo	Conectividade de grafo de dependência, o qual é a taxa de <i>features</i> que referenciam outras <i>features</i> (exceto pais) em suas restrições
RDen	Densidade do grafo	Média do número de <i>features</i> não-pais que são referenciadas em restrições
CoC	Coefficiente de Densidade de Conectividade	(Número de arestas)/NF
NVF	<i>Features</i> Variáveis	NA + NO
SHoF	<i>Single Hotspot Features</i>	Número de <i>features</i> filhas de pontos de variação com cardinalidade [1..1]
MHoF	<i>Multiple Hotspot Features</i>	Número de <i>features</i> filhas de pontos de variação com cardinalidade [1..*]
RNoF	<i>Rigid Nohotspot Features</i>	Número de <i>features</i> não filhas de pontos de variação
RoV	Taxa de Variabilidade	\sum (Média do número de filhas dos nós)
NVC	Número de Configurações Válidas	Número de possíveis configurações válidas do modelo de <i>features</i>
BF Max	Fator de Ramificação Máximo	Número máximo de filhos por <i>feature</i>
NGOr	Número de Grupos Or	Número de pontos de variação com relacionamentos Or
NGXOr	Número de Grupos XOr	Número de pontos de variação com relacionamentos XOr
ROr	Taxa de <i>Features</i> Or	Taxa de <i>features</i> filhas de um relacionamento Or
RXOr	Taxa de <i>Features</i> XOr	Taxa de <i>features</i> filhas de um relacionamento XOr
NC	Número de Contextos	Número de contextos do modelo de <i>features</i>
NAF	Número de <i>Features</i> Ativadas	Número de <i>features</i> ativadas em cada contexto
NDF	Número de <i>Features</i> Desativadas	Número de <i>features</i> desativadas em cada contexto
NCC	Número de Restrições de Contexto	Número de restrições de um contexto
CF	Número de <i>Features</i> de Contextos	Número de <i>features</i> que estão sempre presentes nos contextos ativos do modelo de <i>features</i>
CFC	<i>Features</i> de Contextos em Restrições	Número de <i>features</i> que estão presentes nos contextos e nas restrições do modelo de <i>features</i>
AFCA	Número de <i>Features</i> Ativadas por Contexto	Número de <i>features</i> ativadas em cada contexto / NC
DFCA	Número de <i>Features</i> Desativadas por Contexto	Número de <i>features</i> desativadas em cada contexto / NC

formato XML definido pelo repositório S.P.L.O.T., processa esses modelos e cria, para cada um deles, uma representação interna em memória principal.

- **Visualização de um Modelo de *Features*:** A DyMMer torna possível visualizar e analisar um modelo de *features* específico, previamente importado. Para os mode-

