ANDRESSA IANZEN

DEFINIÇÃO DE ESCOPO EM LINHAS DE PRODUTO DE SOFTWARE: UMA ABORDAGEM SEMIAUTOMÁTICA UTILIZANDO ANOTAÇÃO LINGUÍSTICA

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática.

Curitiba
2013

ANDRESSA IANZEN

DEFINIÇÃO DE ESCOPO EM LINHAS DE PRODUTO DE SOFTWARE: UMA ABORDAGEM SEMIAUTOMÁTICA UTILIZANDO ANOTAÇÃO LINGUÍSTICA

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática.

Área de Concentração: Ciência da Computação

Orientadora: Prof. Drª Sheila Reinehr
Co-Orientadora: Prof. Drª Andreia Malucelli

Curitiba
2013

FICHA CATALOGRÁFICA

Aos meus pais e à minha irmã, que acreditaram e lutaram pela minha entrada na faculdade.

AGRADECIMENTOS

À minha mãe, pai e irmã, por todo o seu amor, carinho, dedicação e apoio. Por toda sua paciência, confiança, pelos incentivos e pela luta que partilharam comigo desde o início.

À minha sobrinha por me distrair e divertir nos momentos de descanso.

Ao Marcelo, por me incentivar a continuar sempre e superar todos os obstáculos e os meus próprios limites.

A todos os meus amigos, principalmente do grupo de mestrado: Joselaine, Kelly, João e Mauda. A amizade e companheirismo de vocês foram essenciais.

E principalmente às minhas orientadoras pela paciência, confiança e todo o tempo dispensado em orientações e revisões.

"A coisa mais indispensável a um homem é reconhecer o uso que deve fazer do seu próprio conhecimento."

- Platão

RESUMO

Linha de Produto de Software (LPS) é uma abordagem de desenvolvimento de software que foca o desenvolvimento com reuso e para reuso. Para a identificação dos ativos reutilizáveis que devem ser desenvolvidos é essencial a realização da atividade de definição de escopo. Esta atividade tem como objetivo mapear o escopo da LPS, identificando e delimitando os produtos, funcionalidades e áreas do domínio que devem fazer parte da LPS, assim como suas funcionalidades comuns e variáveis. A linha de produtos de software pode ser criada de diferentes maneiras, sendo uma delas a partir de sistemas de software existentes. Nesse caso, a atividade de definição de escopo deve se utilizar do conhecimento sobre os produtos que fazem parte destes sistemas de software. As abordagens para a realização de definição de escopo são dependentes do conhecimento do domínio dos produtos. Por este motivo, há a necessidade da participação de especialistas de domínio, porém estes profissionais normalmente não possuem muito tempo livre disponível, o que pode causar atrasos nesta fase. Sendo assim, seria interessante uma abordagem que diminuísse a necessidade da presença destes especialistas. Se for possível semi-automatizar a identificação e classificação das funcionalidades dos sistemas existentes, existe a possibilidade de diminuir ainda mais a necessidade de presença constante do especialista de domínio nesta atividade. Neste contexto, pretende-se identificar nesta pesquisa se é possível semi-automatizar a identificação e a classificação das funcionalidades de sistemas existentes - diminuindo a necessidade da presença constante do especialista de domínio durante a definição de escopo.

Palavras-chaves: Definição de escopo, linhas de produto de software (LPS), abordagem semiautomática, anotação linguística.

ABSTRACT

Software Product Line is a software development approach which focuses on the development with reuse and for reuse. To identify the reuse assets that should be developed, it is necessary to perform an activity called product line scoping. This activity aims at mapping the scope of the product line as a whole, identifying and delimitating the products, characteristics and domain areas that should be part of the product line, and also the communalities and variabilities. The software product line can be created in different ways, one of them from existent legacy systems. In this case, the scoping activity must use the knowledge about the products of these software systems. The approaches for scoping are dependent of the knowledge about the product's domain. Therefore, it is necessary to have the domain expert participation, however this professional often do not have too much free time, which may cause delay in this phase. So, it would be interesting to have an approach that could decrease the need of domain expert's presence. If it is possible to semi-automate the identification and classification of functionalities of existing systems, there is a possibility to decrease the necessity for constant presence of the domain expert in this activity. In this context, it is intended to identify in this research whether it is possible to semi-automate the identification and classification of the functionalities of an existing system – decreasing the need for constant presence of the domain expert.

Keywords: Product line scoping, software product lines, semi-automatic approach, linguistic anotation.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ACM	Association for Computing Machinery
CAVE	Commonality and Variability extraction
CO4	Cooperative Construction of Ontologies
COMA	COmbination of Matching Algorithms
CRCTOL	Concept-Relation-Concept Tuple-based Ontology Learning
Cyc KB	Cycorp Knowledge Base
DAML	DARPA Agent Markup Language
DOLCE	Ontology for Linguistic and Cognitive Engineering
EFP	Engenharia de Famílias de Produto
HTML	HyperText Markup Language
KACTUS	Knowledge About Complex Technical systems for multiple Use
KIF	Knowledge Interchange Format
LD	Lógica descritiva
LPS	Linhas de Produto de Software
NOM	Naive Ontology Mapping
OCML	Ontology Compositional Modelling Language
OIL	Ontology Inference Layer
OKCB	Open Knowledge Based Connectivity
OLA	OWL-Lite Aligner
OWL	Web Ontology Language
P2P	Peer-to-peer
PuLSE	Product Line Software Engineering
PulSE-Eco	Product Line Software Engineering – Economic Scoping
QOM	Quick Ontology Mapping

RDF	Resource Description Framework
SAT	Satisfazibilidade booleana / proposicional
SEI	Software Engineering Institute
Sis	Sistemas de Informação
SUMO	Suggested Upper Merged Ontology
TFIDF	Term frequency x inverted document frequency
XCES	Corpus Encoding Standard for XML
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language
W3C	World Wide Web Consortium

CAPÍTULO 1 - INTRODUÇÃO

"A parte que ignoramos é muito maior que tudo quanto sabemos."

-Platão

A necessidade por Sistemas de Informação (SIs) que auxiliem a executar tarefas rapidamente e com maior qualidade cresce cada vez mais. Há bastante tempo, o papel do software nas empresas é reconhecido como uma parte essencial para seu sucesso: "melhorar performance de negócio geralmente significa que as empresas precisam melhorar drasticamente a performance do seu desenvolvimento de software" (JACOBSON et al., 1997).

Construir software com qualidade e desempenho é o foco da Engenharia de Software, que segundo (SOMMERVILLE, 2007), desenvolve SIs de alta qualidade com custos adequados e se relaciona com todos os aspectos da produção de software.

Segundo (GIMENES; TRAVASSOS, 2002), para obter produtos de alta qualidade e economicamente viáveis é necessário um conjunto sistemático de ferramentas, processos e técnicas, sendo o reuso uma das técnicas mais relevantes nesse conjunto. Ainda, segundo os autores, o enfoque da Linha de Produto de Software (LPS) surge como uma proposta de construção sistemática de software.

De acordo com (CLEMENTS, 2002), a LPS permite que a organização ganhe vantagem competitiva, pois gera o aumento da qualidade, redução dos custos, redução do tempo de entrega e minimização dos riscos dos produtos. Em (LINDEN et al., 2007), são apresentados diversos estudos de caso sobre a utilização de LPS. Alguns exemplos que podem ser citados e demonstram vantagens na utilização desta abordagem são:

- a empresa *Philips Medical Systems* conseguiu uma diminuição do *time-to-market* entre 25% e 50%, assim como redução no esforço para construção de componentes reutilizáveis;
- na *Siemens*, a aceleração dos testes após a implantação parcial de uma linha de produtos foi de aproximadamente 75%;

- na empresa *AKVAsmart,* houve uma redução média do tamanho dos códigos fonte em aproximadamente 70%.

A abordagem de LPS pode ser dividida em duas atividades, segundo (LINDEN et al., 2007), a engenharia de domínio e a engenharia de aplicação. Na engenharia de domínio são identificados e desenvolvidos os ativos que serão reutilizados pelos produtos da LPS e sua arquitetura. Na engenharia de aplicação, os ativos desenvolvidos são selecionados para a criação de determinados produtos.

Os ativos reutilizáveis podem ser, por exemplo, componentes de software. Estes componentes serão utilizados para a criação dos produtos da linha durante a engenharia de aplicação. Por exemplo, no domínio de celulares, poderiam existir os ativos para “realizar chamada”, “enviar mensagem” e “navegar na internet”. Ao criar um produto, pode ser selecionar os ativos que farão parte deste, excluindo desta seleção, por exemplo, o ativo “navegar na internet”.

Para a identificação dos ativos reutilizáveis que devem ser desenvolvidos, é essencial a realização da atividade chamada definição de escopo. Esta atividade é geralmente o primeiro passo a ser realizado ao iniciar a construção da LPS e tem como objetivo mapear o escopo da LPS, identificando e delimitando os produtos, funcionalidades e áreas do domínio que devem fazer parte da LPS, assim como suas funcionalidades comuns e variáveis (JOHN, 2009).

A correta definição de escopo implicará no sucesso da LPS. Se o escopo definido for muito grande, os ativos de software desenvolvidos terão que acomodar muitas variações e ficarão muito complexos para serem úteis. Se for muito pequeno, os ativos de software podem não ser desenvolvidos de forma genérica o suficiente para acomodar crescimento futuro. Neste caso, a oportunidade de incluir novos produtos na LPS será rejeitada por estar fora do escopo, ou, se aceita, resultará em retrabalho. Se o escopo definido abranger os produtos errados, a LPS não encontrará um segmento de mercado para atender (SEI, 2005).

As abordagens para que uma organização inicie o uso das LPS podem ser classificadas em pró-ativas, reativas e extrativas (ALVES et al., 2010) :

- na abordagem do tipo pró-ativa, a base comum de ativos (*assets*) reutilizáveis é desenvolvida primeiro e os produtos são desenvolvidos utilizando os ativos.

- na abordagem reativa, a LPS é iniciada com um ou alguns poucos produtos, sendo incrementada na medida em que novos produtos são necessários.
- a abordagem extrativa utiliza um ou mais sistemas de software existentes para criar a base de ativos comuns.

Quando da utilização da abordagem extrativa, a atividade de definição de escopo deve se utilizar do conhecimento sobre os sistemas existentes que serão utilizados para criação da LPS (ALVES et al., 2010) . Para isto, as abordagens para definição de escopo focam na participação de especialistas de domínio, porém estes profissionais normalmente não possuem muito tempo disponível, o que pode causar atrasos nesta fase. Sendo assim, é necessária uma abordagem que diminua a necessidade da presença dos especialistas de domínio (JOHN, 2006).

Algumas iniciativas estão sendo desenvolvidas neste sentido, como por exemplo, a proposta de (JOHN, 2010), que utiliza a documentação de usuário como fonte para o processo de definição de escopo. Os documentos são analisados por um consultor de linhas de produto que busca alguns itens como palavras e frases que podem ser identificados como funcionalidades ou itens relacionados ao domínio. Esta abordagem utiliza padrões de busca pré-estabelecidos, que são feitos manualmente, portanto estão sujeitos a erros humanos. Depois do processamento manual dos documentos, um especialista de domínio é necessário para realizar a validação do que foi encontrado. Um dos trabalhos futuros citados em (JOHN, 2010) é a automação dos padrões para prover análise automática de documentos e identificação de artefatos para a linha de produtos, pois a tarefa de aplicar manualmente padrões em documentos grandes é uma tarefa maçante.

Portanto, identifica-se a necessidade de automatizar ainda mais a atividade de definição de escopo realizada ao se criar uma LPS de maneira extrativa, ou seja, a partir de sistemas existentes. Se for possível semi-automatizar a identificação e classificação das funcionalidades dos sistemas existentes, existe a possibilidade de diminuir ainda mais a necessidade de presença constante do especialista de domínio nesta atividade.

Neste contexto, a questão principal que se pretende responder nesta pesquisa é: “É possível semi-automatizar a identificação e classificação das funcionalidades de sistemas existentes - diminuindo a necessidade da presença

constante do especialista de domínio durante a definição de escopo realizada ao se criar uma LPS de maneira extrativa?”.

1.1 Objetivos

O objetivo geral do trabalho é **desenvolver uma abordagem semiautomática para auxiliar a definição de escopo de LPS.** Para atender ao objetivo geral, são definidos os seguintes objetivos específicos:

1- Classificar automaticamente funcionalidades de sistemas existentes;
2- Definir o escopo da LPS por meio da análise manual das funcionalidades classificadas automaticamente;
3- Avaliar automaticamente novos produtos em relação ao escopo definido da LPS;
4- Avaliar a abordagem por meio da comparação com a definição de escopo de forma manual.

A abordagem proposta neste trabalho se divide em duas partes: “Definição de escopo” e “Engenharia de produto”. Na “Definição de escopo” pretende-se utilizar artefatos dos produtos da organização para a identificação e classificação semiautomática das funcionalidades. A Figura 1-1 ilustra o fluxo da atividade de “Definição de escopo” proposta neste trabalho.

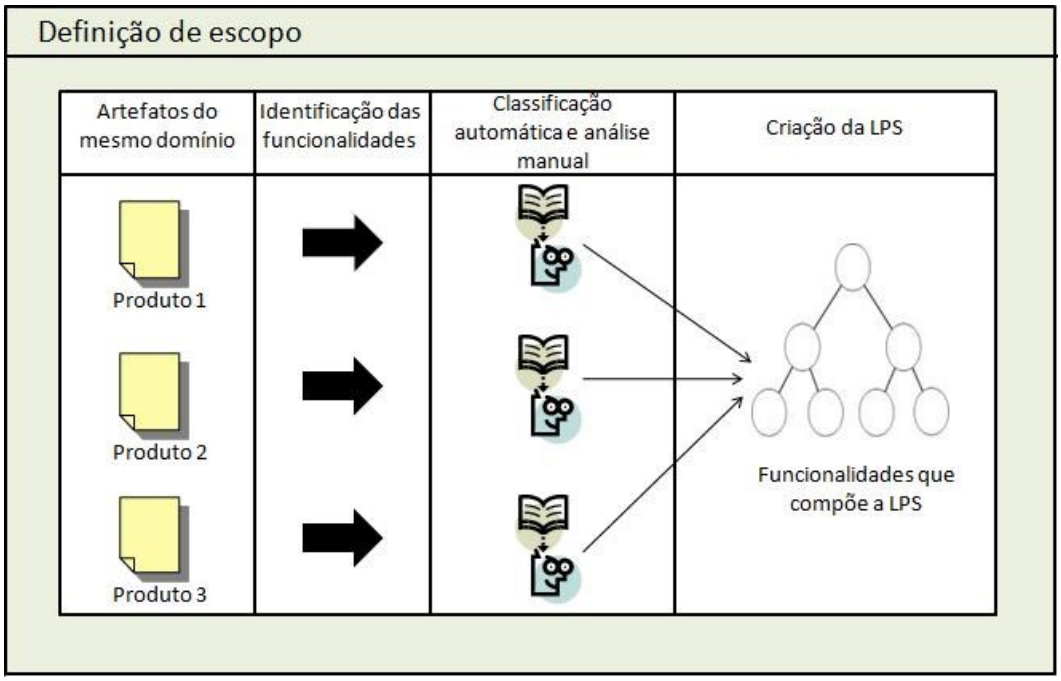

Figura 1-1. Definição de escopo (Fonte: o Autor).

Na “Engenharia de produto” pretende-se identificar as variabilidades e comunalidades entre a LPS criada e um novo produto, apoiando a decisão da inclusão, ou não, do produto à família. A Figura 1-2 ilustra o fluxo da atividade de “Engenharia de produto” proposta neste trabalho.

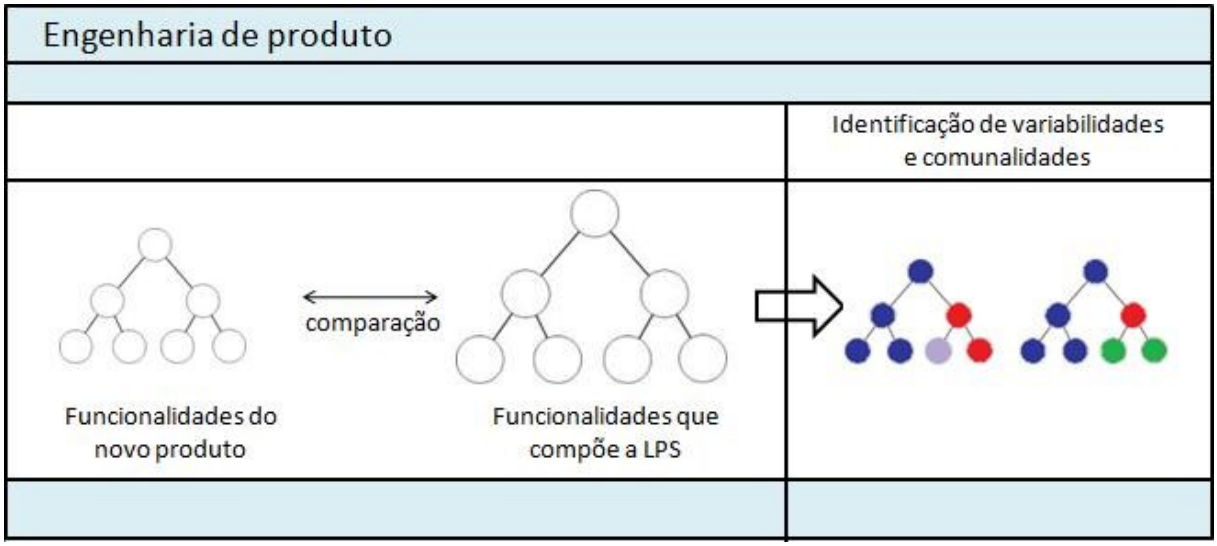

Figura 1-2. Engenharia de produto (Fonte: O Autor).

A abordagem semiautomática proposta pretende auxiliar tanto na criação de uma LPS a partir de um conjunto de artefatos de sistemas existentes que representem as funcionalidades dos produtos que fazem parte da linha, quanto auxiliar, depois da criação da linha, a identificar se um produto novo pode ser considerado integrante da linha, apoiando decisões com relação à implementação deste produto na organização: utilizando ou não a arquitetura da LPS criada.

1.2 Delimitação de escopo

Em (LINDEN et al., 2007) os autores classificam a atividade de definição de escopo em três categorias principais: *product portfolio planning* (planejamento do portfólio de produto)*, domain potential analysis* (análise do potencial do domínio) *e asset scoping* (escopo dos ativos). Na primeira categoria, o objetivo principal é definir os produtos que devem fazer parte da LPS e identificar suas principais funcionalidades; na segunda, o objetivo é avaliar onde os investimentos de reuso devem se focar; na terceira, o objetivo é definir os componentes que serão construídos para reuso.

A abordagem proposta neste trabalho tem o objetivo de apoiar abordagens de definição de escopo da categoria *product portfolio planning* (planejamento do portfólio de produtos), também chamada *product portfolio scoping (*definição do escopo do portfólio de produtos), ou seja, definir os produtos que devem fazer parte da LPS e identificar suas principais funcionalidades. Além disso, o foco da proposta é apoiar a atividade de definição de escopo durante a criação da LPS utilizando abordagens extrativas: abordagens que utilizam sistemas existentes para criação da LPS.

1.3 Processo de trabalho

As seguintes fases foram definidas para a realização dessa pesquisa:

- fase 1 – Preparação da Pesquisa: fase que corresponde à delimitação da área de estudo, coleta e análise das referências bibliográficas, delimitação do tema e estabelecimento dos objetivos, questões e proposições.
- fase 2 – Estruturação da Pesquisa: fase de elaboração de um quadro referencial teórico, seleção do método de pesquisa e do roteiro de pesquisa.
- fase 3 – Execução da Pesquisa: fase da investigação em si, implementação da proposta e realização de experimentos.
- fase 4 - Análise dos Resultados: fase da análise dos dados extraindo as conclusões.

1.4 Estrutura do documento da tese

O capítulo 1 visa apresentar de maneira geral o problema e os objetivos desta pesquisa, contextualizando o leitor. O capítulo 2 apresenta a revisão da literatura sobre os temas principais relacionados ao contexto da pesquisa: Linhas de Produto de Software, Definição de Escopo e Recuperação de Informação. A estruturação da pesquisa é realizada no capítulo 3, com conceitos do método de pesquisa utilizado, caracterização e estratégia de pesquisa. O capítulo 4 apresenta em maiores detalhes a abordagem proposta, e o capítulo 5 finaliza o documento com as considerações finais.

1.5 Considerações sobre o capítulo

Neste capítulo procurou-se apresentar de maneira geral o problema e os objetivos desta pesquisa, assim como a estrutura que será utilizada para realizá-la. No próximo capítulo será apresentada a revisão de literatura onde serão apresentados os principais conceitos e áreas de estudo relacionados com a pesquisa.

CAPÍTULO 2 - REVISÃO DA LITERATURA

"Não há nada que dominemos inteiramente a não ser os nossos pensamentos."
- René Descartes

Este capítulo busca apresentar a fundamentação teórica que será utilizada para tratar o tema e o problema de pesquisa:

- a abordagem linhas de produto de software é tratada na seção 2.1, seguida pelos conceitos de comunalidades e variabilidades na seção 2.1.1 e definição de escopo na 2.1.2.
- na seção 2.2.1, a revisão sistemática que foi realizada para identificar as abordagens mais recentes de definição de escopo é detalhada.
- na seção 2.2 é apresentada uma revisão sobre as abordagens para definição de escopo.
- os conceitos de Recuperação de Informação, Processamento de Linguagem Natural e Anotação Linguística são apresentados na seção 2.3.

2.1 Linhas de Produto de Software (LPS)

Linha de Produto de Software (LPS) é uma abordagem para reuso sistemático de software. Para (KRUEGER, 1992), o reuso de software é o processo de criação de sistemas de software a partir de um software existente, ao invés de construí-lo a partir do zero. O reuso em sistemas de software possibilita o aumento da qualidade e da produtividade nas empresas, pois possibilita a reutilização de ativos já empregados em sistemas anteriores dos quais se conhece a qualidade. Basili e Rombach (BASILI; ROMBACH, 1991) definem o reuso como o uso de tudo o que for associado a um projeto de software, incluindo experiências e conhecimento.

Existem várias maneiras de se trabalhar com o reuso de software. Para (SOMMERVILLE, 2007), "a técnica a ser utilizada para reuso depende dos requisitos do sistema, da tecnologia, dos ativos reusáveis disponíveis e do conhecimento da

equipe de desenvolvimento.”. Isto quer dizer que devem ser avaliadas as abordagens de reuso disponíveis antes de escolher a que se deseja utilizar, de acordo com os objetivos que se deseja alcançar. O Quadro 2-1 apresenta algumas abordagens que apoiam o reuso de software.

Quadro 2-1. Abordagens que apoiam o reuso de software adaptado de (SOMMERVILLE, 2007).

Abordagem	Descrição
Design patterns	Abstrações genéricas que ocorrem ao longo das aplicações são representadas como design patterns que mostram objetos abstratos e concretos e interações.
Desenvolvimento baseado em componentes	Sistemas desenvolvidos pela integração de componentes (conjunto de objetos) que estão em conformidade com padrões de modelos e componentes.
Frameworks de aplicação	Conjunto de classes abstratas e concretas podem ser adaptados e ampliados para criar sistemas de aplicações.
Empacotamento de sistemas legados	Sistemas legados que podem ser ‘empacotados’ pela definição de um conjunto de interfaces e fornecimento de acesso a esses sistemas herdados por meio das interfaces.
Sistemas orientados a serviços	Sistemas desenvolvidos pela ligação de serviços compartilhados, que podem ser providos externamente.
Linhas de produto de aplicação	Um tipo de aplicação generalizada com base em uma arquitetura comum de tal maneira que possa ser adaptada para clientes diferentes.
Integração de COTS	Sistemas desenvolvidos pela integração de sistemas de aplicações existentes.
Aplicações verticais configuráveis	Um sistema genérico projetado de tal maneira que pode ser configurado para as necessidades de clientes de sistemas específicos.
Bibliotecas de programa	Bibliotecas de classes e funções que implementam abstrações comumente usadas disponíveis para reuso.
Geradores de programa	Um sistema gerador que incorpora conhecimento de um determinado tipo de aplicação e pode gerar sistemas ou fragmentos de sistema no domínio.
Desenvolvimento de software orientado a aspectos	Componentes compartilhados integrados em uma aplicação em diferentes lugares quando o programa é compilado.

O reuso de software passou por várias fases:

- **reuso individual e granular**, que possui benefício pequeno e é executado quando pequenas partes de um sistema são reaproveitadas em outro;
- **reuso corporativo e granular**, existe em geral uma biblioteca com partes dos sistemas que podem ser reutilizados e o benefício ainda é pequeno;
- **reuso baseado em frameworks**, normalmente se limita a infraestruturas e possui benefício médio; e,
- **desenvolvimento orientado a reuso**, onde o reuso é sistematizado e o benefício é grande, porém o investimento inicial também pode ser.

Para (KRUEGER, 1992), “reuso sistemático de software é uma técnica que é empregada para atender a necessidade de melhorar a qualidade e a eficiência do desenvolvimento de software, sem confiar na iniciativa individual ou na sorte.”

Linha de Produto de Software (LPS) é uma abordagem para reuso sistemático de software caracterizada como um conjunto de sistemas que compartilham características comuns de um mesmo segmento, como por exemplo, softwares para celulares, impressoras, bibliotecas etc. Neste contexto, é desenvolvida uma arquitetura padrão e um conjunto de componentes comuns reutilizáveis por todos os sistemas que são membros da mesma família. Ao criar um novo membro da família, a arquitetura e os componentes comuns são reutilizados e são inseridos, se necessário, os novos componentes requeridos ao novo membro.

Para (CLEMENTS, 2002):

> Uma linha de produto de software é um conjunto de sistemas que usam software intensivamente, compartilhando um conjunto de características comuns e gerenciadas, que satisfazem as necessidades de um segmento particular de mercado ou missão, e que são desenvolvidos a partir de um conjunto comum de ativos principais e de uma forma preestabelecida. (tradução nossa)

O *Software Engineering Institute* (SEI) descreve um *framework* para as linhas de produto, que é composto por três atividades essenciais, apresentadas na Figura 2-1.

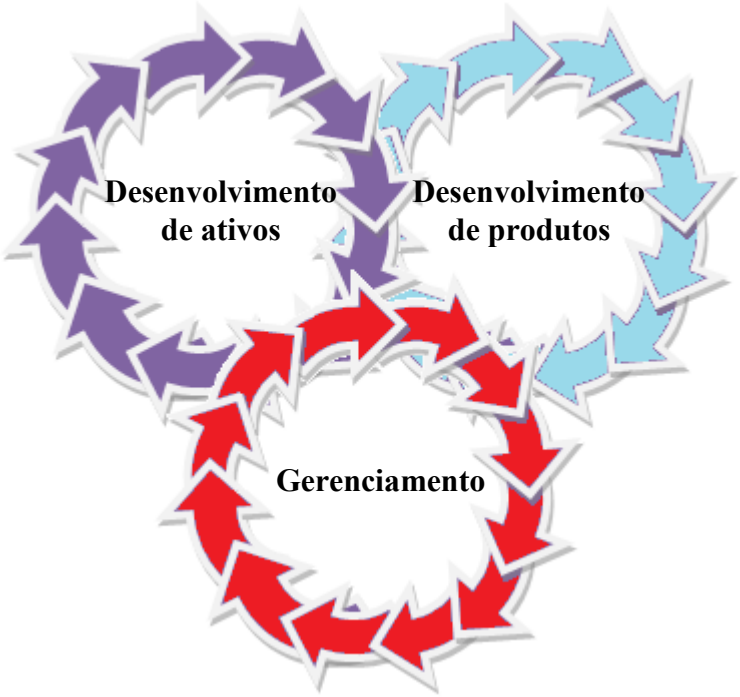

Figura 2-1. Atividades essenciais de Linhas de Produto de Software, adaptado de (CLEMENTS, 2002)

Na atividade de Desenvolvimento de Ativos é desenvolvida a base comum de ativos, ou seja, a arquitetura que será usada pela família. Ela pode, ou não, ser construída a partir de sistemas existentes. Também é chamada de Engenharia de Domínio. Na atividade Desenvolvimento de Produtos ou Engenharia de Aplicação, os produtos individuais são construídos e na atividade de Gerenciamento, ocorrem as atividades relacionadas aos aspectos gerenciais da abordagem.

Embora os termos Linhas de Produto de Software e Engenharia de Domínio sejam muitas vezes utilizados como sinônimos, existe uma diferença fundamental. A abordagem de Linhas de Produto de Software pressupõe a existência de dois ciclos distintos: um para tratar as questões da linha de produto (família) e outro para tratar as questões da produção do produto (indivíduo da família), enquanto a Engenharia de Domínio não trata as questões da produção do produto.

A Figura 2-2 apresenta a separação existente entre o desenvolvimento para o reuso, na engenharia de domínio, e o desenvolvimento com reuso, na engenharia de aplicação. Esta abordagem é denominada de Engenharia de Famílias de Produtos

(do original, em inglês, *Product Family Engineering*)[1]. Conforme (LINDEN et al., 2007), as atividades de cada ciclo se resumem em:

- engenharia de domínio:
  - gerenciamento de produtos: define os produtos que estarão na linha; identifica comunalidades e variabilidades entre eles e os analisa economicamente.
  - engenharia de Requisitos de Domínio: analisa os requisitos para os vários produtos da linha; constrói o modelo inicial de variabilidade.
  - design de Domínio: desenvolve a arquitetura da linha.
  - realização de Domínio: detalha o *design* e implementa os componentes de software reutilizáveis.
  - teste de domínio: testa os componentes, gera ativos de teste que podem ser reutilizados nos testes de aplicação.
- engenharia de aplicação:
  - engenharia de requisitos de aplicação: identifica os requisitos do produto, avaliando as comunalidades e variabilidades existentes na infraestrutura, procurando manter-se o mais próximo possível desta.
  - *design* de aplicação: deriva uma instância da arquitetura, de acordo com os requisitos identificados, construindo então adaptações específicas para o produto.
  - realização de aplicação: desenvolve a implementação final do produto, inclui reuso e configuração de componentes existentes ou constrói novos de acordo com os requisitos do produto.
  - teste de aplicação: testa o produto, validando-o de acordo com seus requisitos.

[1] O termo Linhas de Produto de Software (LPS) é mais utilizado nos USA, enquanto que o termo Engenharia de Famílias de Produto (EFP) é mais utilizado na Europa.

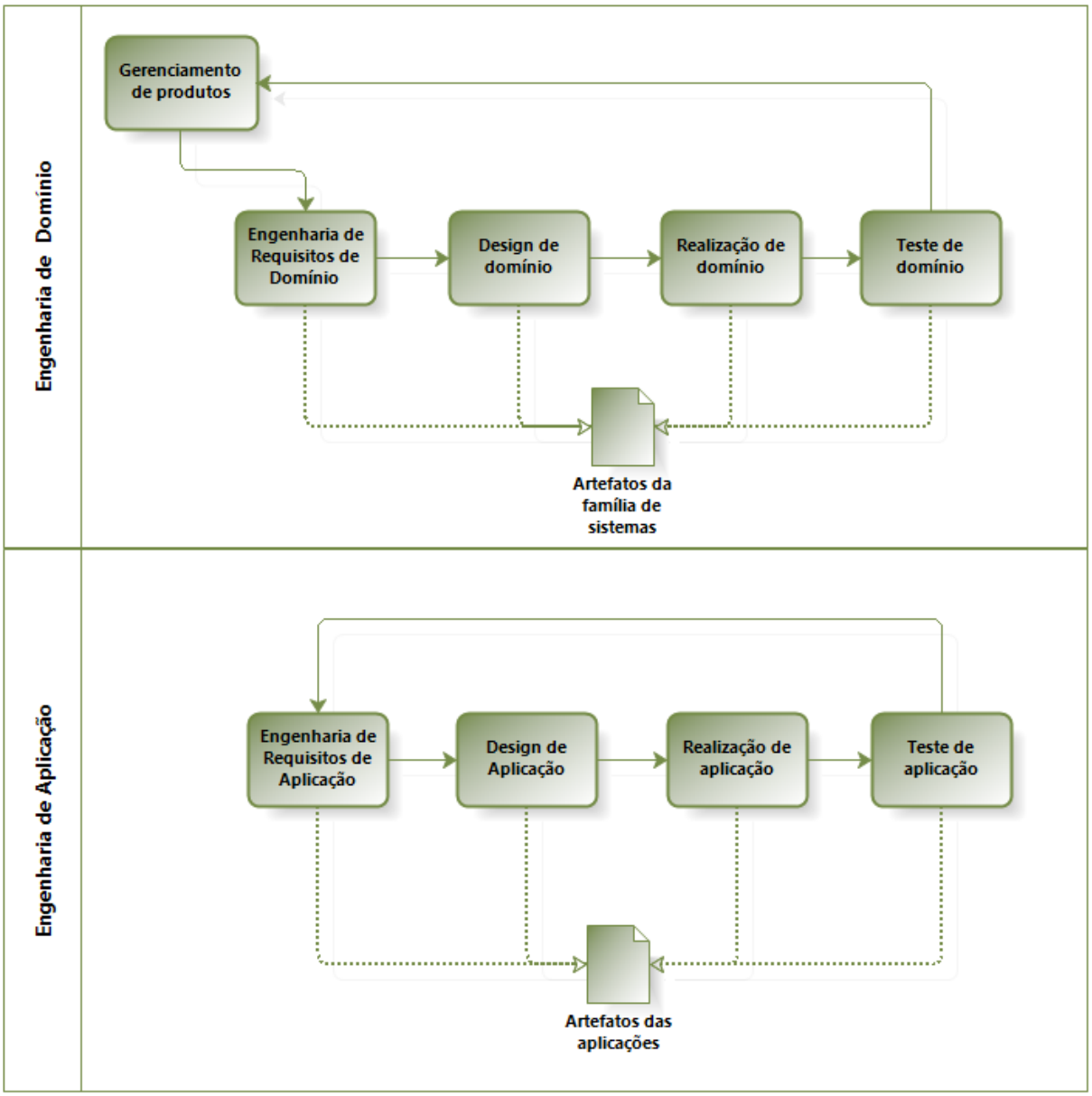

Figura 2-2. Os dois ciclos de vida da Engenharia de Famílias de Produtos (PFE), adaptado de (LINDEN et al., 2007).

Para (LINDEN et al., 2007):

> Ao contrário de muitas abordagens tradicionais de reuso que focam em ativos de código, a infraestrutura de linhas de produto engloba todos os ativos que são relevantes por todo o ciclo de vida de desenvolvimento de software. Os vários ativos cobrem todo o conjunto desde a fase dos requisitos até arquitetura e implementação e testes. Este conjunto de ativos define a infraestrutura da linha de produto. (p. 6, tradução nossa)

Algumas das vantagens das linhas de produto são citadas em (SEI, 2005a): maior produtividade em até 10%, maior qualidade em até 10%, diminuição de custos em até 60%, diminuição da necessidade de mão-de-obra em até 87%, diminuição do

time to market em até 98%, além da capacidade de entrada em novos mercados em meses, e não anos.

Embora as vantagens de utilizar a abordagem de reuso sistemático com linhas de produto de software pareça atraente à primeira vista, sua implantação não é trivial, uma vez que implica em diversas mudanças de cunho organizacional e tecnológico. As abordagens para que uma empresa inicie o uso das linhas de produto de software podem ser classificadas em pró-ativas, reativas e extrativas (ALVES et al., 2010) :

- na abordagem do tipo pró-ativa, a base comum de ativos (*assets)* reutilizáveis é desenvolvida primeiro e os produtos são desenvolvidos utilizando os ativos.
- na abordagem reativa, a linha é iniciada com um ou alguns poucos produtos, sendo incrementada na medida em que novos produtos são necessários.
- a abordagem extrativa se utiliza de um ou mais produtos de software existentes para criar a base de ativos comuns.

Dentre os ativos comuns que são criados para uma linha de produtos de software, existem os ativos que são utilizados por todos os produtos da linha, e ativos que são variáveis, ou seja, utilizados apenas em alguns produtos da linha. São as chamadas comunalidades e variabilidades, tratadas na próxima seção.

2.1.1 Comunalidades e Variabilidades

Uma linha de produto precisa definir suas características comuns a todos os sistemas da família, e suas características variáveis, que serão utilizadas de acordo com a necessidade. Segundo (BOSCH, 2005):

> Os artefatos chave em famílias de produto de software são o desenvolvimento, evolução e uso da arquitetura de família de produtos e um conjunto de componentes compartilhados. Ser capaz de desenvolver um artefato uma vez e usá-lo em vários produtos ou sistemas é, obviamente, um dos principais benefícios a ser alcançado. (p. 1, tradução nossa)

Para (LINDEN et al., 2007), a variabilidade pode ser separada em três tipos principais, conforme ilustra a Figura 2-3:

- comunalidades: se refere ao que é comum para todos os produtos da família;

- variabilidades: se refere apenas ao que é comum a alguns produtos da família;
- específico de produto: que normalmente não é necessário para o negócio com um todo, mas sim para um membro específico da família (esta especialidade pode não ser integrada no conjunto de ativos da família).

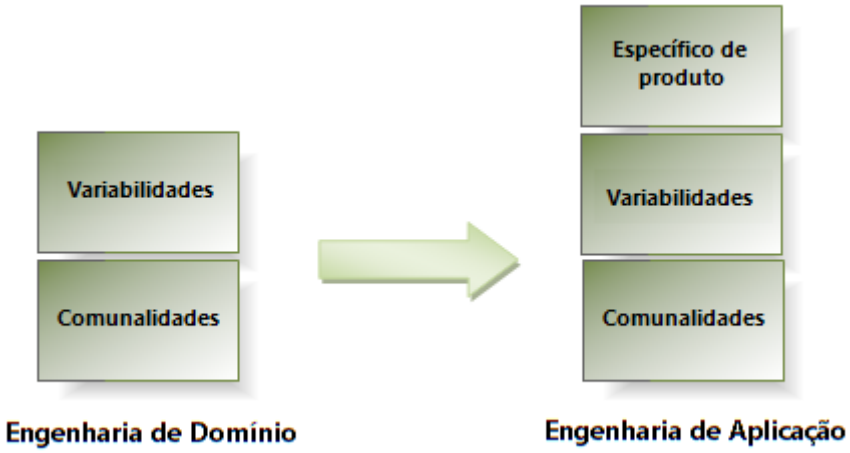

Figura 2-3. O relacionamento entre diferentes tipos de variabilidade, adaptado de (LINDEN et al., 2007)

A Figura 2-3 apresenta o relacionamento entre os diferentes tipos de variabilidade, na engenharia de domínio e na engenharia de aplicação. As variabilidades e comunalidades normalmente são tratadas durante a engenharia de domínio, enquanto as características específicas de produto normalmente são tratadas durante a engenharia de aplicação.