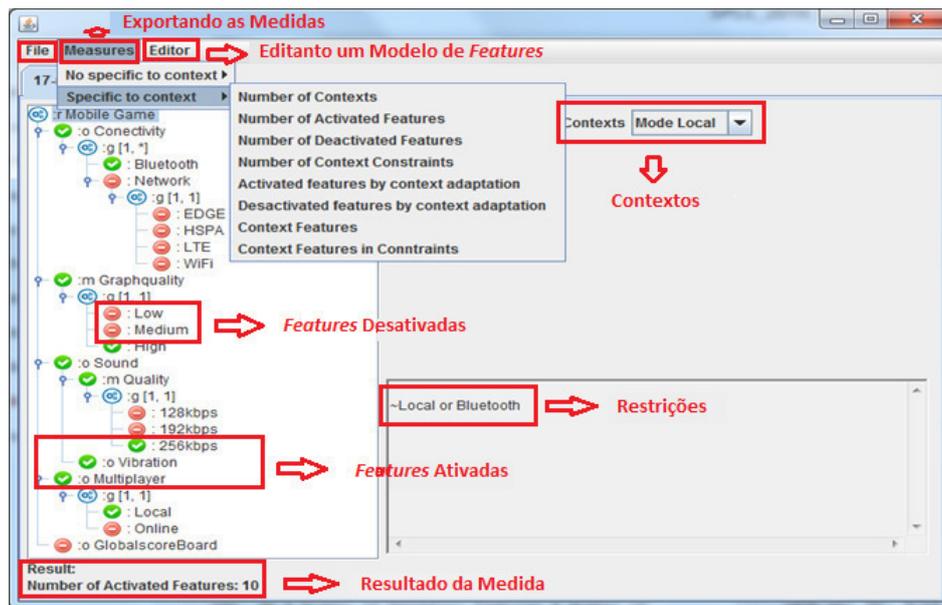


Figura 2. Visão geral da Ferramenta DyMMER

los de *features* que já possuem adaptações de contextos, é possível visualizar, para cada contexto, as *features* ativadas e desativadas, de acordo com suas restrições. Além de visualizar a estrutura do modelo de *features*, a ferramenta DyMMer permite, por meio da aba de medidas, selecionar um subconjunto das medidas de qualidades suportadas, computar e visualizar os valores dessas medidas.

- **Edição de um Modelo de *Features*:** Além possibilitar a importação e visualização de modelos de *features*, a ferramenta DyMMer permite também a edição desses modelos, tornando possível, por exemplo, adicionar informações de contexto (em modelos de *features* sem contexto), por meio da adição ou remoção de *features* ou restrições de contexto, ou ainda pela inclusão de informações sobre a ativação e desativação de *features* de contexto. Vale destacar que em LPSDs um modelo de *features* pode conter um ou mais contextos. Esta funcionalidade possibilita que o engenheiro de domínio possa lidar com modelos de *features* de LPSDs, já que o S.P.L.O.T. não suporta a modelagem deste tipo de modelo. A Figura 2 ilustra a edição de um modelo de *features* na ferramenta DyMMer.
- **Exportação de Medidas:** Na DyMMer é possível exportar, de forma automática, os valores das medidas de qualidade para uma planilha no formato *Microsoft Office Excel*. O engenheiro de domínio pode exportar todas as medidas de uma só vez para um ou mais modelos de *features*. Essa é uma grande vantagem da ferramenta DyMMer, pois possibilita a análise de vários modelos de *features* em conjunto. A DyMMer permite também que o engenheiro de domínio exporte apenas um subconjunto das 40 medidas de qualidade suportadas.

6. Datasets

Neste trabalho foram desenvolvidos três *datasets* de medidas de qualidade para avaliação do modelo de *features*, denominados: MACchiATO, AFFOgaTO e ESPREssO. Esses *datasets* foram construídos a partir das medidas do catálogo COFFEE e de um conjunto de modelos de *features* extraídos do repositório S.P.L.O.T [Mendonca et al. 2009] e da li-

teratura. Com a finalidade de apoiar a construção destes três *datasets*, foi desenvolvida uma ferramenta, denominada DyMMer. A ferramenta DyMMer possibilita a edição de modelos de *features* e a extração automática das 40 medidas de qualidade que compõem o catálogo COFFEE.

MACchiATO (*MeAsures dATaset for feaTure mOdel*) - tem a finalidade de suportar a avaliação da qualidade de modelos de *features* de LPSs. O *dataset* MACchiATO contém os valores de 32 medidas pertencentes ao catálogo COFFEE computadas para 218 modelos de *features* de LPSs extraídos do repositório S.P.L.O.T. Essas 32 medidas foram selecionadas por não envolverem informações de contexto, uma vez que o objetivo deste *dataset* é auxiliar a avaliação de modelos de *features* de LPSs, e não de LPSDs. O *dataset* MACchiATO³ está disponível de forma *online* e pode ser utilizado gratuitamente pela comunidade de engenharia de software.

AFFOgaTO (*dAtaset For the Feature mOdel evoluTiOn*) - tem por finalidade possibilitar o estudo do impacto do processo de evolução dos modelos de *features* na qualidade destes modelos. Por este motivo, o *dataset* AFFOgaTO é composto por: (i) um conjunto de 16 modelos de *features*, com suas respectivas versões, obtidos a partir do repositório S.P.L.O.T.; e (ii) uma planilha compilada contendo os valores de 21 medidas estruturais, para cada versão de cada um dos 16 modelos selecionados. O *dataset* AFFOgaTO⁴ está disponível para *download* e pode ser usado livremente pela comunidade de engenharia de software.