Ao incluir um novo produto na família, seus requisitos são avaliados de acordo com o que está disponível na base de ativos comuns. Se a base não contempla o requisito necessário ao novo produto, deve ser avaliada a sua real necessidade, para então ser decidido se a nova característica é incluída na base da família, ou se será específica para o novo produto. Pode acontecer de se optar pelo não desenvolvimento de produtos que não sejam aderentes ao escopo definido na linha. Esta avaliação deve levar em conta o escopo da linha, que é identificado na atividade de definição de escopo, que será tratada na próxima seção.

2.1.2 Definição de Escopo

O processo de definição de escopo para LPS é geralmente o primeiro passo a ser realizado ao iniciar a construção da linha e tem como objetivo mapear o escopo da LPS, identificando e delimitando os produtos, funcionalidades e áreas do domínio que devem fazer parte da LPS, assim como suas funcionalidades comuns e variáveis. Segundo (JOHN, 2009) “Definição de escopo é o processo de decidir em quais partes dos produtos de uma organização o reuso sistemático é economicamente útil e deve então ser suportado por uma infraestrutura de linhas de produto”.

Para (LINDEN et al., 2007), definição de escopo é um fator chave para o sucesso no desenvolvimento de linhas de produto, pois é preciso integração entre o planejamento da parte técnica da linha e da parte orientada ao mercado. A infraestrutura da linha de produtos precisa estar alinhada com os produtos que a organização irá desenvolver.

O processo de definição de escopo é citado como uma das práticas de gerenciamento técnico no framework geral de práticas de linha de produto do SEI (SEI, 2005). Os autores afirmam que o processo de definição de escopo também é realizado no desenvolvimento de sistemas convencionais, porém de maneira informal, normalmente imediatamente anterior à atividade de engenharia de requisitos.

O processo de definição de escopo é considerado a primeira saída da atividade de desenvolvimento de ativos (*Core Asset Development)* (SEI, 2005), e pode ocorrer em vários contextos, não somente ao criar uma linha de produtos nova. Na atividade de desenvolvimento de produtos, a definição de escopo é utilizada para decidir se um produto seria um membro viável para a linha. O produto pode ser um novo produto ou um produto proveniente de um sistema legado. Em resumo, a definição do escopo responde a questão: “Que produtos devem estar em minha linha de produtos de software?”.

Segundo (SCHMID, 2000), o processo de definição de escopo pode ser separado em três categorias: *product portfolio definition* (definição do portfólio de produtos), que identifica os produtos que devem ser contemplados e suas funcionalidades macro; *domain-centric scoping* (definição de escopo centrada no domínio) que identifica os domínios que são relevantes para estar em uma linha de produtos, a partir dos produtos identificados na categoria anterior; e, *asset-centric*

scoping (definição do escopo centrada nos ativos) que identifica os ativos reusáveis que serão desenvolvidos e que requisitos eles suportarão, para atender ao domínio identificado na categoria anterior.

Em (LINDEN et al., 2007) os autores classificam estas três categorias principais como sendo *product portfolio planning* (planejamento do portfolio de produtos)*, domain potential analysis* (análise do potencial do domínio) *e asset scoping* (definição do escopo dos ativos). Para a primeira categoria, o objetivo principal é capturar os produtos que devem fazer parte da linha e identificar seus principais requisitos (normalmente realizado sob o ponto de vista do mercado); para a segunda, o objetivo é avaliar por meio de uma análise sistemática, onde os investimentos de reuso devem se focar; para a terceira, o objetivo é definir os componentes que serão construídos para reuso.

O processo de definição do escopo não é realizado apenas no início da linha de produto. Ele também pode, e deve ser realizado durante sua evolução:

> Algumas vezes o processo de definição de escopo é mal entendido como uma ação que precisa ser executada apenas quando está sendo configurada uma nova linha de produtos. (...). Gerenciar o escopo é uma parte chave no gerenciamento de mudanças na situação de uma linha de produtos. Somente quando o escopo é constantemente gerenciado, pode ser assegurado que a organização reage adequadamente a novos desenvolvimentos e oportunidades. (LINDEN et al., 2007) p. 298, tradução nossa)

Um processo genérico de definição de escopo é apresentado em (JOHN, 2009), conforme ilustra a Figura 2-4.

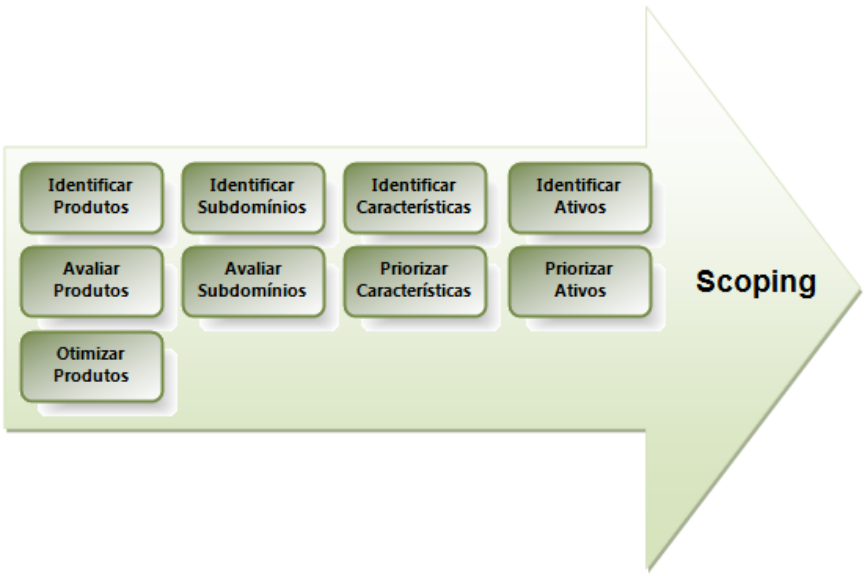

Figura 2-4. Um processo genérico de definição de escopo, adaptado de (JOHN, 2009).

No processo apresentado por (JOHN, 2009) os engenheiros da linha de produto e os especialistas no domínio interagem e selecionam as atividades a serem realizadas. As atividades são divididas em atividades de identificação, que listam e descrevem elementos; as atividades de avaliação, que julgam se os elementos estão apropriados, baseado em fatores padronizados; e, as atividades de otimização, que procuram melhorar a configuração da linha.

De acordo com (JOHN, 2009), definição de escopo é um processo contínuo que monitora o desenvolvimento em todos os mercados relevantes, domínios técnicos ou aplicações e solicitações de clientes. Ele pode então ser realizado na engenharia de linhas de produto e na evolução das linhas de produto.

A realização do processo de definição de escopo pode auxiliar as empresas a visualizar oportunidades que muitas vezes demorariam mais para serem percebidas. Um exemplo é a identificação da possibilidade de inclusão de produtos na linha sem um custo exagerado a mais, expandindo o mercado atendido.

Exemplo disso é o caso do “Market Maker Software AG” descrito em (LINDEN et al., 2007), onde a realização da definição de escopo auxiliou na identificação de que o layout e o *design* do software eram fatores chave para seus clientes. Assim, foi planejada uma arquitetura que possibilitasse a rápida definição de layout. O estimado é que, por conta disso, o tempo e esforço inicial gastos no projeto diminuíram em mais de 50%.

2.2 Abordagens para Definição de Escopo em Linhas de Produto de Software

O framework do SEI (SEI, 2005) cita algumas práticas específicas que podem ser utilizadas para a realização do *scoping*:

- aplicar o padrão *What to build* (CLEMENTS, 2002): o padrão situa esta atividade com outras atividades da área prática como “*Understanding Relevant Domains*” (Compreender Domínios Relevantes), “*Building a Business Case*” (Construir um Caso de Negócio), “*Market Analysis*” (Análise de Mercado) e “Technology Forecasting” (Previsão Tecnológica) que auxiliam na definição e evolução do escopo;
- examinar produtos existentes:
    - identificar produtos existentes similares aos que farão parte da linha;

    - buscar documentos disponíveis e conduzir demonstrações dos produtos;
    - conduzir pesquisas orais e escritas com especialistas dos produtos e desenvolvedores dos produtos atuais, usuários e pessoal que realiza manutenção;
    - identificar as capacidades, estrutura e evolução dos produtos, assim como quaisquer outros fatos relevantes sobre eles;
    - determinar quais desses produtos devem ser considerados parte da linha.
- conduzir um *workshop* para entender os objetivos da linha e dos produtos: reunir *stakeholders* e identificar:
    - os objetivos de negócio que devem ser satisfeitos pela linha;
    - o mapeamento dos objetivos de negócio da linha com os objetivos de negócio da organização e com as necessidades dos usuários;
    - descrição dos produtos atuais e potenciais que constituirão a linha;
    - restrições dos produtos e da produção, que pode incluir plataformas, padrões, protocolos e processos.
- construir um diagrama de contexto: o diagrama coloca a linha de produtos no contexto dos seus usuários e outros sistemas relacionados, e retrata as entidades importantes que afetam ou são afetadas pela linha. Deve também descrever possíveis opções e variações, assim como a forma de selecioná-las.
- desenvolver uma matriz de atributos / produtos: a matriz classifica, em ordem de prioridade, os atributos importantes pelos quais os produtos da linha se diferem. Define a variabilidade da linha.
- desenvolver cenários para linhas de produto: descrevem as interações dos usuários com os produtos da linha, auxiliando a identificar as interações comuns a vários produtos. Além de construir os *scenarios* é importante realizar *walkthroughs* para identificar se existem entidades, domínios ou produtos que não foram mapeados.

- utilizar PuLSE-Eco[2] (JOHN et al., 2006): primeiro os produtos candidatos são mapeados, incluindo sistemas existentes, planejados e potenciais, gerando uma lista de características potenciais para produtos da linha. Esta lista é mapeada como uma matriz, e são criadas funções de avaliação utilizando os objetivos dos *stakeholders* e de negócio. Os produtos são então caracterizados utilizando a matriz e as funções de avaliação.

Estas práticas para definição do escopo podem ser combinadas entre si, aumentando a probabilidade de sucesso da atividade. Existem diversas abordagens na literatura para realização da atividade de definição de escopo, e para identificar as abordagens mais recentes foi realizada uma revisão sistemática, apresentada na próxima seção.

2.2.1 Revisão Sistemática

A revisão sistemática foi realizada em Maio de 2012, com o objetivo de identificar as abordagens mais recentes de definição de escopo. A pesquisa foi realizada nas bases científicas: Scirus, IEEE Xplore, ACM (*Association for Computing Machinery*), Scopus e ScienceDirect. As buscas foram realizadas utilizando as palavras-chave "*product line scoping*" e "*product line planning*". Foram pesquisados trabalhos a partir do ano de 2006. A Tabela 2-2 apresenta o número de artigos encontrados nas respectivas bases científicas.

Tabela 2-2. Resumo do resultado da pesquisa

	Scirus	**IEEE Xplore**	**ACM**	**Scopus**	**Science Direct**
"product line scoping"	42	43	52	12	7
"product line planning"	28	20	4	9	6
TOTAL	70	63	56	21	13

A pesquisa retornou 223 documentos, sendo excluídos os documentos com títulos duplicados e aqueles que não eram artigos científicos, resultando em 151 artigos científicos. Os resumos destes artigos foram lidos, sendo excluídos os artigos

[2] PuLSE-Eco é um método específico para determinar o escopo de uma LPS. Faz parte de um *framework* de engenharia de Linhas de Produto chamado PuLSE (*Product Line Software Engineering*) (DeBAUD, 1999), desenvolvido e mantido pelo Fraunhofer Institute.

que não tinham como foco a atividade de definição de escopo. Em caso de dúvidas na análise do resumo, o artigo completo foi analisado.

Ao final desta análise restaram 45 artigos, dos quais oito não estavam disponíveis na íntegra, sendo possível realizar a leitura completa e análise de 37 artigos. Após a leitura completa, apenas 14 artigos apresentavam abordagens para definição de escopo, melhorias ou apoio nesse processo, os quais estão descritos a seguir.

O Quadro 2-3 apresenta um resumo das principais características das abordagens analisadas. A coluna “Tipo” identifica se o trabalho apresentou em sua maioria processamentos manuais (ou enfatiza a necessidade da presença de *experts* no domínio), semiautomático ou automático.

Quadro 2-3. Resumo dos Resultados da Pesquisa (Fonte: o Autor)

Autoria	Tipo	Objetivo
(GANESAN et al., 2006)	Manual	Identificar experts no domínio dos produtos existentes
(NOOR et al., 2007)	Manual	Migrar sistemas existentes para LPS de forma colaborativa
(NOOR et al., 2008)	Manual	Incorporar princípios ágeis durante o planejamento da LPS
(CARBON et al., 2008)	Manual	Estabelecer feedback da engenharia de aplicação para a engenharia da família, auxiliando evolução da LPS
(JOHN, 2009)	N/A	Comparar 16 abordagens relevante identificadas
(LIU, 2010)	Manual	Identificar comunalidades entre diferentes domínios
(ULLAH et al., 2010)	Semiautomática	Gerar portfólios de produtos para sistemas existentes baseados na preferência de segmentos de clientes
(JOHN, 2010)	Manual	Definir escopo por meio da análise de documentação de usuário

(LEE et al., 2000)	N/A	Comparar e analisar abordagens identificando seus componentes essenciais
(MULLER, 2011)	Semiautomática	Auxiliar a identificação de quais são as características mais importantes a serem desenvolvidas
(DUSZYNSKI, 2011)	Semiautomática	Identificar funcionalidades a partir da análise do código fonte
(ZIADI et al., 2012)	Semiautomática	Identificar funcionalidades a partir da análise do código fonte
(YOSHIMURA et al., 2008)	Semiautomática	Identificar variabilidades a partir de histórico de versão dos produtos existentes
(ARCHER et al., 2012)	Semiautomática	Auxiliar na transição de descrição de produtos para modelo de funcionalidades

Em (GANESAN et al., 2006) é proposta uma abordagem para identificar as pessoas que conhecem os produtos atuais da organização, por meio da análise de log de alterações em código fonte. As atividades descritas neste trabalho são definidas como recuperação de *ownership architecture*, e foram incorporadas como um dos componentes da abordagem PuLSE (*Product Line Software Engineering)*. O principal benefício citado para a fase de definição de escopo é a identificação das pessoas que conhecem os produtos atuais da empresa e suas implementações, possibilitando sua participação nos *workshops* e entrevistas realizados nessa fase.

Uma abordagem colaborativa para realizar a definição de escopo é proposta em (NOOR et al., 2007). O foco do trabalho é a migração de sistemas existentes para uma abordagem de LPS. Para a engenharia de colaboração foi utilizado o padrão *ThinkLets*, e para a definição de escopo foi o PulSE-Eco (*Product Line Software Engineering – Economic Scoping*). Também foi utilizada a técnica *EasyWinWin* na negociação de requisitos. Nesta abordagem, é necessária a presença dos *stakeholders* para a realização da fase *Product Line Mapping* do Pulse, que é realizado com um ou mais *workshops* ou entrevistas com os especialistas no domínio e no escopo.

Uma abordagem colaborativa que aplica princípios das metodologias ágeis durante o planejamento da LPS é proposta em (NOOR et al., 2008). Nesta abordagem a necessidade da participação dos *stakeholders* em todas as atividades do processo é enfatizada.

Em (CARBON et al., 2008) a prática ágil *planning game* foi adaptada para ser usada com LPS. Os resultados mostraram que a técnica é eficiente para estabelecer um ciclo de *feedback* da engenharia de aplicação para a engenharia da família. A técnica se mostrou complementar ao processo contínuo de definição de escopo. A definição de escopo, neste trabalho, foca principalmente em alterações ou em novos requisitos de clientes externos. O *planning game* em LPS endereça os problemas dos engenheiros de aplicação como clientes internos da engenharia de família, reutilizando os componentes da linha de produtos.

Em (JOHN, 2009), foram identificadas na literatura várias abordagens para definição de escopo e foram consideradas como relevantes as abordagens que apresentavam uma relação clara com linhas de produto, alguma maturidade e documentação suficiente para que fossem entendidas e aplicadas. O intervalo de datas de publicação utilizado para a pesquisa não foi informado, porém as referências apresentam trabalhos entre o ano 2000 e 2008. De todas as abordagens encontradas, 16 abordagens foram consideradas relevantes. Estas abordagens foram classificadas de acordo com alguns fatores como: objetivo, entradas e saídas, variabilidade, maturidade, se existem publicações recentes sobre a abordagem, dentre outros. Algumas conclusões citadas em (JOHN, 2009) são listadas a seguir:

- as abordagens de definição de escopo podem ser utilizadas para vários objetivos, em basicamente três cenários distintos: definição de escopo na engenharia de LPS, durante a engenharia de requisitos ou para a evolução da LPS;
- os principais artefatos de entrada para a atividade de definição de escopo são descrições dos produtos, do domínio e requisitos;
- a forma de apresentação do resultado da definição de escopo é diversificada, e ainda não está claro como utilizá-la nas fases seguintes;
- não foi identificado um domínio específico de sistemas onde esta atividade fosse mais ou menos apropriada;
- a atividade de definição de escopo envolve todos os *stakeholders* de uma organização.

Identificar comunalidades entre diferentes domínios é o foco do trabalho de (LIU, 2010), porém a identificação também é dependente de pessoas. A abordagem utiliza os resultados da análise de domínio e outros modelos para fazer com que as comunalidades sejam visíveis para os desenvolvedores, por meio da construção de diagramas. Neste trabalho é afirmado que o problema mais difícil é encontrar comunalidades essenciais entre sistemas e guardá-las de maneira sistemática.

A abordagem proposta em (ULLAH et al., 2010), usa preferências dos clientes sobre as características do produtos para gerar múltiplos portfólios de produtos, cada um contendo uma variação de produto por segmento de cliente. O método é para sistemas que já existem e estão evoluindo para uma estrutura de LPS, e também leva em consideração o impacto da evolução na estrutura atual do sistema, analisando sua estrutura para sugerir a variação dos produtos.

A abordagem CAVE (*commonality and variability extraction*) proposta por (JOHN, 2010) é uma abordagem manual que utiliza documentação de usuário como fonte para o processo de definição de escopo. Um consultor de linhas de produto seleciona os documentos do sistema que serão utilizados, avalia suas semelhanças e em seguida aplica os padrões definidos para buscar os elementos selecionados (funcionalidade, domínio, elemento opcional, etc.). Após a aplicação dos padrões é possível gerar uma matriz com os elementos encontrados relacionados com os produtos nos quais foram identificados. Essa matriz então é avaliada por um especialista de domínio. Exemplos dos padrões utilizados são apresentados no Quadro 2-4.

Quadro 2-4. Exemplo de padrões, adaptado de (JOHN, 2010)

Input	Pattern short description	Output
Cabeçalho	Cabeçalhos de seções ou sub-seções tipicamente contêm características	Características
Palavra/frase	Palavras ou frases que são repetidas em diferentes partes da documentação podem ser domínios ou sub-domínios.	Domínio
Todos os elementos	Elementos que ocorrem apenas em um manual de usuário são elementos opcionais.	Elemento opcional

O trabalho apresentado em (LEE et al., 2000) compara e analisa as abordagens tradicionais para a realização da atividade de definição de escopo para encontrar seus componentes essenciais e desenvolvê-los em uma abordagem única. Os autores afirmam que não existe uma única abordagem que abrange as três

categorias existentes, e que não é fácil identificar os pontos comuns, diferentes, fortes e fracos de cada uma delas.

Uma abordagem adicional às abordagens de definição de escopo é apresentada em (MULLER, 2011), com o objetivo de auxiliar organizações a identificar quais funcionalidades são mais importantes tendo como objetivo aumentar lucro ou diminuir despesas. A abordagem identifica entidades relevantes que influenciam na rentabilidade da linha e seus relacionamentos, em seguida formula o problema de otimização matematicamente.

Dois artigos encontrados buscam identificar funcionalidades utilizando o código fonte. Em (DUSZYNSKI, 2011) é proposta uma abordagem de engenharia reversa para extrair informações de variabilidade do código fonte de produtos de software similares. O foco do trabalho é auxiliar organizações que falharam na tentativa de construir uma linha de produtos de software de maneira pró-ativa e acabaram construindo um único produto inicial e depois vários clones deste para atender às novas demandas. Esta abordagem assume que os códigos que serão analisados são clones um dos outros, e, portanto possuem uma estrutura de código muito similar.

Já em (ZIADI et al., 2012) a abordagem proposta utiliza engenharia reversa para criar um diagrama de classes simplificado do código fonte dos produtos e a partir disso decompõe o diagrama em partes menores, chamadas "primitivas de construção", utilizadas para comparar os produtos na segunda etapa e identificar funcionalidades candidatas. Se o nome das classes e métodos que fazem a mesma coisa tiverem nomes diferentes entre os produtos, eles precisam ser uniformizados para que a abordagem possa ser utilizada.

Em (YOSHIMURA et al., 2008) é apresentada uma abordagem para analisar variabilidades candidatas a partir do histórico de versão de produtos, considerando que suas variabilidades constam no histórico de alterações. Já em (ARCHER et al., 2012) o objetivo é auxiliar na transição de descrição de produtos para modelo de funcionalidades (*feature model*) de forma semiautomática. As descrições dos produtos devem estar organizadas em tabelas, onde cada linha representa um produto.

Com relação aos artigos que buscavam auxiliar na identificação dos produtos e funcionalidades da linha, foi identificado que na maioria das abordagens

apresentadas existe a necessidade da presença de um especialista, *stakeholder* ou cliente:

- em (NOOR et al., 2007), é proposta uma abordagem colaborativa, que precisa da presença de especialistas de domínio;
- em (NOOR et al., 2008) a proposta colaborativa é aplicada em metodologias ágeis, e necessita da presença dos *stakeholders* que conheçam o domínio;
- uma das conclusões da pesquisa apresentada em (JOHN, 2009) é que após a análise dos papéis envolvidos e requeridos durante a aplicação das abordagens encontradas, se conclui que a atividade de definição de escopo é uma atividade que envolve vários *stakeholders* das organizações;
- em (LIU, 2010), as comunalidades entre diferentes domínios são identificadas para posterior análise pelos desenvolvedores;
- a abordagem proposta em (ULLAH et al., 2010), usa preferências dos clientes sobre as características do produtos;
- a abordagem CAVE, proposta por (JOHN, 2010) procura diminuir a necessidade da presença dos especialistas, analisando documentação de usuário manualmente por meio de padrões de busca pré-estabelecidos.
- em (DUSZYNSKI, 2011), (ZIADI et al., 2012) e (YOSHIMURA et al., 2008) as abordagens apoiam os especialistas na identificação das funcionalidades porém são limitadas pelo contexto em que se aplicam, pois em (DUSZYNSKI, 2011) os códigos analisados precisam ser muito semelhantes (clones), em (ZIADI et al., 2012) os elementos como classes e métodos precisam ter os mesmos nomes e em (YOSHIMURA et al., 2008) o histórico de alterações precisa existir e ter informações relevantes.

A abordagem CAVE diminui o tempo necessário da presença de um especialista de domínio, porém possui atividades manuais, que demandam tempo e que estão sujeitas a vários erros por serem executadas por humanos. Um dos trabalhos futuros citados em (JOHN, 2010) é a automação dos padrões para prover análise automática de documentos e identificação de artefatos para a LPS, pois a tarefa de aplicar manualmente padrões em documentos grandes é uma tarefa tediosa.

Para automatizar a análise de documentos e identificação de artefatos para a LPS, pode-se utilizar técnicas de Recuperação de Informação, pois seu objetivo é encontrar material de natureza não estruturada (usualmente texto) que satisfaz uma necessidade de informação de dentro de grandes coleções.

Se for possível automatizar a análise de documentos e identificação de artefatos para a LPS por meio de técnicas de Recuperação de Informação, pode-se minimizar a necessidade da presença constante dos especialistas de domínio durante a fase de definição de escopo. Portanto a próxima seção tratará o conceito de Recuperação de Informação.

2.3 Recuperação de Informação

De acordo com (BAEZA-YATES; RIBEIRO-NETO, 1999), “Recuperação de Informação” (RI) lida com a representação, armazenamento, organização e acesso à itens de informação. Os itens de informação estão relacionados à necessidade de informação do usuário, que devem ser representados por uma *query* para ser processada por um sistema de RI. Desta forma, o objetivo chave de um sistema de RI é recuperar informação que pode ser relevante para o usuário, e não apenas dados.

Para (MANNING et al., 2009) um sistema de RI tem como objetivo encontrar material (usualmente documentos) de natureza não estruturada (usualmente texto) que satisfaz uma necessidade de informação de dentro de grandes coleções (usualmente armazenadas em computadores).

Segundo (CALADO, 2004), um problema tratado por RI pode ser interpretado como um conjunto de três partes: o usuário (que tem a necessidade de informação, traduzida para o sistema como uma *query*), o sistema de RI e o repositório de dados digitais que compõe os documentos da coleção.

Sistemas de RI podem ser utilizados para recuperar informação representada por diversos tipos de documentos como, por exemplo: textos, imagens, sons, vídeos e páginas *web* (FERNEDA, 2003).Com relação aos sistemas de RI que trabalham com texto, (SINGHAL, 2001) afirma que os primeiros sistemas de RI eram sistemas *booleanos* que permitiam aos usuários especificar a informação necessária usando uma complexa combinação de ANDs, ORs e NOTs. Entretanto estes sistemas *booleanos* são menos efetivos que os sistemas de recuperação que trabalham com *ranking*.

Com relação ao processamento de documentos texto, (MANNING et al., 2009) apresenta alguns itens que devem ser considerados e tratados para que se possa extrair as informações:

- o esquema de codificação do documento, como por exemplo UTF-8;
- a representação binária do documento, como por exemplo arquivos no formato DOC do *Microsoft Word* ou arquivos em formatos comprimidos como arquivos ZIP.
- a escolha da "unidade de documento" que será tratada e utilizada na busca das informações. Por exemplo, ao analisar arquivos de uma pasta, cada arquivo pode ser considerado uma unidade, assim como ao analisar emails, pode-se desejar dividir o texto do email como uma unidade, e cada um dos seus anexos como unidades independentes.

Algumas operações em texto para reduzir a complexidade da representação do documento são citadas por (BAEZA-YATES; RIBEIRO-NETO, 1999):

- eliminação de *stop words* (como artigos e conectivos);
- *stemming* (que reduz palavras para sua raiz gramatical);
- identificação de grupos de substantivos (eliminando adjetivos, advérbios e verbos).

Outras operações que podem ser aplicadas aos documentos de texto são citadas por (MANNING et al., 2009), para determinar o vocabulário de termos:

- tokenização: é a atividade de separar determinado texto em pedaços, chamados *tokens*, às vezes removendo ao mesmo tempo alguns caracteres como os de pontuação. Ao ser utilizada uma *query* para busca em uma coleção, é importante aplicar nela o mesmo processo de tokenização aplicado nos documentos que serão pesquisados para que os resultados não sejam afetados.
- eliminação de *stop words*, que são palavras extremamente comuns e que não auxiliam na seleção de documentos relevantes para o usuário. A eliminação geralmente é feita por meio da criação de uma *stop list*, que pode ser gerada manualmente ou por meio da análise dos termos que aparecem com mais frequência na coleção. Entretanto, dependendo do domínio da informação, a eliminação de *stop words* pode não ser necessária e até prejudicar os resultados. Por isto, a tendência geral do uso desta lista em sistemas de RI pelo tempo tem passado pelo uso de

grandes listas com mais de 200 termos para pequenas listas ou nenhuma lista.

- normalização: é o processo de transformar *tokens* de forma que diferenças superficiais não prejudiquem o resultado da busca, como por exemplo quando o usuário busca por “janelas” e se deseja que o sistema leve em consideração também os termos “Janela” e “janela”. Isto pode ser realizado de diversas maneiras, como por exemplo criando classes de equivalência ou reduzindo todos os temos para caixa baixa (*lower case*).
- *stemming* e *lemmatization*: o objetivo dos dois é reduzir a forma de uma palavra para sua forma base. O *stemming* é um processo heurístico que normalmente retira o início ou o final das palavras, enquanto o *lemmatization* utiliza análises de vocabulário e morfológicas com o objetivo de encontrar a forma base de uma palavra, conhecida como *lemma*.

O objetivo principal de um sistema de RI é recuperar todos os documentos que são relevantes para uma *query* do usuário, recuperando também o menor número possível de documentos irrelevantes (BAEZA-YATES; RIBEIRO-NETO, 1999). As duas estatísticas chave que são normalmente utilizadas para avaliar a efetividade de um sistema de RI são definidas por (MANNING et al., 2009) como*:*

- *precision*: qual fração dos resultados retornados são relevantes com relação à informação necessária?
- *recall*: qual fração dos documentos relevantes na coleção foram retornados pelo sistema?

De acordo com (SINGHAL, 2001), um bom sistema de recuperação de informação deve recuperar tantos documentos relevantes quanto possível (ter um alto *recall*), e deve recuperar poucos documentos não relevantes (ter alto *precision*). Entretanto, as técnicas que tendem a melhorar o *recall* tendem a piorar o *precision* e vice-versa.

Segundo (BAEZA-YATES; RIBEIRO-NETO, 1999), para que o sistema de RI seja efetivo na recuperação de informações relevantes ao usuário, é necessário de alguma forma interpretar o conteúdo dos itens de informação em uma coleção e organizá-los de forma a gerar uma classificação (ranking) de acordo com a relevância da *query* do usuário. Esta interpretação envolve a extração de informações sintáticas e semânticas do conteúdo que está sendo processado.

A utilização de técnicas de Processamento de Linguagem Natural (PLN) em sistemas de RI pode enriquecer seus resultados, pois o PLN faz com que máquinas sejam capazes de ler, escrever e traduzir textos (GOTTSCHALG-DUQUE, 2005) e trabalham com informações sintáticas e semânticas. Segundo (FERNEDA, 2003), PLN é um conjunto de técnicas computacionais para a análise de textos com o objetivo de simular o processamento da línguagem pelos humanos.

O PLN é um ramo da linguística que estuda a geração e recepção automática de textos (GOTTSCHALG-DUQUE, 2005). Para (OTHERO, 2006), a área de PLN estuda a linguagem por meio da construção de *softwares*, aplicativos e sistemas computacionais como, por exemplo, tradutores, *parsers* e etc, capazes de interpretar ou gerar informações em linguagem natural.De acordo com (OTHERO, 2006), a Sintaxe e Semântica são duas áreas da linguística de grande importância para o desenvolvimento de programas de PLN, pois a sintaxe estuda a ordem dos constituintes de uma frase, enquanto a semântica estuda o significado das palavras e proposições.

Para (SILVA et. al., 2007), um sistema de PLN pode ser utilizado para fazer revisões ortográficas de textos, análises sintáticas ou traduzir frases ou textos, por exemplo.

Segundo (SILVA et. al., 2007), o material de entrada de um sistema de PLN é um texto, que deve ser analisado para que suas informações linguísticas possam ser manipuladas. As palavras de uma sentença podem ser caracterizadas de diversas maneiras, chamadas de estatuto:

- *fonético-fonológico*: identificação da identidade sonora dos elementos da palavra;
- *morfológico*: as unidades mínimas que possuem significado são compreendidas com relação à flexão e formação;
- *sintático*: a distribuição das palavras resulta nas funções que estas desempenham na sentença;
- *semântico*: o conteúdo significativo da palavra possui relação com a identificação de objetos no mundo.
- *pragmático-discursivo*: a força expressiva das palavras possui relação com a identificação de objetos no mundo de acordo com o contexto e produção discursiva (mundo extra-linguístico).

De acordo com (SILVA et. al., 2007), os módulos de processamento que podem fazer parte de um sistema de PLN são:

- analisador Léxico: identifica e separa os componentes significativos da sentença sob análise;
- analisador Sintático (ou *parser*): constrói uma estrutura sintática para a sentença sob análise;
- analisador Semântico: interpreta componentes da sentença ou a sentença como um todo, necessitando informações sobre o domínio para que a interpretação seja correta;
- analisador do Discurso: trata dos relacionamentos entre as sentenças, quando o significado de uma sentença pode depender das sentenças anteriores ou influenciar as posteriores;
- analisador Pragmático: interpreta a sentença considerando o contexto da ocorrência do discurso.

Em (GONZALEZ; LIMA, 2003) são citadas algumas estratégias de sistemas de PLN que envolvem conhecimento linguístico:

- etiquetagem de texto: identifica, por meio da colocação de uma etiqueta (*tag*), a categoria gramatical de cada item do texto analisado, indicando as funções sintáticas das palavras, com sujeito e objeto direto.
- normalização de variações linguisticas: pode ser morfológica (por meio do *stemming* e do *lemmatization*), sintática (quando existem frases semanticamente equivalentes porém sintaticamente diferentes) ou léxico-semântica (por meio do agrupamento de similaridades semânticas);
- eliminação de *stopwords*: eliminação de palavras funcionais como artigos, conectivos e preposições.

Parsing é o processo de interpretação automática ou semiautomática de sentenças de linguagem natural, classificando morfossintaticamente as palavras e atribuindo a elas sua estrutura sintagmática (OTHERO, 2006). Porém, construir um *parser* com as regras da gramática da língua portuguesa possui alcance limitado, pois não existe uma gramática definitiva e muitos critérios de gramaticalidade não são matematizáveis (SILVA et. al., 2007).

Etiquetagem de texto, etiquetagem morfossintática, ou anotação linguística, é o processo de atribuir uma etiqueta (tag), pertencente a um conjunto definido de etiquetas, a cada palavra ou pontuação existente no texto que está sendo analisado,

de acordo com o contexto em que aparecem. Por exemplo, na frase “A casa é verde.” o “a” receberia a etiqueta correspondente à artigo, a “casa” receberia etiqueta de substantivo, o “é” receberia etiqueta de verbo, o “verde” de adjetivo e o “.” de ponto (ÁLVARES, 2007).

O processo de *parsing* em conjunto com anotação linguística (etiquetagem de texto ou etiquetagem morfossintática) são as técnicas escolhidas para serem utilizadas neste trabalho, pois identificam as funções sintáticas das palavras de um texto, possibilitando a análise e busca automática de determinados padrões usados normalmente para descrever funcionalidades.

2.4 Considerações sobre o capítulo

Este capítulo apresentou conceitos relacionados aos temas de pesquisa deste trabalho. Na primeira seção foram apresentados conceitos sobre LPS e em seguida sobre o foco deste trabalho: definição de escopo, assim como uma revisão sobre as abordagens existentes. Logo após os conceitos de recuperação de informação, processamento de linguagem natural e anotação linguística, que também são relevantes para este trabalho, foram apresentados.

CAPÍTULO 3 - ESTRUTURAÇÃO DA PESQUISA

Este capítulo apresentará conceitos sobre metodologia e métodos de pesquisa e detalhará como esta pesquisa foi estruturada para atingir o objetivo geral do trabalho.

3.1 Conceitos relevantes sobre metodologia e métodos de pesquisa

Segundo (SANTOS, 1999), as pesquisas podem ser caracterizadas segundo objetivos, segundo procedimentos de coleta ou segundo as fontes utilizadas na coleta de dados. De acordo com os objetivos, a pesquisa pode ser caracterizada como exploratória, descritiva ou explicativa (SANTOS, 1999):

- **exploratória**: visa criar maior familiaridade em relação a um fato ou fenômeno. É quase sempre feita como levantamento bibliográfico, entrevistas com profissionais que estudam/atuam na área, entre outros.
- **descritiva**: descrever um fato ou fenômeno após sua exploração. É um levantamento das características conhecidas, normalmente feita na forma de levantamentos ou observações sistemáticas.
- **explicativa**: visa criar uma teoria aceitável a respeito de um fato ou fenômeno. Preocupa-se com a identificação dos fatores que contribuem ou determinam a ocorrência, ou a maneira de ocorrer de fatos ou fenômenos.

Com relação aos procedimentos de coleta, a pesquisa pode ser classificada como experimento, levantamento, estudo de caso, pesquisa bibliográfica, pesquisa documental, pesquisa-ação, pesquisa-participante, pesquisa ex-post-facto, pesquisa quantitativa, pesquisa qualitativa (SANTOS, 1999):

- **experimento**: quanto um fato ou fenômeno da realidade é reproduzido de forma controlada, para descobrir fatores que o produzem ou que são produzidos. São geralmente feitos por “amostragem”.
- **levantamento**: busca informação diretamente com um grupo de interesse a respeito dos dados que se deseja obter. Normalmente os dados coletados são tabulados e analisados quantitativamente.

- **estudo de caso**: aprofundar aspectos característicos em um objeto de pesquisa restrito. Exige grande capacidade de observação do pesquisador e parcimônia com relação à generalização de resultados.
- **pesquisa bibliográfica**: utilização total ou parcial de quaisquer fontes bibliográficas tais como materiais escritos/gravados, livros, revistas, *web sites* entre outros.
- **pesquisa documental**: se utiliza de documentos como tabelas estatísticas, relatórios de empresas, documentos que ainda não receberam organização, tratamento analítico e organização.
- **pesquisa-ação**: acontece quando qualquer dos processos é desenvolvido envolvendo pesquisadores e pesquisados no mesmo trabalho.
- **pesquisa-participante**: ocorre quando o pesquisador faz parte de um dos dados pesquisados.
- **pesquisa ex-post-facto**: examina um fato ou fenômeno que já está pronto, não é de controle do pesquisador.
- **pesquisa quantitativa**: é a pesquisa onde é importante a coleta e a análise quantificada dos dados e geração de resultados por meio destas análises. Gera resultados que se impõem como evidência imediata.
- **pesquisa qualitativa**: é a pesquisa cujos dados só tem sentido por meio de um tratamento lógico feito posteriormente pelo pesquisador. Depende do “olho clínico” do pesquisador.

Com relação às fontes de informação, a pesquisa pode ser caracterizada em pesquisa de campo, de laboratório, ou de bibliografia (SANTOS, 1999):

- **campo**: Recolhe os dados no local onde ocorrem os fatos ou fenômenos, conforme percebidos pelo pesquisador.
- **laboratório**: quando ocorre a interferência artificial na produção do fato/fenômeno ou a artificialização de sua leitura.
- **bibliografia**: os dados capturados via campo ou via laboratório são fonte para raciocínio e conclusões a respeitos dos fatos / fenômenos. É uma fonte de informações com dados já organizados.