ESPREssO (*mEasures dataSet for dsPl featuRE mOdel*) - foi concebido para suportar a avaliação da qualidade de modelos de *features* de LPSDs. O *dataset* ESPREssO⁵ está disponível *online* e pode ser utilizado gratuitamente pela comunidade de engenharia de software. O *dataset* ESPREssO é composto por: (i) um conjunto de 30 modelos de *features* de LPSDs, extraídos da literatura; e (ii) uma planilha compilada contendo os valores de 13 medidas de qualidade, para cada um dos 30 modelos selecionados. As 13 medidas que compõem o *dataset* ESPREssO foram selecionadas a partir do catálogo COFFEE, sendo 9 medidas voltadas para LPSs e 4 específicas para LPSD.

7. Estudo de Caso

Um dos objetivos deste estudo de caso exploratório é investigar como as medidas de qualidade do catálogo COFFEE podem ser aplicadas com a finalidade de suportar a avaliação da qualidade do modelo de *features* em LPSs. Para isso, foi utilizado o *dataset* MACchiATO. O *dataset* MACchiATO contém os valores de 32 medidas pertencentes ao catálogo COFFEE computadas para 218 modelos de *features* de LPSs extraídos do repositório S.P.L.O.T. Essas 32 medidas foram selecionadas por poderem ser computadas para LPSs, foco deste estudo de caso. Esse estudo foi publicado em [Bezerra et al. 2016a], e os detalhes da execução do estudo podem ser consultados no artigo.

Para atingir o objetivo proposto, foram definidas três questões de pesquisa:

- **QP1:** Existem correlações entre as medidas utilizadas para avaliar a qualidade dos modelos de *features* em LPSs? Respondendo a QP1, será possível reduzir

³<http://carlabezerra.great.ufc.br/macchiato>

⁴<https://goo.gl/gye5ma>

⁵<https://goo.gl/ONfTL3>

a quantidade de medidas a serem utilizadas para avaliar a manutenibilidade de modelos de *features*.

- **QP2:** Como agrupar as medidas relacionadas? Respondendo a QP2, será possível agrupar medidas que cobrem aspectos semelhantes e, conseqüentemente, reduzir a quantidade de medidas a serem utilizadas para avaliar a manutenibilidade de modelos de *features*.
- **QP3:** Como definir *thresholds* para uma determinada medida? Responder a QP3 é essencial para permitir que as medidas sejam utilizadas na prática, ou seja, em cenários reais.

7.1. Seleção dos Objetivos

A fim de especificar o contexto para a realização deste estudo de caso, foram selecionadas as medidas de qualidade e os modelos de *features* a serem utilizados. Esta seleção foi baseada no método de amostragem de conveniência [Wohlin et al. 2012], que é um tipo de técnica de amostragem não probabilística com base no julgamento do pesquisador. Este método foi utilizado devido ao fato das medidas serem selecionadas com base em revisões da literatura e os modelos de *features* terem sido extraídos de um repositório público. Assim, o *dataset* MACchiATO foi concebido por meio da utilização deste método.

7.2. Procedimentos para Coleta e Análise de Dados

Neste estudo de caso, foi utilizada uma combinação de dados qualitativos e quantitativos, que é conhecido como método misto [Runeson 2009]. A análise dos dados é executada de forma distinta para os dados quantitativos e qualitativos. A análise dos dados quantitativos, em geral, envolve métodos baseados em estatística descritiva e análise de correlação. Os métodos de estatística descritiva envolvem médias, desvios-padrão, histogramas e gráficos de dispersão, com a finalidade de facilitar a compreensão dos dados coletados. A análise de correlação é realizada com o objetivo de descrever a forma como uma medida se relaciona com outra. Para os dados qualitativos, o principal objetivo da análise é extrair conclusões, mantendo uma clara cadeia de provas. Isto é, um leitor deve ser capaz de seguir a derivação dos resultados e chegar às conclusões a partir dos dados coletados. Além disso, os avanços das técnicas de aprendizagem de máquina e análise de dados têm fornecido meios eficazes de extrair informações úteis e conhecimento a partir dos dados. Neste estudo de caso exploratório foram utilizadas diferentes técnicas de análise de dados, tais como: estatística descritiva, análise de correlação, PCA (Análise de Componentes Principais), além de métodos paramétricos e não paramétricos para definir os *thresholds*.

7.3. Resultados

Para responder as questões de pesquisa do estudo de caso, foram obtidos os resultados detalhados a seguir.

QP1 - Para responder essa questão de pesquisa, foi realizada a análise de correlação utilizando o coeficiente de correlação de *Spearman* (com nível de significância de 0.05) para as 32 medidas do *dataset* MACchiATO, normalizando os dados. A análise da correlação entre duas medidas é importante para descobrir se estas cobrem aspectos similares do modelo de *features*. Como resultado da análise de correlação, pode-se argumentar que não é necessário o uso das 32 medidas, a fim de avaliar a manutenibilidade

do modelo *feature*. Seria suficiente usar o conjunto composto por 15 medidas: NF, NM, NLeaf, NTop, DTMax, CogC, FEX, FoC, SCDF, MCDF, RDen, RoV, NVC, NGOR e NGXOr.

QP2 - Para responder essa questão de pesquisa, foi utilizada neste trabalho a técnica de agrupamento denominada Análise de Componentes Principais (PCA), por ser uma técnica bastante utilizada pela comunidade acadêmica para agrupamento de dados. Com o PCA, um número de combinações de medidas não correlacionadas são selecionadas para capturar informações sobre o modelo de *features* como um todo. PCA é um procedimento matemático que utiliza transformação ortogonal para converter um conjunto de variáveis (dimensões), possivelmente correlacionadas, para um conjunto de variáveis não correlacionadas linearmente chamadas de componentes principais (PCs). Como resultado da execução do PCA foram selecionados 9 PCs selecionados para as 15 medidas de qualidade selecionadas na análise de correlação. Cada PC representa o agrupamento das 15 medidas, compondo as novas medidas agrupadas (9 PCs).

QP3 - Por último, para responder a QP3, foram definidos *thresholds* para cada um dos 9 PCs selecionados. Para definir os *thresholds*, foi utilizada a técnica de intervalo de tolerância. Em estatística, um intervalo de tolerância é um intervalo estatístico em que uma determinada proporção de uma amostra de população cai com algum nível de confiança. Assim, neste trabalho foi utilizado um intervalo de tolerância de 95%, que é um valor típico para problemas semelhantes [Altman et al. 2013]. Foram definidos o limite superior, a média e o limite inferior de cada PC. Esses *thresholds* podem ajudar a avaliações de qualidade inicial dos modelos de *features* em LPSs. No entanto, um valor diferente pode ser escolhido pelo engenheiro de domínio (especialista em LPS).

7.4. Implicações para os Pesquisadores e Engenheiros de Domínio

Os resultados do estudo mostraram que as correlações entre as medidas de qualidade existem. Assim, nem todas as medidas são necessárias para revelar as características de qualidade de um modelo de *features*. Além disso, as constatações obtidas neste trabalho indicam que a técnica de PCA pode ser usada para construção de novas medidas agrupadas mais representativas do que as medidas individuais. Finalmente, se a inspeção manual nos modelos de *features* para definir *thresholds* não é possível, estratégias centradas em dados podem ser aplicadas. Entretanto, se os *thresholds* não estão devidamente definidos, é difícil realmente saber se um determinado valor da medida indica um problema potencial no modelo de *features*.

Uma vez que tenha sido estabelecido que as correlações entre as medidas de qualidade existem, acredita-se que as abordagens de seleção de característica podem ser usadas para selecionar um subconjunto de medidas relevantes de qualidade (também conhecidas em aprendizagem de máquina e estatísticas como característica, subcaracterística ou variável), a fim de construir um modelo (por exemplo, um modelo de previsão). As abordagens de seleção de característica são utilizados por três razões: simplificação de modelos, menor tempo de treinamento e redução à sobreposição. A premissa principal quando se utiliza uma abordagem de seleção de características é que os dados contém muitas características que são redundantes ou irrelevantes, e assim pode ser removido sem muita perda de informações, que foi exatamente o que aconteceu no estudo.