3.2 Caracterização da pesquisa

De acordo com a seção anterior, esta pesquisa se classifica em:

- de acordo com os objetivos: exploratória.

- de acordo com os procedimentos de coleta: qualitativa.
- de acordo com as fontes de informação utilizadas: laboratório.

3.3 Estratégia de pesquisa

Neste tópico serão explicadas as etapas para o desenvolvimento da pesquisa e como cada etapa será realizada. A pesquisa está dividida em 6 fases, conforme ilustrado na Figura 3-1:

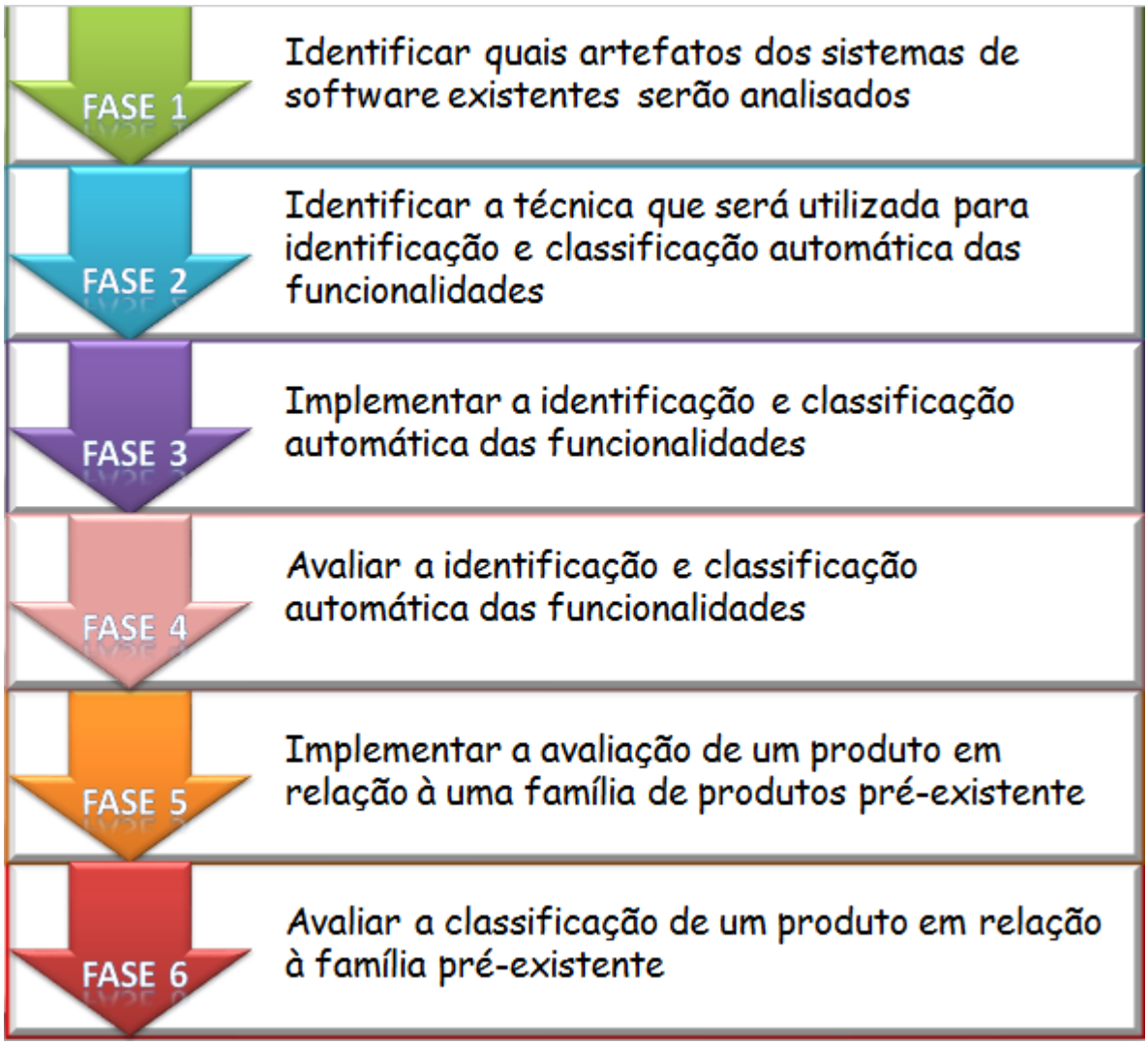

Figura 3-1. Fases da pesquisa (Fonte: o Autor).

3.3.1 Fase 1 - Identificar quais artefatos dos sistemas de software existentes serão analisados

Nesta fase foram selecionados e caracterizados os artefatos que foram utilizados nesta proposta para criação da LPS. A proposta deste trabalho consiste

em auxiliar na definição de escopo realizada ao se criar uma LPS de maneira extrativa, ou seja, a partir de sistemas existentes. Portanto, o artefato selecionado para ser utilizado nesta abordagem tem a necessidade de ser um artefato comumente encontrado nas organizações e que também possa ser analisado manualmente, pois é o processo de análise manual destes artefatos que se pretende melhorar.

3.3.2 Fase 2 – Identificar a técnica que será utilizada para identificação e classificação automática das funcionalidades

O foco desta fase foi o estudo para a identificação da técnica mais adequada que possibilitasse a identificação e classificação automáticas ou semiautomáticas. O objetivo era realizar testes para analisar qual forneceria os melhores resultados.

3.3.3 Fase 3 – Implementar a identificação e classificação automática das funcionalidades

Nesta fase foi realizada a implementação para identificar e classificar automaticamente as funcionalidades, utilizando os artefatos selecionados na fase 1 e a técnica selecionada na fase 2.

O método proposto foi desenvolvido para plataforma web utilizando a linguagem Java, versão 6, por meio do IDE (Integrated Development Environment) Eclipse. Os detalhes da implementação, com o fluxo de interação com o usuário são descritos no capítulo 4.

3.3.4 Fase 4 – Avaliar a identificação e classificação automática das funcionalidades

A avaliação da identificação e classificação automática das funcionalidades foi realizada em três etapas:

- primeiro experimento ou Pré-experimento: processamento manual e automático realizado pelo próprio pesquisador, com o objetivo de avaliar, de forma preliminar, a abordagem proposta.
- segundo experimento: Na segunda etapa, foi realizado um experimento semelhante, porém com 6 pessoas (sem a participação do pesquisador).

O objetivo era analisar o desempenho da abordagem proposta e a identificação de pontos de melhoria necessários.

- terceiro experimento: Na terceira etapa foi realizado um experimento com 2 pessoas, semelhante ao anterior, porém com as melhorias identificadas no segundo experimento já implementadas na abordagem proposta.

As pessoas selecionadas para participar dos experimentos deveriam ter, no mínimo, formação em nível de pós-graduação. O nível mais desejado, no entanto, era mestrando. Os resultados dos experimentos estão no capítulo 4.

3.3.5 Fase 5 – Implementar a avaliação de um produto em relação a uma família de produtos pré-existente

Nesta fase foi implementado o algoritmo que avalia se um novo produto faz parte ou não de uma LPS já criada. Quando a LPS é criada automaticamente, cada produto é analisado para identificar qual é a porcentagem de funcionalidades comuns (que existem em todos os produtos) e variáveis (que existem em pelo menos dois produtos) em relação ao total de funcionalidades identificadas naquele produto. A menor porcentagem encontrada é guardada como referência para este processo de avaliação de um produto novo.

Para validar se um novo produto faz parte ou não de uma LPS existente, leva-se em consideração que este novo produto precisa, pelo menos, ter o mesmo percentual de funcionalidades classificadas como comuns ou variáveis com relação à linha que o valor de referência identificado anteriormente.

A abordagem proposta indica, por meio dessa comparação, se o produto faz parte ou não da LPS existente. Os detalhes da implementação, com o fluxo de interação com o usuário são descritos no capítulo 4.

3.3.6 Fase 6 – Avaliar a classificação de um produto em relação à família pré-existente

Nesta fase foram realizados testes com o objetivo de avaliar a análise de um produto em relação à família pré-existente. Para isso, foi utilizada a LPS criada na Fase 4 e foram realizadas quatro avaliações: utilizando um manual de celular da família *Smartphones*, utilizando um manual de celular que não é da família *Smartphones,* um manual que não é de um celular e um manual de um celular

compatível com a família. Os resultados destas avaliações estão descritos em detalhes no Capítulo 4.

3.4 Considerações sobre o capítulo

Este capítulo apresentou os conceitos relacionados à classificação de uma pesquisa científica, assim como a caracterização desta pesquisa de acordo com os conceitos citados. As fases e etapas que foram realizadas durante esta pesquisa também foram citadas e detalhadas.

CAPÍTULO 4 - DESENVOLVIMENTO DA PESQUISA

Neste capítulo será descrito o desenvolvimento das fases desta pesquisa relacionadas à implementação e validação da proposta deste trabalho. Algumas fases também foram descritas em (IANZEN et al., 2012).

4.1 Fase 1 - Identificar quais artefatos dos sistemas de software existentes serão analisados

Os artefatos selecionados para serem utilizados nesta proposta foram documentos textuais que continham informações relacionadas às funcionalidades dos sistemas existentes. Cada documento textual deveria descrever as funcionalidades de um único produto. Para utilização durante a construção e avaliação da proposta, foi definida a utilização de manuais de usuário, pois estes normalmente descrevem os produtos de maneira orientada a características (JOHN et al., 2006). O formato mais comum de disponibilização de manuais de forma eletrônica é o formato PDF (*Portable Document Format*), portanto este deveria ser aceito na abordagem proposta.

O documento utilizado pode estar em formato PDF ou em formato TXT. Caso esteja em formato PDF, será feita a conversão para TXT. Contudo, esta conversão pode afetar a qualidade do documento, pois dependendo da formatação do PDF, o algoritmo utilizado que realiza a conversão pode não reconhecer alguns caracteres e gerar o arquivo TXT com erros, como mostra a Figura 4-1.

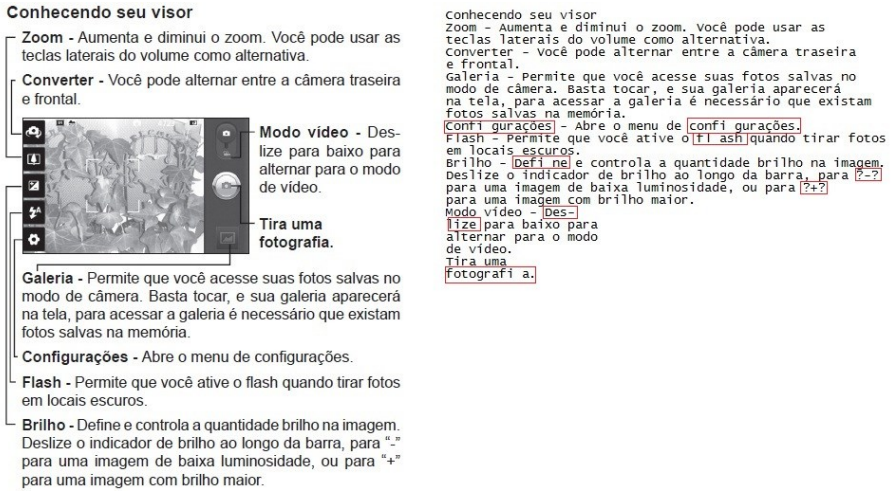

Figura 4-1. Exemplo dos problemas na conversão de PDF para TXT (Fonte: o Autor).

A Figura 4-1 mostra a conversão realizada de um arquivo no formato PDF (lado esquerdo da figura) para o formato TXT (lado direito da figura). Os problemas decorrentes da conversão do arquivo foram marcados em vermelho. Algumas palavras tiveram um espaço em branco incluído, separando-as em duas, e as aspas não foram reconhecidas.

Outros documentos, como casos de uso ou lista de requisitos também poderiam ser utilizados. Os documentos de casos de uso não foram utilizados porque não foi possível coletar documentos de empresas para utilização. Listas de requisitos não foram utilizadas, pois já descrevem as funcionalidades de forma organizada e simplificada.

4.2 Fase 2 – Identificar a técnica que será utilizada para identificação e classificação automática das funcionalidades

Inicialmente foram identificadas duas ferramentas: PorOnto (ZAHRA; MALUCELLI, 2009), baseada no parser TreeTagger e LX-Parser (SILVA et. al., 2010).

Para a identificação e classificação automática das funcionalidades foram realizadas simulações com a ferramenta PorOnto (ZAHRA; MALUCELLI, 2009) que foi desenvolvida para construir ontologias de forma semiautomática, utilizando como artefatos de entrada arquivos em formato PDF. Esta ferramenta foi utilizada porque realiza busca de padrões em textos escritos em língua portuguesa, e é gratuita.

Para as simulações realizadas na ferramenta PorOnto, foram utilizados manuais de celular, conforme definido na Fase 1, encontrados gratuitamente na *web*.

A ferramenta PorOnto permite a busca e visualização da frequência de termos simples ou compostos encontrados nos arquivos fornecidos como entrada. Como resultado das simulações realizadas, verificou-se que vários termos compostos que foram identificados nos manuais poderiam ser considerados funcionalidades. Para identificação dos termos, a ferramenta PorOnto utiliza a ferramenta de anotação linguística TreeTagger (SCHMID, 1994). No trabalho apresentado em (ZAHRA; MALUCELLI, 2009), esta ferramenta foi selecionada após a comparação entre as ferramentas gratuitas de anotação linguística disponíveis na época, portanto uma nova busca foi realizada para definir a ferramenta que seria utilizada nesta proposta.

A única ferramenta mais recente encontrada de anotação linguística para língua portuguesa, gratuita e disponível integralmente para uso foi o LX-Parser (SILVA et. al., 2010). Portanto, foram realizados testes para que fosse possível escolher qual das duas ferramentas seria utilizada na proposta, TreeTagger ou LX-Parser.

Os testes foram realizados com a implementação de um algoritmo que, a partir da leitura de manuais em formato texto (TXT), aplicava a anotação linguística e buscava então termos compostos em cada linha do arquivo anotado, utilizando as anotações que indicavam verbos e substantivos, em qualquer ordem e em qualquer número.

Em um teste realizado com apenas um manual, foram identificados 2260 termos compostos, dentre estes 799 foram encontrados apenas pelo LX-Parser, 777 apenas pelo TreeTagger e 684 foram encontradas pelos dois parsers.

Os resultados foram analisados e testes específicos foram realizados buscando identificar a qualidade da anotação realizada pelas ferramentas, já que a busca pelos termos após a anotação era realizada da mesma maneira. Durante os testes foi identificado que a anotação realizada pelo TreeTagger estava mais correta do que a realizada pelo LX-Parser. O LX-Parser foi desenvolvido e treinado com português europeu, talvez por este motivo seus resultados tenham sido inferiores. Portanto, a ferramenta TreeTagger foi selecionada para utilização nesta proposta.

4.3 Fase 3 – Implementar a identificação e classificação automática das funcionalidades

Após a análise dos resultados dos testes do algoritmo implementado na Fase 2, identificou-se que o mesmo algoritmo usado para os testes das ferramentas de anotação linguística poderia ser reaproveitado para a implementação da abordagem. Para isso, foi identificada a necessidade de refinar o algoritmo para:

- definir padrões de busca para cada linha anotada pela ferramenta, incluindo o uso de adjetivos;
- retirar caracteres especiais que atrapalham a realização da anotação linguística pela ferramenta;
- ignorar resultados que contenham verbos auxiliares, pois estes não correspondem a funcionalidades;
- identificar a raiz (*stemm*) das palavras anotadas para que funcionalidades escritas em tempos verbais diferentes possam ser comparadas corretamente; e
- identificar verbos sinônimos.

Além dos itens listados acima, relacionados diretamente ao algoritmo, também foi verificada a necessidade de "limpar" os arquivos texto utilizados para o processamento. Este processo implica diretamente na qualidade final do processamento, pois dependendo do formato do manual em PDF (imagens entre o texto, texto em várias colunas), sua versão em formato TXT, gerado automaticamente, pode conter problemas que atrapalham a busca pelos padrões no algoritmo, como por exemplo, frases incompletas e palavras separadas por espaço em branco, como pôde ser visto anteriormente na Figura 4-1. Por este motivo o arquivo em formato TXT precisou ser pré-processado, utilizando as seguintes regras:

- deve conter uma frase por linha (sem necessidade de incluir pontuação);
- não deve possuir palavras separadas incorretamente por espaços em branco ou hifenização;

Para aumentar a velocidade de execução do algoritmo, alguns capítulos dos manuais, que não contêm funcionalidades, podem ser retirados, como por exemplo: apresentações da marca/fabricante, índice, introdução, cuidados gerais (segurança), especificações técnicas, endereços, garantia, acessórios.

Além de aumentar a velocidade de execução do algoritmo, a retirada destes capítulos diminuiu o número de funcionalidades encontradas pelo algoritmo, reduzindo, portanto o trabalho manual de análise dos resultados, realizado posteriormente pelo especialista de domínio.

O algoritmo desenvolvido para identificar e classificar automaticamente as funcionalidades é composto por cinco etapas, apresentadas na Figura 4-2:

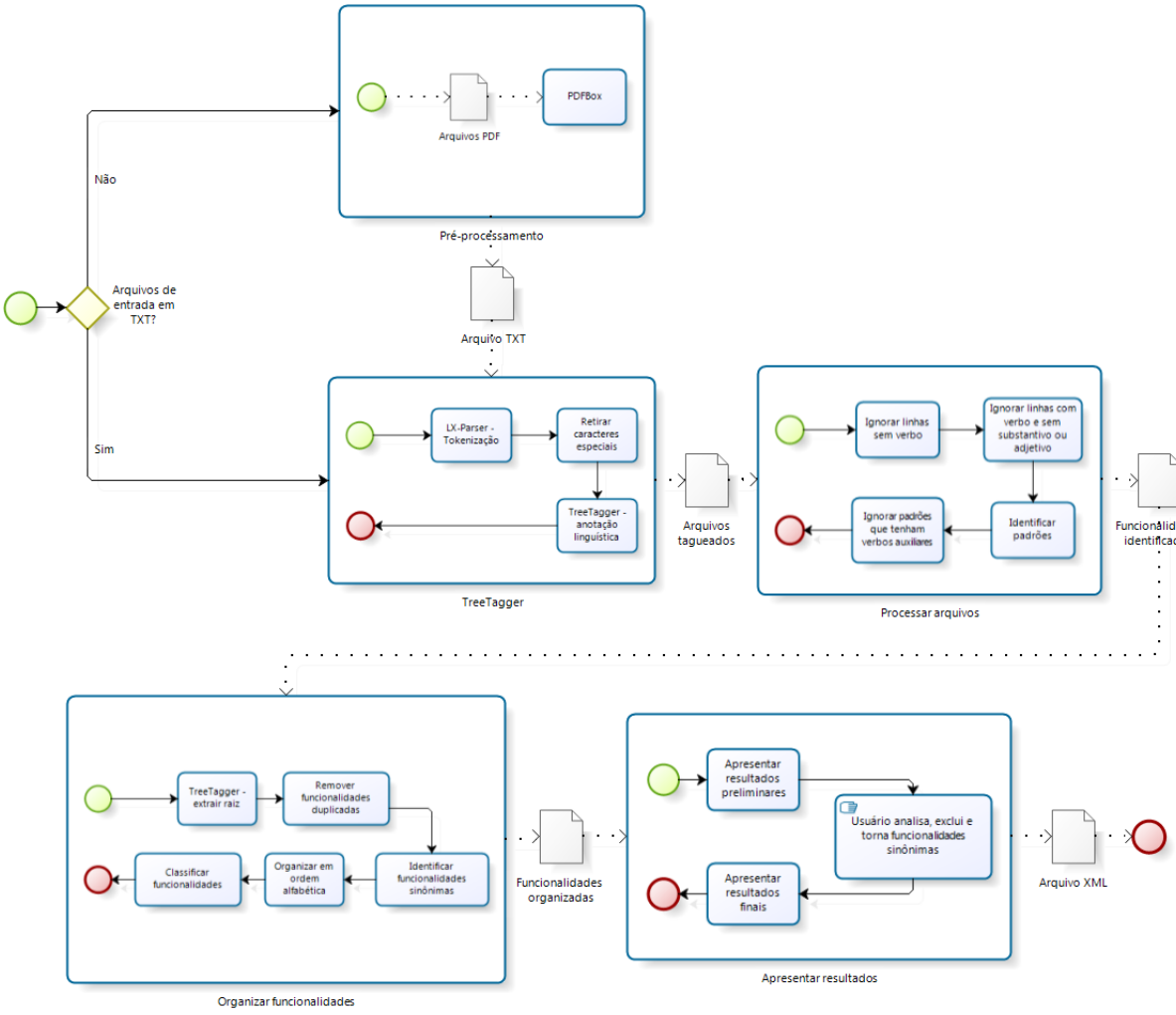

Figura 4-2. Processo proposto (Fonte: o Autor).