O processo de projeto e execução do trabalho de investigação necessário para

este estudo resultou em importantes reflexões sobre muitos aspectos da realização da investigação empírica sobre o campo de LPSs. O primeiro desafio enfrentado foi obter um conjunto de modelos de *features* de LPSs bem estabelecidas e acessíveis ao público. Foi encontrado apenas um repositório público, o S.P.L.O.T. No entanto, a maioria dos modelos de *features* compartilhados foram concebidos para fins acadêmicos e de investigação. É fundamental reunir os modelos de *features* industriais e reais dentro desses repositórios para fortalecer estudos empíricos futuros e avaliações. Para trabalhos futuros, o ideal seria selecionar modelos de *features* em escala real e industriais. O segundo desafio foi no que diz respeito à complexidade das medidas estruturais. Os resultados apoiam a ideia de que medidas simples, que são correlacionadas com medidas complexas, são bastante úteis para indicar atributos de qualidade externos. O terceiro desafio está relacionado com a ausência do grau de qualidade nos modelos de *features*, em outras palavras, se cada modelo de *features* tivesse, pelo menos, uma indicação de qualidade como boa ou má, seria possível usar algoritmos mais sofisticados de aprendizagem de máquina para definir os *thresholds* das medidas ou classificar novos modelos de *features*.

Por último, de acordo com os resultados deste trabalho, não é possível provar que a melhoria da qualidade dos modelos de *features* implica em uma melhor reutilização de software. Para isso, é necessário monitorar a qualidade e o nível de reutilização de LPSs reais. Além disso, este trabalho não forneceu diretrizes para melhoria da qualidade dos modelos de *features*. Estes aspectos serão investigados em trabalhos futuros.

7.5. Ameaças à Validade

Validade Interna. Para aumentar a validade interna, foram coletadas todas as medidas automaticamente utilizando a ferramenta DyMMer. Além disso, foi realizada a análise de correlação seguindo um processo de cálculo padrão do coeficiente de correlação de *Spearman* e foram escolhidas apenas os resultados com significância estatística ($p < 0,05$) para análise. No entanto, não se pode garantir que os resultados da análise do experimento deste estudo dependem de definições específicas das medidas de qualidade no catálogo COFEE. O repositório S.P.L.O.T. possui muitos modelos de *features* que foram projetados para fins acadêmicos e de investigação (exemplos fictícios). Este fator pode influenciar nos *thresholds* definidos. Além disso, as medidas no catálogo COFEE possuem diferentes distribuições probabilísticas.

Validade de Construção. A validade de construção está preocupada com a relação entre a teoria e a observação. Neste contexto, a principal preocupação do estudo é a redução do número de medidas de qualidade executadas na etapa de análise de correlação, a partir das 32 medidas iniciais no catálogo COFEE para 15 medidas. Este conjunto resultante de 15 medidas de qualidade foi usado como entrada na etapa PCA. Este processo de seleção foi realizado manualmente e com base nas subcaracterísticas qualidade. O resultado de um conjunto diferente de medidas resultou em diferentes componentes principais.

Validade Externa. Os resultados do estudo de caso válido externamente podem ser generalizados e aplicados com segurança para a prática de engenharia de software e serem recomendados como padrões para avaliação da qualidade do modelo de *features*. Para mitigar esta ameaça, foi utilizado um grande conjunto de modelos de *features*, incluindo modelos de *features* com diferentes tamanhos e complexidades. Além disso, todas as medidas de qualidade utilizadas neste trabalho foram extraídas a partir da literatura

existente. No entanto, os resultados dos experimentos deste estudo não são automaticamente transferíveis para todos os outros *datasets* de modelos de *features*.

Validade de Conclusão. A validade de conclusão é a extensão em que as conclusões sobre a presença de uma relação estatisticamente significativa entre os tratamentos e os resultados são válidos. Para mitigar as ameaças à validade da conclusão e aumentar a confiabilidade deste estudo de caso exploratório, foi utilizada a técnica de validação cruzada, com 10 rodadas. Esta validação é importante para avaliar como a abordagem proposta vai generalizar para um *dataset* independente.