4.3.1 Etapa 1 – Pré-processamento

O usuário informa o diretório onde se encontram os manuais dos produtos que serão utilizados para criação da LPS. Se os manuais estiverem em formato PDF, são transformados para o formato TXT com a biblioteca PDFBox (APACHE, 2012) e salvos no mesmo diretório.

4.3.2 Etapa 2 - TreeTagger

Os manuais transformados em arquivos no formato TXT são processados pelo TreeTagger. Para isto, cada linha lida do arquivo é transformada em uma lista de palavras separadas por espaço em branco (tokenização), utilizando uma biblioteca do LXParser, e são retirados caracteres especiais que prejudicam a anotação linguística realizada, como o parênteses, colchetes, chaves e caracteres como “#@?!”. Estes caracteres foram identificados por meiodos testes realizados durante a seleção da ferramenta de anotação linguística que seria utilizada na proposta.

Os arquivos são então processados pelo TreeTagger. Para cada arquivo processado, outro é gerado e salvo em uma nova pasta, contendo as anotações linguísticas geradas pela ferramenta. O TreeTagger realiza a anotação linguística utilizando as tags especificadas no Quadro 4-1.

Quadro 4-1. Tags do TreeTagger, adaptado de (SCHMID, 1994)

Tag	Descrição
ADJ	Adjetivo
ADV	Advérbio
DET	Determinante
CARD	Número cardinal / ordinal
NOM	Nome Comum / Próprio
P	Pronome
PREP	Preposição
V	Verbo
I	Interjeição
VIRG	Separadores dentro da oração
SENT	Separadores de orações
F	Palavra estrangeira

4.3.3 Etapa 3 – Processar Arquivos

Os arquivos que contêm as anotações linguísticas são processados, linha a linha, em busca de funcionalidades. As linhas que não possuem verbo, ou as que possuem verbo, porém não possuem substantivo ou adjetivo, são ignoradas. As que restam, são processadas com a finalidade de identificar sequências de palavras anotadas da seguinte forma:

- VERBO + SUBSTANTIVO + ADJETIVO
- VERBO + ADJETIVO + SUBSTANTIVO
- VERBO + SUBSTANTIVO + SUBSTANTIVO
- VERBO + SUBSTANTIVO.

Uma linha do arquivo pode conter nenhuma, uma ou mais sequências válidas. Ou seja, na linha “Para enviar mensagem de texto use o telefone”, seriam encontradas duas sequências válidas: “enviar mensagem texto” e “use telefone”. É importante destacar que a linha inteira é anotada de uma só vez, para que em seguida os padrões desejados sejam procurados. Por exemplo, a linha acima seria anotada linguisticamente da seguinte forma: (PRP para)(V enviar)(NOM mensagem)(PRP de)(NOM texto)(V use)(DET o)(NOM telefone).

Em seguida, apenas os verbos, substantivos e adjetivos são levados em consideração, e o formato da linha inteira é identificado, neste caso, como VERBO + SUBSTANTIVO + SUBSTANTIVO + VERBO + SUBSTANTIVO (ou seja “enviar mensagem texto use telefone”). Na sequência, os padrões procurados são comparados com o formato identificado na frase, se existe correspondência, as palavras que correspondem ao padrão são retiradas da linha e são consideradas uma funcionalidade. A análise continua, sem estas palavras, até o final da linha.

No exemplo anterior, primeiro seria identificada a correspondência com o padrão VERBO + SUBSTANTIVO + SUBSTANTIVO (enviar mensagem texto) e em seguida a correspondência com o padrão VERBO + SUBSTANTIVO (use telefone).

As sequências válidas identificadas são analisadas novamente, para verificar quais delas estão relacionadas a verbos auxiliares, sendo então descartadas. A lista de verbos auxiliares utilizada foi retirada do Dicionário Priberam da Língua Portuguesa (PRIBERAM, 2012).

4.3.4 Etapa 4 – Organizar Funcionalidades

As funcionalidades identificadas são processadas novamente pelo TreeTagger, desta vez com outra finalidade: identificar a raiz de cada palavra, para que seja possível a comparação de funcionalidades que estejam relacionadas ao mesmo substantivo, porém este aparece de formas diferentes, como por exemplo: “enviar mensagem” e “enviar mensagens”.

Este processamento também possibilita a avaliação dos verbos sinônimos. As funcionalidades são organizadas de acordo com o substantivo contido, ao mesmo tempo em que as funcionalidades repetidas são removidas e as funcionalidades sinônimas são identificadas. Os verbos sinônimos utilizados nesta etapa foram retirados do dicionário de sinônimos OpenThesaurusPT (OPENTHESAURUSPT, 2012). O nome dos arquivos processados que continham as funcionalidades são

armazenados para que as funcionalidades possam ser classificadas dentro da família. Por fim, as funcionalidades são organizadas em ordem alfabética.

4.3.5 Etapa 5 – Apresentar Resultados

As funcionalidades são apresentadas ao usuário, com suas respectivas funcionalidades sinônimas, quando aplicável, e com sua classificação identificada, que foi realizada com base na definição de (LINDEN et al., 2007) que afirma que a variabilidade pode ser separada em três tipos principais: comunalidades, o que é comum para todos os produtos; variabilidades, comum apenas a alguns produtos da família; específico de produto, normalmente não é necessário para o negócio. Este último tipo pode não ser integrado no conjunto de ativos da família.

As funcionalidades, neste trabalho, foram classificadas em comuns, variáveis ou opcionais. As funcionalidades comuns são as que foram encontradas em todos os produtos, as variáveis foram encontradas em mais de um produto, porém não em todos, e as opcionais foram encontradas apenas em um produto. Elas são identificadas, pelas cores verde (comuns), amarela (variáveis) e vermelha (opcionais).

O usuário avalia as funcionalidades identificadas e tem a possibilidade de excluir as que desejar e de sinalizar funcionalidades sinônimas. As funcionalidades são então apresentadas novamente ao usuário, com um mecanismo de “filtragem” conforme apresentado na Figura 4-3, para facilitar sua visualização, e são gravadas em um arquivo de formato XML (*eXtensible Markup Language*) para proporcionar sua reutilização e distribuição.

FUNCIONALIDADES IDENTIFICADAS

Substantivo	Funcionalidade raiz (Funcionalidade original)	Classificação	Produto
CONTATO	apagar contato (apagar contatos)	COMUM	LG KB775 LG Optimus 2X P990 Samsung Corby
CONTATO	chamar contato (chamar contato)	OPCIONAL	LG Optimus 2X P990
MENSAGEM	enviar mensagem (enviar mensagem)	VARIÁVEL	LG Optimus 2X P990 Samsung Corby

Figura 4-3. Lista das funcionalidades identificadas e classificadas (Fonte: o Autor).

4.4 Fase 4 – Avaliar a identificação e classificação automática das funcionalidades

Nesta seção serão apresentados os resultados dos experimentos realizados com o objetivo de avaliar a identificação e classificação automática das funcionalidades. Foram realizados 3 experimentos.

4.4.1 Pré-experimento

Foram processadas três páginas de um manual de celular retirado da *web*, relacionadas às funcionalidades listadas no capítulo Contatos. O processamento foi manual e também automático, com uso do algoritmo proposto, para ser possível realizar uma comparação. Este experimento foi realizado pelo próprio pesquisador.

O processamento manual identificou 41 funcionalidades em aproximadamente 8 minutos. As mesmas páginas foram processadas pelo algoritmo proposto e apresentadas ao usuário após 9 segundos de processamento (desconsiderando o tempo para converter o arquivo em TXT). Para este experimento o arquivo foi pré-processado manualmente, de modo a manter apenas o capítulo de Contatos e também foi "limpo" de acordo com os itens informados anteriormente na Seção 4.3:

- deve conter uma frase por linha (sem necessidade de incluir pontuação);
- não deve possuir palavras separadas incorretamente por espaços em branco ou hifenização.

O algoritmo encontrou 106 funcionalidades, divididas entre 42 substantivos. As funcionalidades foram todas apresentadas ao usuário, que demorou 04 minutos e 30 segundos para excluir 45 funcionalidades que não eram relevantes, restando 61 funcionalidades relevantes.

Entre estas 61 funcionalidades relevantes, 26 também haviam sido encontradas na busca manual e 35 foram localizadas apenas pelo algoritmo e consideradas relevantes pelo usuário.

Das 41 funcionalidades encontradas na busca manual, apenas 14 não foram localizadas pelo algoritmo. Estas funcionalidades estão listadas no Quadro 4-2, assim como a frase do arquivo onde cada funcionalidade deveria ter sido encontrada, com suas respectivas anotações linguísticas.

Quadro 4-2. Funcionalidades não encontradas no algoritmo (Fonte: o Autor)

Funcionalidade	Frase com anotação linguística
1 - Chamar contato	(P a)(V partir)(PRP+DET da)(NOM lista)(V toque)(DET o)(NOM contato)(PR que)(V deseja)(V chamar)
2 - Enviar cartão de visita do contato via mensagem multimídia	(NOM escolha)(V enviar)(CONJ como)(DET uma)(NOM mensagem)(PRP de)(NOM texto)(NOM mensagem)(ADJ multimídia)(V usando)(DET o)(F email)(CONJ ou)(V via)(NOM bluetooth)
3 - Enviar cartão de visita do contato via email	(NOM escolha)(V enviar)(CONJ como)(DET uma)(NOM mensagem)(PRP de)(NOM texto)(NOM mensagem)(ADJ multimídia)(V usando)(DET o)(F email)(CONJ ou)(V via)(NOM bluetooth)
4 - Procurar contato	(V procurar)(PRP por)(QUOTE -)(V permite)(V buscar)(V digitando)(DET o)(NOM número)(CONJ ou)(NOM grupo). No entanto, outra funcionalidade encontrada se refere à busca de contatos pelo grupo ou pelo número do telefone: “digitar número grupo (digitando número grupo)”
5 - Configurar lista de contatos para gravar no telefone	(NOM escolha)(P se)(V quer)(V ver)(DET os)(NOM contatos)(ADJ salvos)(PRP+DET no)(NOM telefone)(CONJ e)(PRP+DET no)(ADV sim)(ADV somente)(PRP+DET no)(NOM telefone)(CONJ ou)(ADV somente)(PRP+DET no)(ADV sim). No entanto, outras funcionalidades encontradas se referem à configuração da lista de contatos: “adaptar configuração contato (adaptar configurações contatos), alterar configuração contato (alterando configurações contato), configurar listar contato (configurar lista contatos)”
6- Configurar lista de contatos para gravar no cartão SIM	
7 - Configurar lista de contatos para gravar no telefone e no cartão SIM	
8 - Configurar lista de contatos para mostrar o primeiro nome	(P você)(ADV também)(V pode)(V selecionar)(V mostrar)(DET o)(CARD primeiro)(CONJ ou)(DET o)(ADJ último)(NOM nome)(PRP+DET do)(NOM contato). No entanto, outra funcionalidade encontrada se refere à esse tipo de configuração: “mostrar último nome (mostrar último nome)”
9 - Copiar contatos entre o cartão SIM e o telefone	(V copiar)(QUOTE -)(V copia)(ADJ seus)(NOM contatos)(PRP+DET do)(ADV sim)(PRP para)(DET o)(ADJ seu)(NOM dispositivo)(CONJ ou)(PRP+DET do)(ADJ seu)(NOM dispositivo)(PRP para)(DET o)(ADV sim)
10 - Mover contatos entre o cartão SIM e o telefone	(V mover)(QUOTE -)(P este)(V trabalha)(PRP+DET da)(ADJ mesma)(NOM forma)(PR que)(V copiar)(CONJ mas)(DET o)(NOM contato)(V será)(ADJ salvo)(ADV somente)(PRP+DET na)(NOM localidade)(PRP para)(DET o)(PR qual)(V for)(V movido)(PRP por)(NOM exemplo)(P se)(P você)(V mover)(DET o)(NOM contato)(PRP+DET do)(ADV sim)(PRP para)(DET o)(NOM telefone)(P ele)(V será)(V apagado)(PRP+DET da)(NOM memória)(PRP+DET do)(X sm)
11 - Enviar contatos via bluetooth	(V enviar)(ADJ todos)(DET os)(NOM contatos)(PRP por)(NOM bluetooth)
12 - Fazer cópia de segurança dos contatos para o telefone	(V permite)(V fazer)(DET uma)(NOM cópia)(PRP de)(ADJ todos)(DET os)(ADJ seus)(NOM contatos)(PRP para)(DET o)(NOM telefone)(CONJ ou)(PRP para)(DET o)(ADV sim). No entanto, outra funcionalidade encontrara se refere à esta: “fazer cópia todo (fazer cópia todos)”
13 - Fazer cópia de segurança dos contatos para o cartão SIM	
14 - Apagar todos os contatos	(V apagar)(NOM contatos)(QUOTE -)(V apagar)(ADJ todos)(DET os)(ADJ seus)(NOM contatos)

Analisando as 14 funcionalidades não encontradas no processamento automático identifica-se que, apesar de algumas não terem sido encontradas exatamente com o texto esperado, outras semelhantes foram identificadas. Isso foi identificado em 7 funcionalidades (50%): funcionalidades de número 4, 5, 6, 7, 8, 12 e 13 no Quadro 4-2. Todas as 14 funcionalidades não foram identificadas no processamento automático pelo mesmo motivo, o estilo de escrita das frases não se encaixou em nenhum dos padrões procurados pelo algoritmo.

O desempenho da abordagem proposta foi analisado com as medidas de *precision* e *recall* (MANNING et al., 2009) que são definidas pelas fórmulas (i) e (ii):

(i) *Precision* = tp / (tp + fp)

(ii) *Recall* = tp / (tp + fn)

Nas fórmulas, tp significa *true positive* (funcionalidades recuperadas relevantes), fp significa *false positive* (funcionalidades recuperadas não-relevantes), e fn significa *false negative* (funcionalidades não recuperadas relevantes).

O *precision* é a fração das funcionalidades recuperadas que são relevantes, o *recall* é a fração das funcionalidades relevantes que foram recuperadas (MANNING et al., 2009).

Para calcular as medidas serão considerados os valores tp = 61 (funcionalidades relevantes consideradas pelo usuário), fp = 45 (fucionalidades recuperadas e excluídas pelo usuário, ou seja, os falsos positivos), fn = 14 (funcionalidades relevantes encontradas pelo usuário, porém não encontradas pelo algoritmo, ou seja, falsos negativos).

Os resultados das medidas são *precision* = 0,57 e *recall* = 0,81, ou seja, dos resultados recuperados 81% são relevantes para o critério pesquisado e 57% são precisos, isto é, segundo um avaliador humano, a resposta seria considerada correta para o critério pesquisado.

A partir destas medidas também pode-se extrair a medida *F-measure*, que é uma média harmônica entre as medidas *precision* e *recall*, onde quanto mais próximo de 1 for o resultado, melhor. A fórmula da medida *F-measure* é definida na fórmula (iii), onde r = *recall* e p = *precision*:

(iii) *F-measure* = 2 * r * p / r + p.

O resultado da medida *F-measure* é de 0,67.

Para melhorar as medidas *precision*, *recall* e *F-measure*, é possível alterar o algoritmo para levar em consideração, na identificação das funcionalidades, apenas aquelas que tenham uma palavra específica considerada como substantivo ou adjetivo pelo TreeTagger.

Fazendo uma simulação com os resultados do processamento manual e automático apresentados anteriormente, considerando para o processamento automático apenas as funcionalidades que apresentaram como substantivo ou como adjetivo a palavra Contato, teríamos os seguintes valores: tp = 26, fp = 10 e fn = 14 (idem anterior), resultando em *precision* = 0,72; *recall* = 0,65; e *F-Measure* = 0,68.

O *recall* diminuiu porque o número de funcionalidades relevantes identificadas foi reduzido em comparação ao cálculo anterior, no entanto o número de funcionalidades relevantes que não foram identificadas permaneceu o mesmo.

Esta possível alteração no algoritmo não foi contemplada neste trabalho, pois demandaria como artefato de entrada, além dos documentos com as funcionalidades, uma lista dos termos relevantes do domínio que deveriam ser considerados. Esta lista provavelmente precisaria ser feita pelo especialista de domínio, aumentando a necessidade da sua presença durante a fase de definição de escopo e, portanto, contrariando os objetivos iniciais deste trabalho que eram de reduzir a necessidade do especialista na fase de definição do escopo.

Analisando os resultados obtidos neste pré-experimento, conclui-se que o tempo gasto para identificação das funcionalidades diminuiu mais de 40%, enquanto o número de funcionalidades relevantes encontradas aumentou mais de 48%.

4.4.2 Segundo experimento

Os resultados apresentados durante o pré-experimento demonstraram que é possível semi-automatizar a análise de documentos na busca por funcionalidades de produtos que serão utilizadas posteriormente para a criação de uma LPS.

Foi realizado um novo experimento utilizando as mesmas três páginas de um manual de celular retirado da *web*, relacionadas às funcionalidades para Contatos. O processamento também foi manual e automático, com uso do algoritmo proposto, para ser possível realizar uma comparação, porém foi realizado com 6 pessoas, sem a participação do pesquisador. As características das pessoas que participaram do experimento estão descritas no Quadro 4-3.

Quadro 4-3. Características das pessoas (Fonte: o Autor)

ID	Formação	Relação com o domínio - celular
P1	Mestre em informática	Usuário
P2	Mestrando em informática	Usuário
P3	Doutorando em informática	Usuário
P4	Mestrando em informática	Usuário
P5	Pós-graduado em informática	Usuário
P6	Mestrando em informática	Usuário

A Tabela 4-4 apresenta a quantidade de funcionalidades encontradas manualmente e por meio do processo automático, por cada pessoa que participou do experimento, assim como o tempo gasto para cada processo (manual ou automático).