8. Conclusões

Os principais resultados desta pesquisa de doutorado foram: (i) um catálogo de medidas de qualidade denominado COFFEE, o qual contém 40 medidas relacionadas à manutenibilidade do modelo de *features*; (ii) uma ferramenta, denominada DyMMer, para suportar a avaliação da qualidade do modelo de *features*, possibilitando a coleta automática das medidas pertencentes ao catálogo COFFEE e a modelagem das informações de contexto utilizadas nos modelos de *features* de LPSDs; e (iii) a construção de três *datasets*, denominados AFFOgaTO, MACchiATO e ESPREssO, que podem ser utilizados para auxiliar a avaliação da manutenibilidade de modelos de *features*. Para avaliação das medidas de manutenibilidade do modelo de *features* foi realizado um estudo de caso para investigar a aplicação das medidas voltadas para LPSs, por meio da análise da correlação entre essas medidas, agrupamento dessas medidas e da definição de *thresholds* para as medidas.

Como trabalhos futuros foram identificadas as seguintes oportunidades de pesquisa: definir uma abordagem de avaliação da qualidade do modelo de *features*; estender o catálogo COFFEE para outras características, subcaracterísticas e medidas; construir um *dataset* com modelos de *features* de LPSs e LPSDs reais; elaborar diretrizes para melhoria do modelo de *features*; e utilizar técnicas de aprendizagem de máquina para classificação dos modelos de *features*.

Referências

- Alferez, G. H., Pelechano, V., Mazo, R., Salinesi, C., and Diaz, D. (2014). Dynamic adaptation of service compositions with variability models. *Journal of Systems and Software*, 91:24–47.
- Altman, D., Machin, D., Bryant, T., and Gardner, M. (2013). *Statistics with confidence: confidence intervals and statistical guidelines*. John Wiley & Sons.
- Asadi, M., Gröner, G., Mohabbati, B., and Gašević, D. (2016). Goal-oriented modeling and verification of feature-oriented product lines. *Software & Systems Modeling*, 15(1):257–279.
- Bagheri, E. e Gasevic, D. (2011). Assessing the maintainability of software product line feature models using structural metrics. *Software Quality Control*, 19(3):579–612.
- Benavides, D., Segura, S., and Ruiz-Cortés, A. (2010). Automated analysis of feature models 20 years later: A literature review. *Information Systems*, 35(6):615–636.
- Benavides Cuevas, D. F., Segura Rueda, S., Trinidad Martín Arroyo, P., and Ruiz Cortés, A. (2007). Fama: Tooling a framework for the automated analysis of feature models.

- Berger, T. e Guo, J. (2014). Towards system analysis with variability model metrics. In *Proceedings of the Eighth International Workshop on Variability Modelling of Software-Intensive Systems*, page 23. ACM.
- Bezerra, C. I., Andrade, R. M., and Monteiro, J. M. (2016a). Exploring quality measures for the evaluation of feature models: A case study. *Journal of Systems and Software*.
- Bezerra, C. I., Barbosa, J., Freires, J. H., Andrade, R., and Monteiro, J. M. (2016b). Dymmer: a measurement-based tool to support quality evaluation of dspl feature models. In *Proceedings of the 20th International Systems and Software Product Line Conference*, pages 314–317. ACM.
- Bezerra, C. I., Monteiro, J. M., Andrade, R., and Rocha, L. S. (2016c). Analyzing the feature models maintainability over their evolution process: An exploratory study. In *Proceedings of the Tenth International Workshop on Variability Modelling of Software-intensive Systems*, pages 17–24. ACM.
- Bezerra, C. I., Monteiro, J. M. S., and Andrade, R. (2013). Avaliação da qualidade do modelo de *Features* em linhas de produto de software utilizando medidas. In *Simpósio Brasileiro de Qualidade de Software*, page 15.
- Bezerra, C. I. M., Andrade, R. M. C., and Monteiro, J. M. (2015). Measures for quality evaluation of feature models. In *Software Reuse for Dynamic Systems in the Cloud and Beyond - 14th International Conference on Software Reuse, ICSR 2015, Miami, FL, USA, January 4-6, 2015. Proceedings*, pages 282–297.
- Capilla, R. and Bosch, J. (2011). The promise and challenge of runtime variability. *Computer*, 44(12):93–95.
- Capilla, R., Bosch, J., and Kang, K.-C. (2013). *Systems and software variability management*. Springer.
- Capilla, R., Bosch, J., Trinidad, P., Ruiz-Cortés, A., and Hinchey, M. (2014). An overview of dynamic software product line architectures and techniques: Observations from research and industry. *Journal of Systems and Software*, 91:3–23.
- Cetina, C., Giner, P., Fons, J., and Pelechano, V. (2009). Using feature models for developing self-configuring smart homes. In *Autonomic and Autonomous Systems, 2009. ICAS'09. Fifth International Conference on*, pages 179–188. IEEE.
- Etxeberría, L. and Sagardui, G. (2008a). Quality assessment in software product lines. In *High Confidence Software Reuse in Large Systems*, pages 178–181. Springer.
- Etxeberría, L. and Sagardui, G. (2008b). Variability driven quality evaluation in software product lines. In *Software Product Line Conference, 2008. SPLC'08. 12th International*, pages 243–252. IEEE.
- Hallsteinsen, S., Hinchey, M., Park, S., and Schmid, K. (2008). Dynamic software product lines. *Computer*, 41(4):93–95.
- ISO/IEC (2014). *Iso/iec 25000 - systems and software engineering – systems and software quality requirements and evaluation (square) – guide to square*. Technical report.
- Kim, T., Ko, I. Y., Kang, S. W., and Lee, D. H. (2008). Extending atam to assess product line architecture. In *Computer and Information Technology. CIT 2008. 8th IEEE International Conference on*, pages 790–797. IEEE.