Tabela 4-4. Funcionalidades e tempo gasto (Fonte: o Autor)

	P1	P2	P3	P4	P5	P6
Encontradas manualmente	33	24	34	27	27	22
Tempo gasto manualmente	00:10:00	00:06:20	00:08:17	00:06:17	00:12:00	00:07:00
Encontradas automaticamente	64	54	52	61	53	78
Tempo de análise do especialista	00:06:00	00:05:10	00:04:40	00:05:55	00:09:00	00:06:00

As duas primeiras linhas do quadro mostram, respectivamente, a quantidade de funcionalidades que cada pessoa que participou do experimento encontrou durante a leitura das páginas e extração manual das funcionalidades, e em seguida o tempo gasto neste processo. Por exemplo, a pessoa P1 encontrou 33 funcionalidades na análise manual e consumiu 10 minutos. A Figura 4-4 apresenta graficamente os dados relacionados à quantidade de funcionalidades encontradas.

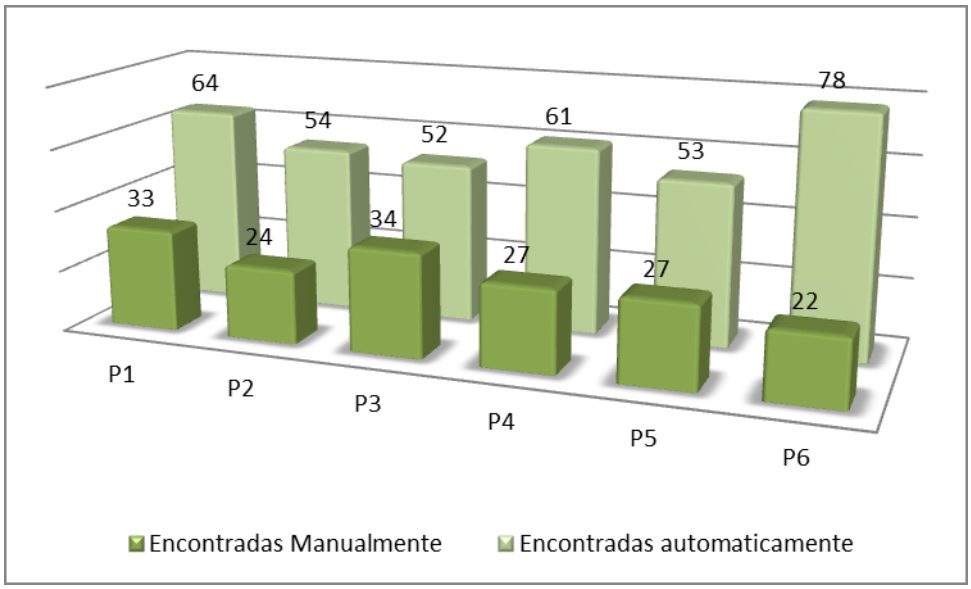

Figura 4-4. Funcionalidades encontradas – segundo experimento (Fonte: o Autor).

As funcionalidades que estão na linha “Encontradas automaticamente” são as funcionalidades que foram encontradas pelo processo automático e que não foram excluídas pelo usuário quando analisou o resultado do processamento automático, ou seja, são as funcionalidades que o usuário considerou corretas e relevantes. Neste experimento, assim como no pré-experimento, foram encontradas automaticamente 106 funcionalidades e tiveram que ser analisadas. A linha “Tempo de análise do especialista” mostra o tempo gasto nesta análise feita pelos usuários que participaram do experimento. Por exemplo, a pessoa P1 considerou que 64 funcionalidades encontradas pelo processo automático eram relevantes e descartou 42 funcionalidades (falsos positivos). Para esta análise ela consumiu 6 minutos.

Analisando a Tabela 4-4 como um todo, pode-se verificar que para todos os casos a quantidade de funcionalidades encontradas por meio do processo automático foi maior que a quantidade encontrada no processo manual. De forma semelhante, o tempo gasto para o processo automático foi menor que o tempo gasto no processo manual, incluindo a análise do resultado e a definição das funcionalidades relevantes. A Figura 4-5 apresenta graficamente os dados relacionados ao tempo gasto.

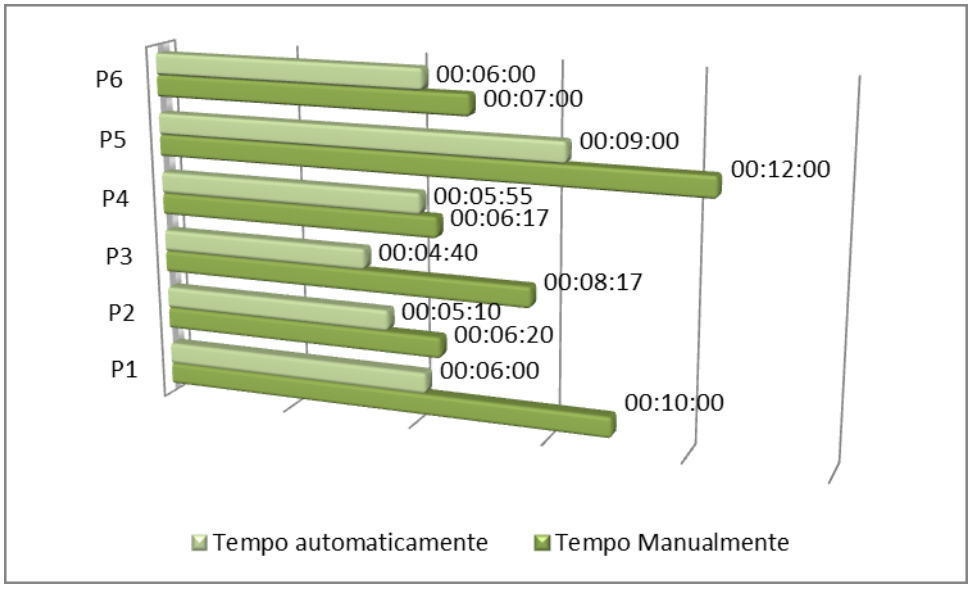

Figura 4-5. Tempo gasto – segundo experimento (Fonte: o Autor).

O tempo gasto no processamento automático pelo algoritmo, antes de apresentar o resultado ao usuário, foi o mesmo tempo do pré-experimento: 9 segundos (desconsiderando o tempo para converter o arquivo em TXT). O tempo é o mesmo pois, para facilitar a realização deste experimento, os participantes

receberam um documento em formato HTML já com o resultado do processamento automático, executado no computador do pesquisador, para então analisar as funcionalidades e considera-las relevantes ou não.

O processamento não foi executado no computador de cada participante pois seria necessária a realização de várias configurações e instalações de softwares que dificultariam a realização do experimento. Além disso, não influenciariam em nada o resultado do experimento, pois as mesmas funcionalidades seriam apresentadas e analisadas.

O documento HTML que os participantes receberam foi desenvolvido para apresentar as funcionalidades excluídas na própria tela, ao final da análise do participante. Os participantes foram instruídos a enviar essa lista ao pesquisador. O tempo gasto nas análises (manual e automática) foi anotado pelos próprios participantes.

A Tabela 4-5 apresenta a variação percentual entre a quantidade de funcionalidades identificadas automaticamente e manualmente e também entre o tempo gasto para a análise manual e a análise do resultado automático. No tempo para a análise do resultado automático foi considerando também o tempo de processamento do algoritmo:

Tabela 4-5. Funcionalidades e tempo gasto - percentuais (Fonte: o Autor)

	P1	P2	P3	P4	P5	P6
% da Quantidade de funcionalidades	93,94%	125,00%	52,94%	125,93%	96,30%	254,55%
% do Tempo Gasto	-38,5%	-16,05%	-41,85%	-3,45%	-23,75%	-12,14%

Em todos os casos, a quantidade de funcionalidades identificadas e consideradas relevantes foi maior no processamento automático, variando de 52,94% a 254,55% de aumento. De forma similar, o tempo gasto para a análise do resultado automático foi menor, variando de -38,5% à -3,45%; ou seja, gastou-se menos tempo na análise dos resultados automáticos, porém o número de funcionalidades identificadas relevantes foi maior. Na média, foram identificadas 124,77% a mais de funcionalidades e gasto 22,62% a menos de tempo.

As funcionalidades identificadas e consideradas relevantes nos processamentos automáticos realizados no experimento foram analisadas para gerar um só conjunto de funcionalidades relevantes e irrelevantes. Foram consideradas relevantes as funcionalidades que não foram excluídas por mais de 50% das

pessoas que participaram do experimento. Da mesma forma, foram consideradas irrelevantes as funcionalidades que foram excluídas por mais de 50% das pessoas.

Os resultados foram os mesmos do pré-experimento: 61 funcionalidades encontradas automaticamente foram consideradas relevantes e 45 funcionalidades foram consideradas irrelevantes, ou seja, excluídas durante a análise do resultado apresentado no processamento automático.

A Tabela 4-6 mostra a quantidade de funcionalidades encontradas automaticamente e consideradas relevantes (duas primeiras linhas), e as encontradas apenas manualmente, para cada pessoa que participou do experimento:

Tabela 4-6. Funcionalidades relevantes - conjunto (Fonte: o Autor)

	P1	P2	P3	P4	P5	P6
Relevantes encontradas apenas automaticamente	36	39	31	41	39	43
Relevantes encontradas manualmente e automaticamente	25	22	30	20	22	18
Relevantes encontradas apenas manualmente	8	2	4	7	5	4
Total de funcionalidades	69	63	65	68	66	65

A soma das duas primeiras linhas em cada coluna é igual ao total de funcionalidades encontradas automaticamente, analisadas e consideradas relevantes, ou seja, 61 funcionalidades. É interessante também verificar que o número total de funcionalidades encontradas por cada pessoa variou pouco, de 63 a 69 funcionalidades. A Figura 4-6 apresenta graficamente estes dados.

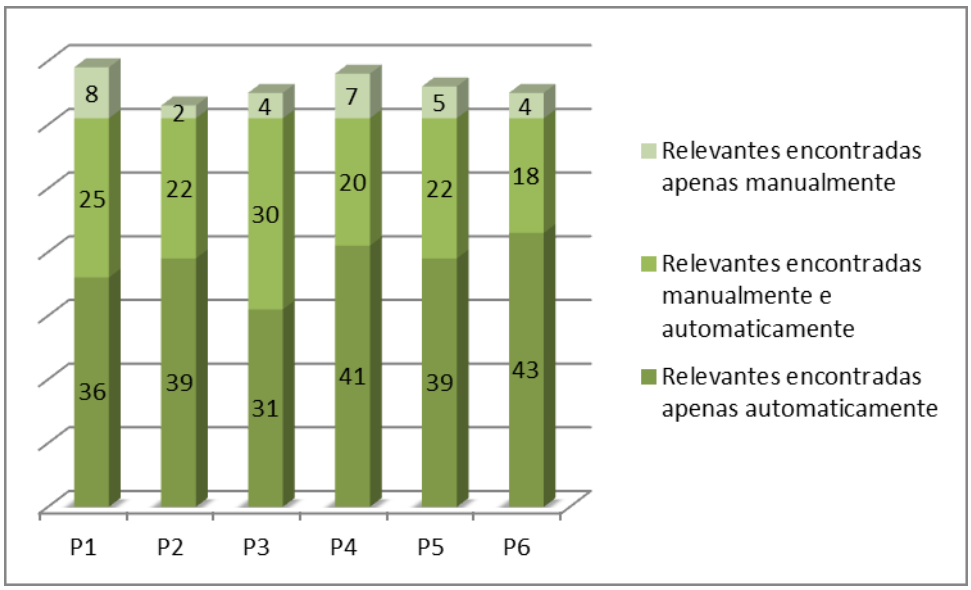

Figura 4-6. Funcionalidades relevantes – segundo experimento (Fonte: o Autor).

Na Tabela 4-7 verifica-se a porcentagem de participação de cada categoria: encontradas apenas automaticamente, encontradas manualmente e automaticamente e encontradas apenas manualmente, em relação ao total de funcionalidades encontradas:

Tabela 4-7. Porcentagem em relação ao total (Fonte: o Autor)

	P1	P2	P3	P4	P5	P6
Encontradas apenas automaticamente	52,17%	61,90%	47,69%	60,29%	59,09%	66,15%
Encontradas manualmente e automaticamente	36,23%	34,92%	46,15%	29,41%	33,33%	27,69%
Encontradas apenas manualmente	11,59%	3,17%	6,15%	10,29%	7,58%	6,15%

Analisando os resultados da Tabela 4-7 é possível verificar que, em todos os casos, as funcionalidades encontradas apenas automaticamente foram as mais representativas no número total de funcionalidades encontradas, variando de 47,69% a 66,15%. Em seguida estão as funcionalidades que foram encontradas de duas formas, automaticamente e manualmente, variando de 27,69% a 46,15% do total. As menos representativas foram as funcionalidades encontradas apenas manualmente, variando de 3,17% a 11,59% do total. A Figura 4-7 apresenta graficamente os dados da Tabela 4-7.

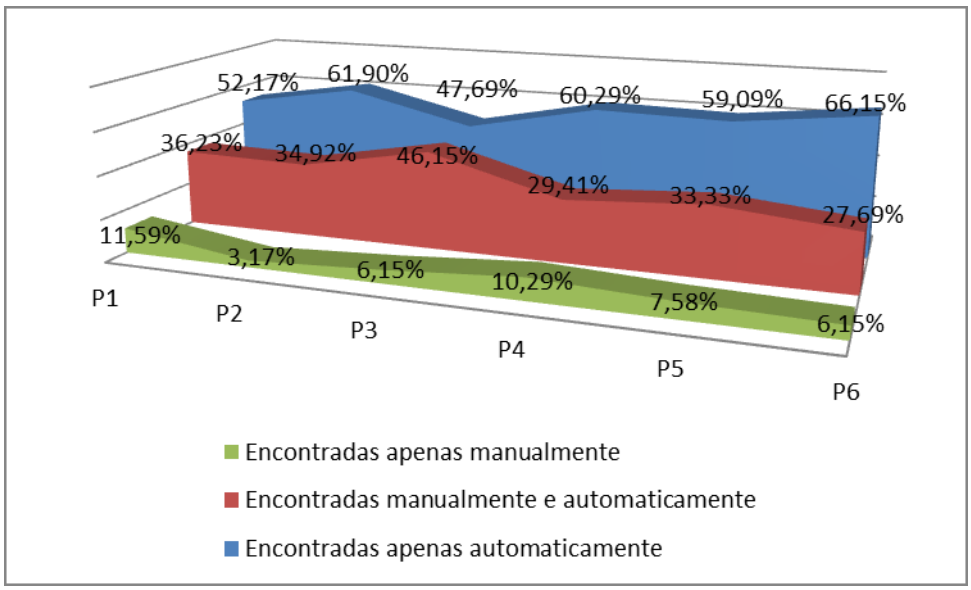

Figura 4-7. Porcentagens – segundo experimento (Fonte: o Autor).

Avaliando a efetividade do processamento automático, se pode somar as duas primeiras categorias que representam todas as funcionalidades encontradas

automaticamente e concluir que sua representatividade no número total de funcionalidades é muito grande: varia de 88,41% (P1) a 96,85% (P2).

Por outro lado, considerando apenas as funcionalidades que foram encontradas manualmente, sua representatividade no número total de funcionalidades varia de 33,85% (P6) a 52,31% (P3).

Realizando uma comparação com os resultados do primeiro experimento, com relação às medidas de *precision* e *recall*, considerando-se a média entre os resultados das 6 pessoas os resultados foram *precision* = 0,57 (antes 0,57), *recall* = 0,92 (antes 0,81) e *F-measure* = 0,70 (antes 0,67). Ou seja, dos resultados recuperados 92% são relevantes para o critério pesquisado e 57% são precisos.

Estes cálculos levaram em consideração o total de 106 funcionalidades identificadas pelo processamento automático e as funcionalidades consideradas relevantes como *true positive* (positivo verdadeiro), as funcionalidades excluídas como *false positive* (falsos positivos) e as encontradas apenas manualmente como *false negative* (falsos negativos). A Tabela 4-8 apresenta os dados utilizados nestes cálculos.

Tabela 4-8. Análise dos resultados obtidos (Fonte: o Autor)

	P1	**P2**	**P3**	**P4**	**P5**	**P6**
True positive:	64	54	52	61	53	78
False positive:	42	52	54	45	53	28
False negative:	8	2	4	7	5	4
Precision:	0,60	0,51	0,49	0,58	0,50	0,74
Recall:	0,89	0,96	0,93	0,90	0,91	0,95
F-measure:	0,72	0,67	0,64	0,70	0,65	0,83

As funcionalidades encontradas apenas manualmente, *false negative* (falsos negativos) da Tabela 4-8, foram analisadas para identificar o motivo pelo qual o algoritmo não as encontrou. O Quadro 4-9 apresenta as funcionalidades encontradas manualmente pelas pessoas que participaram do experimento e os trechos do manual onde cada funcionalidade poderia ter sido encontrada, com suas anotações linguísticas.

No Quadro 4-9, os trechos tiveram algumas marcações feitas em negrito, mostrando as funcionalidades que foram identificadas pelo algoritmo, e em vermelho para alguns casos, correspondendo às palavras que formariam as funcionalidades que apenas os usuários encontraram.

Quadro 4-9. Funcionalidades encontradas apenas manualmente (Fonte: o Autor)

ID	Funcionalidade	Usuários	Trechos
1	Chamar contato	Todos	(P a)**(V partir)**(PRP+DET da)**(NOM lista)(V toque)**(DET o)**(NOM contato)**(PR que)(V deseja)(V chamar) (NOM toque)(V chamar)(CONJ ou)**(V pressione)**(DET a)**(NOM tecla)**(PRP para)(V iniciar)(DET a)(ADJ chamada)
2	Enviar todos os contatos por Bluetooth	Todos	**(V enviar)(ADJ todos)**(DET os)**(NOM contatos)**(PRP por)(NOM bluetooth) (V envia)(ADJ todos)(DET os)(ADJ seus)(NOM contatos)(PRP para)(ADJ outro)(NOM dispositivo)(CONJ ou)(NOM computador)**(V usando)**(DET o)**(NOM bluetooth)**
3	Adicionar endereços de email	P1, P4, P5, P6	(P você)(V pode)**(V adicionar)**(PRP até)(CARD dois)**(NOM endereços)**(PRP de)(F email)
4	Adicionar página inicial	P1	(P você)(ADV também)(V pode)**(V adicionar)**(DET uma)**(NOM imagem)(NOM toque)**(NOM página)(NOM inicial)(NOM end)(NOM casa)(NOM nome)(PRP+DET da)(NOM empresa)(NOM notas)
5	Adicionar endereço	P1	(P você)(ADV também)(V pode)**(V adicionar)**(DET uma)**(NOM imagem)(NOM toque)**(NOM página)(NOM inicial)(NOM end)(NOM casa)(NOM nome)(PRP+DET da)(NOM empresa)(NOM notas)
6	Adicionar nome da empresa	P1	(P você)(ADV também)(V pode)**(V adicionar)**(DET uma)**(NOM imagem)(NOM toque** (NOM página)(NOM inicial)(NOM end)(NOM casa)(NOM nome)(PRP+DET da)(NOM empresa)(NOM notas)
7	Adicionar notas	P1	(P você)(ADV também)(V pode)**(V adicionar)**(DET uma)**(NOM imagem)(NOM toque)**(NOM página)(NOM inicial)(NOM end)(NOM casa)(NOM nome)(PRP+DET da)(NOM empresa)(NOM notas)
8	Mostrar o primeiro nome de contato	P1, P3	(P você)(ADV também)(V pode)(V selecionar)**(V mostrar)**(DET o)(CARD primeiro)(CONJ ou)(DET o)**(ADJ último)(NOM nome)**(PRP+DET do)(NOM contato)
9	Apagar todos os seus contatos	P3	**(V apagar)(NOM contatos)**(QUOTE -)(V apagar)(ADJ todos)(DET os)(ADJ seus)(NOM contatos)
10	Digitar letras	P4	Nenhum trecho encontrado com essas palavras em uma mesma frase.
11	Digitar números	P4	Nenhum trecho encontrado com essas palavras em uma mesma frase.
12	Classificar contato	P4	Nenhum trecho encontrado com essas palavras em uma mesma frase.
13	Enviar Mensagem Multimídia para contato	P4, P6	(NOM escolha)**(V enviar)**(CONJ como)(DET uma)**(NOM mensagem)**(PRP de)**(NOM texto)**(NOM mensagem)(ADJ multimídia)(V usando)(DET o)(F email)(CONJ ou)**(V via)(NOM bluetooth)**
14	Mantém os contatos, mesmo apagando o grupo de toque	P5	(NOM nota)(P se)(P você)**(V apagar)**(DET um)**(NOM grupo)**(DET os)**(NOM contatos)**(PR que)(V estão)**(V assinalados)**(PRP a)(DET esse)**(NOM grupo)**(ADV não)(V serão)(ADJ perdidos)
15	Abre o teclado ao clicar	P5	**(V dica)**(DET o)**(NOM teclado)(ADJ alfanumérico)(V**

em caixas de texto	**aparece)**(DET uma)**(NOM vez)**(PR que)**(V tocar)**(DET um)**(NOM espaço)**(PRP em)**(NOM branco)**(PR que)(V necessite)(V digitar)(ADJ algum)(V dado)

Para cada funcionalidade encontrada apenas manualmente, é explicado a seguir o motivo pelo qual o algoritmo não as localizou.