- Lochau, M., Bürdek, J., Hölzle, S., and Schürr, A. (2015). Specification and automated validation of staged reconfiguration processes for dynamic software product lines. *Software & Systems Modeling*, pages 1–28.
- Marinho, F. G., Maia, P. H., Andrade, R., Vidal, V. M., Costa, P. A., and Werner, C. (2012). Safe adaptation in context-aware feature models. In *Proceedings of the 4th International Workshop on Feature-Oriented Software Development*, pages 54–61. ACM.
- Mendonca, M., Branco, M., and Cowan, D. (2009). S.p.l.o.t.: Software product lines online tools. In *Proceedings of the 24th ACM SIGPLAN Conference Companion on Object Oriented Programming Systems Languages and Applications, OOPSLA '09*, pages 761–762, New York, NY, USA. ACM.
- Montagud, S., Abrahão, S., and Insfran, E. (2012). A systematic review of quality attributes and measures for software product lines. *Software Quality Journal*, 20(3-4):425–486.
- Montagud, S. and Abrahão, S. (2009). Gathering current knowledge about quality evaluation in software product lines. In *Proceedings of the 13th International Software Product Line Conference, SPLC '09*, pages 91–100, Pittsburgh, PA, USA. Carnegie Mellon University.
- Oliinyk, O., Petersen, K., Schoelzke, M., Becker, M., and Schneickert, S. (2015). Metrics for the evaluation of feature models in an industrial context: A case study at opel. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*, pages 33–48. Springer.
- Olumofin, F. G. and Mišić, V. B. (2007). A holistic architecture assessment method for software product lines. *Information and Software Technology*, 49(4):309–323.
- Pascual, G. G., Lopez-Herrejon, R. E., Pinto, M., Fuentes, L., and Egyed, A. (2015). Applying multiobjective evolutionary algorithms to dynamic software product lines for reconfiguring mobile applications. *Journal of Systems and Software*, 103:392–411.
- Runeson, P. e Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering*, 14(2):131–164.
- Saller, K., Lochau, M., and Reimund, I. (2013). Context-aware dspls: model-based runtime adaptation for resource-constrained systems. In *Proceedings of the 17th International Software Product Line Conference co-located workshops*, pages 106–113. ACM.
- Thörn, C. (2010). On the quality of feature models.
- Thüm, T., Apel, S., Kästner, C., Schaefer, I., and Saake, G. (2014). A classification and survey of analysis strategies for software product lines. *ACM Computing Surveys (CSUR)*, 47(1):6.
- Trinidad, P., Benavides, D., Durán, A., Ruiz-Cortés, A., and Toro, M. (2008). Automated error analysis for the agilization of feature modeling. *Journal of Systems and Software*, 81(6):883–896.
- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). *Experimentation in software engineering*. Springer Science & Business Media.