- no ID 1, a funcionalidade poderia ter sido localizada em dois trechos do manual. No primeiro trecho, não foi encontrada porque o estilo de escrita da frase não se encaixou em nenhum dos padrões procurados pelo algoritmo. As funcionalidades encontradas no primeiro trecho foram: “partir lista” e “toque contato”. Com relação ao segundo trecho, ela não foi encontrada, pois a palavra “chamada” foi incorretamente anotada como adjetivo e não como substantivo. Se a palavra tivesse sido anotada corretamente, a funcionalidade “iniciar chamada” seria identificada. A funcionalidade encontrada no segundo trecho foi: “pressione tecla”.
- no ID 2, a funcionalidade também poderia ter sido localizada em dois trechos do manual. Não foi encontrada em nenhum dos dois porque o estilo de escrita da frase não se encaixou em nenhum dos padrões procurados pelo algoritmo. A funcionalidade encontrada no primeiro trecho foi: “enviar todos contatos”. A funcionalidade encontrada no segundo trecho foi: “usando bluetooth”.
- no ID 3, a funcionalidade poderia ser localizada em um trecho do manual. Não foi encontrada porque a palavra “email” foi anotada como palavra estrangeira (F) e não como substantivo. Se a palavra tivesse anotada corretamente a funcionalidade “adicionar endereços email” seria identificada. A funcionalidade encontrada foi: “adicionar endereços”.
- nos IDs 4, 5, 6 e 7 a funcionalidade poderia ser localizada no mesmo trecho do manual. Não foi encontrada porque o estilo de escrita da frase não se encaixou em nenhum dos padrões procurados pelo algoritmo. A funcionalidade encontrada foi: “adicionar imagem toque”.
- no ID 8, a funcionalidade poderia ser localizada em um trecho do manual. Não foi encontrada porque o estilo de escrita da frase não se encaixou em nenhum dos padrões procurados pelo algoritmo. A funcionalidade encontrada foi: “mostrar último nome”.

- no ID 9, a funcionalidade poderia ser localizada no mesmo trecho do manual. Não foi encontrada porque a palavra “seus” foi anotada como adjetivo e não como pronome. Se a palavra tivesse anotada corretamente a funcionalidade “apagar todos contatos” seria identificada. A funcionalidade encontrada foi: “apagar contatos”.
- para os IDs 10, 11 e 12, nenhum trecho do manual que tivesse essas palavras em uma linha foi localizado.
- no ID 13, a funcionalidade poderia ser localizada em um trecho do manual. Não foi encontrada porque o estilo de escrita da frase não se encaixou em nenhum dos padrões procurados pelo algoritmo. As funcionalidades encontradas foram: “enviar mensagem texto” e “via bluetooth”.
- no ID 14, a funcionalidade poderia ser localizada em um trecho do manual. Não foi encontrada porque o estilo de escrita da frase não se encaixou em nenhum dos padrões procurados pelo algoritmo. As funcionalidades encontradas foram: “apagar grupo contatos” e “assinalados grupo”.
- no ID 15, a funcionalidade poderia ser localizada em um trecho do manual. Não foi encontrada porque o estilo de escrita da frase não se encaixou em nenhum dos padrões procurados pelo algoritmo. As funcionalidades encontradas foram: “dica teclado alfanumérico”, “aparece vez” e “tocar espaço branco”.

A maioria das funcionalidades não foi encontrada pelo algoritmo porque o estilo de escrita da frase não se encaixou em nenhum dos padrões procurados, ou porque alguma palavra foi anotada incorretamente, prejudicando a correta identificação dos padrões. O Quadro 4-9 também mostrou que, levando em consideração o conjunto de todas as pessoas que participaram do experimento, 15 funcionalidades ao todo foram encontradas apenas manualmente.

Considerando que o processamento realizado em apenas 3 páginas retornou 106 funcionalidades e o manual processado possui 65 páginas (já desconsiderando os capítulos como introdução, índices e etc. conforme citado anteriormente neste trabalho), pôde-se concluir que era necessária a realização de ajustes no algoritmo para melhores resultados, pois o número de funcionalidades identificadas no manual como um todo, considerando a média encontrada em 3 páginas, seria de mais de 2200 funcionalidades.

Ao processar manuais de uma mesma família de produtos, é comum que sua estrutura seja muito parecida, assim como alguns de seus conteúdos textuais. Dessa maneira, identificou-se a necessidade de avaliar os benefícios de incluir no algoritmo uma lista de "*stop features*", que seria populada cada vez que o usuário excluísse funcionalidades irrelevantes durante o processo já existente. Desta forma, as funcionalidades que estiverem na lista de "stop features" seriam ignoradas e não seriam apresentadas ao usuário, diminuindo ainda mais o tempo de análise dos resultados e possibilitando uma espécie de treinamento do algoritmo para determinado domínio, melhorando os resultados finais.

4.4.3 Terceiro experimento

A possibilidade de utilização de uma lista de "*stop features*" identificada no experimento anterior foi analisada e implementada antes da realização deste terceiro experimento, pois neste buscou-se os artefatos que seriam utilizados para criar uma linha completa para posterior avaliação de um novo produto possibilitando a continuidade da avaliação do trabalho proposto.

Os artefatos selecionados foram manuais da família *Smartphones* da LG, disponíveis na *web.* Os manuais disponíveis foram analisados com relação à sua estrutura, e foi possível identificar 4 padrões, conforme Figura 4-8. Para identificar os padrões foi levada em consideração a estrutura geral do documento como, por exemplo, a separação dos capítulos. Também foi verificado que, em algumas seções, alguns os manuais tinham praticamente o mesmo texto.

	Padrão 1	Padrão 2	Padrão 3
LG Hotmail Phone C570			X
LG Optimus 2X P990	X		
LG Optimus Black P970	X		
LG Optimus GT540		X	
LG Optimus L3 E400	X		
LG Optimus Pro C660	X		

Figura 4-8. Padrão dos manuais LG (Fonte: o Autor).

Os manuais classificados com o padrão 1 são os mais semelhantes, o padrão 2 é semelhante com o padrão 1 porém contém alguns textos e estruturas diferentes e o padrão 3 é o que contém mais estruturas e textos diferentes.

Os quatro últimos manuais não estavam disponíveis para *download*, portanto não puderam ser analisados e utilizados neste trabalho. Foram utilizados 6 manuais, totalizando 282 páginas. Os manuais foram examinados e apenas os capítulos que descrevem as funções dos aparelhos foram utilizados no experimento. Os capítulos de índice, garantia e acessórios, por exemplo, não foram utilizados.

Quatro dos seis manuais disponíveis apresentam um padrão muito semelhante em sua estrutura Figura 4-8, portanto concluiu-se que a lista de "*stop features*" deveria ser implementada, pois poderia realmente diminuir o trabalho manual de análise das funcionalidades encontradas no processo automático.

Portanto, o processo de identificação das funcionalidades apresentado anteriormente na Figura 4-2 teve algumas modificações: as etapas de "Processar Arquivos", "Organizar funcionalidades" e "Apresentar resultados preliminares" foram alteradas para serem acumulativas.

Após a etapa "TreeTagger", que faz a anotação linguística de todos os artefatos disponibilizados pelo usuário para criação da LPS, o algoritmo passa a tratar um artefato de cada vez. O primeiro artefato é processado, organizado e apresentado ao usuário. O usuário analisa as funcionalidades, exclui as que deseja, cria funcionalidades sinônimas se necessário e finaliza a análise deste artefato. Neste momento, as funcionalidades que foram excluídas pelo usuário são guardadas em uma lista de "*stop features*".

O algoritmo processa o próximo artefato, e leva em consideração a lista de "*stop features*". Se alguma funcionalidade identificada estiver nesta lista, ela é descartada e não é apresentada ao usuário como resultado do processamento deste segundo artefato. As funcionalidades deste segundo artefato são organizadas e classificadas, junto com as que já haviam sido identificadas no primeiro artefato, e apresentadas novamente ao usuário para exclusão e avaliação de funcionalidades sinônimas.

Ao invés de processar, organizar e apresentar as funcionalidades encontradas em todos os artefatos disponibilizados pelo usuário de uma só vez, as informações são apresentadas de forma acumulativa para que seja possível a utilização da lista de "*stop features*" com o objetivo de diminuir o número de funcionalidades apresentadas ao usuário. A Figura 4-9 apresenta o processo alterado.

Quando o usuário finaliza a análise do último artefato, a LPS pode ser considerada finalizada. São gravados três arquivos no computador do usuário, com informações sobre a LPS:

- um arquivo chamado “linha.xml” que contém todas as funcionalidades da LPS, com suas classificações;
- um arquivo chamado “SF.txt” que contém a lista de “*stop features*” identificadas;
- um arquivo chamado “linha.properties” que contém dois valores: o primeiro é o número total de funcionalidades da LPS, o segundo representa o menor percentual de funcionalidades comuns e variáveis encontrado durante a análise de todos os produtos da LPS.

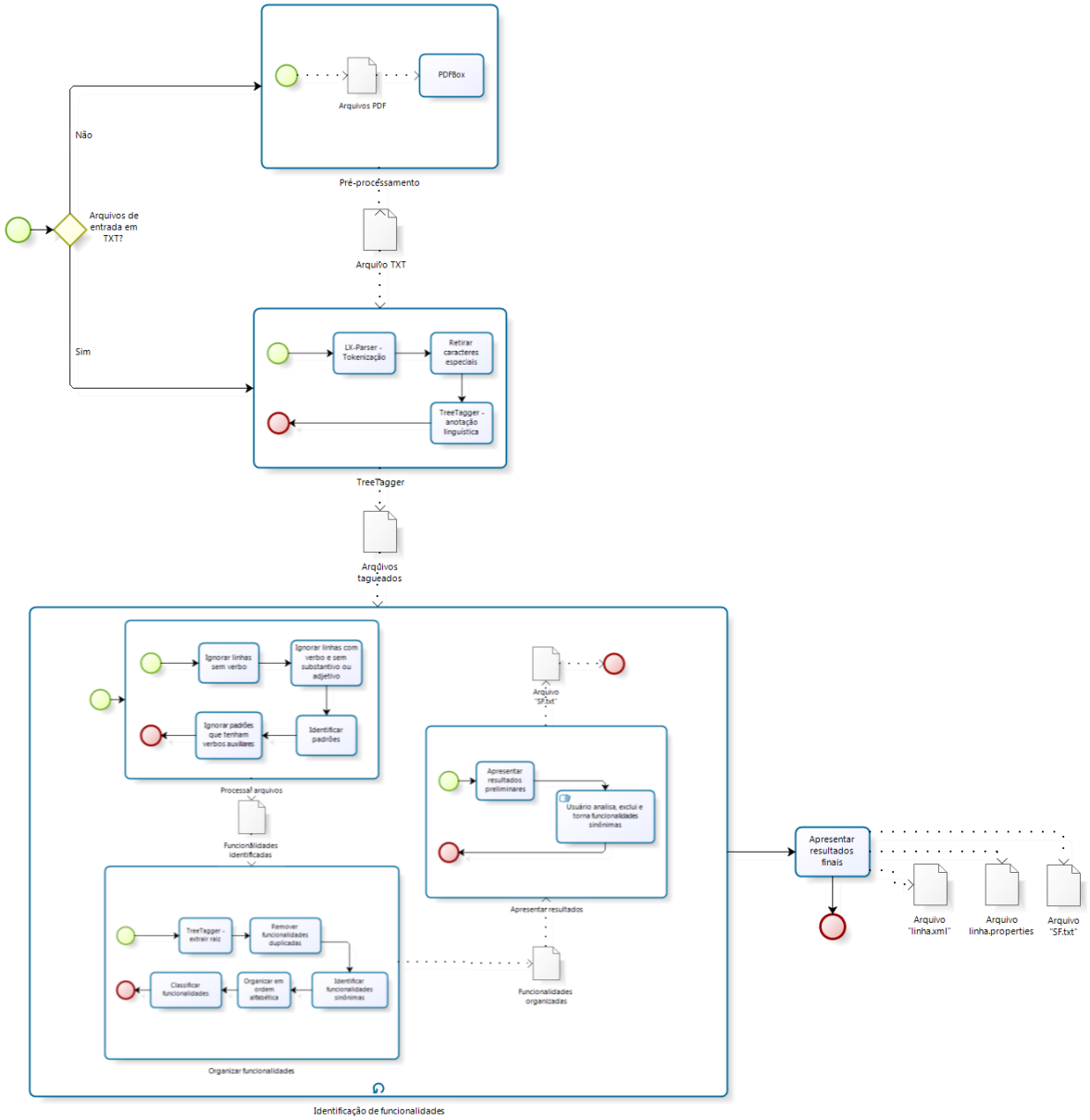

Figura 4-9. Processo alterado (Fonte: o Autor).

O arquivo “linha.xml” é o arquivo que representa a LPS criada. Por meio dele é possível visualizar todas as funcionalidades que compõe a LPS, em que produtos se encontram e suas classificações. Este arquivo também é utilizado para que seja possível avaliar um novo produto em relação à LPS existente.

O arquivo “SF.txt” e o arquivo “linha.properties” não estavam descritos na Figura 4-2 descrita anteriormente pois foram implementados apenas depois da realização do segundo experimento, que possibilitou a identificação da necessidade da criação das *stop features* e a identificação das informações adicionais que seriam necessárias para que fosse possível avaliar um novo produto em relação à uma LPS existente.

Depois da realização destes ajustes na implementação, foi realizado o terceiro experimento que teve como objetivo comparar os resultados da leitura e análise manual de 6 documentos (manuais de usuário) de uma mesma família de *Smartphones* da LG para criação de uma LPS, com o processamento dos mesmos manuais, porém, utilizando a abordagem proposta para criação da LPS.

Para este experimento os arquivos foram pré-processados manualmente para conter apenas os capítulos que descrevem funções dos aparelhos (capítulos como índice, garantias, endereços e acessórios foram excluídos) e também foram “limpos” de acordo com os itens informados anteriormente na Seção 4.3.

Todas as pessoas que participaram do segundo experimento foram convidadas a participar deste terceiro, porém, por ser um experimento que demandou um tempo razoável de dedicação para ser realizado, apenas duas conseguiram finalizá-lo. As duas pessoas que participaram são usuários de *smartphones* e possuem mestrado em andamento ou finalizado.

O experimento foi realizado em duas etapas. Na primeira, os 6 documentos foram lidos e as funcionalidades identificadas foram extraídas manualmente. No primeiro manual foram lidas 62 páginas, no segundo foram lidas 51 páginas, no terceiro foram lidas 48 páginas, no quarto 39 páginas, no quinto foram lidas 30 páginas e no sexto foram lidas 52 páginas. No total, 282 páginas foram lidas no experimento, por ambos participantes.

Ao mesmo tempo em que o documento era lido, as funcionalidades identificadas eram anotadas em outro documento. O tempo gasto para a leitura e extração das funcionalidades dos documentos também foi anotado. A Tabela 4-10

apresenta os resultados do Participante 1, enquanto a Tabela 4-11 apresenta os resultados do Participante 2. São apresentados os tempos gastos para análise de cada documento assim como o número de funcionalidades identificadas em cada documento:

Tabela 4-10. Resumo etapa manual – Participante 1 (Fonte: o Autor)

Nome do documento	Tempo gasto	Funcionalidades identificadas
LG Hotmail Phone C570	01:13:00	331
LG Optimus 2X P990	01:06:00	342
LG Optimus GT540	01:17:00	339
LG Optimus L3 E400	00:42:00	321
LG Optimus Pro C660	00:32:00	184
LG Optimus Black P970	00:41:00	279

O tempo total gasto pelo Participante 1 foi de 5 horas e 31 minutos, utilizado para identificar 1796 funcionalidades. Já o Participante 2 gastou 2 horas e 41 minutos para identificar 697 funcionalidades.

Tabela 4-11. Resumo etapa manual – Participante 2 (Fonte: o Autor)

Nome do documento	Tempo gasto	Funcionalidades identificadas
LG Hotmail Phone C570	00:54:00	177
LG Optimus 2X P990	00:22:00	103
LG Optimus GT540	00:17:00	94
LG Optimus L3 E400	00:23:00	114
LG Optimus Pro C660	00:24:00	103
LG Optimus Black P970	00:21:00	106

Analisando as tabelas, verifica-se que o Participante 1 identificou cerca de 60% a mais de funcionalidades que o Participante 2. Com relação ao tempo, o participante 1 gastou cerca de 50% a mais.

Para interpretar estes números, as funcionalidades identificadas manualmente pelos dois participantes foram analisadas e comparadas . Foi possível identificar que o Participante 1 foi bem mais detalhista que o Participante 2 ao descrever as funcionalidades, por isso encontrou maior número e gastou mais tempo. Enquanto o Participante 2 descrevia uma funcionalidade como “Configurar sons”, o Participante 1 descrevia em detalhes identificando as seguintes funcionalidades, como exemplo da análise do documento “LG Hotmail Phone C570”:

- selecionar o som de alerta que soará quando um novo e-mail for recebido;
- selecionar o som de alerta que soará quando uma nova mensagem SMS for recebida;

- selecionar o som de alerta que soará quando uma nova mensagem instantânea for recebida;
- selecionar o som de alerta que soará quando uma nova mensagem de canal for recebida;
- selecionar o som de alerta que soará quando uma nova mensagem de ferramenta SIM for recebida;
- selecionar o som de alerta que soará quando o aparelho recebe uma informação;
- selecionar o som de alerta que soará quando o aparelho solicitar ao usuário uma pergunta;
- selecionar o som de alerta que soará quando o aparelho exibir uma mensagem de aviso;
- definir o som das teclas para quando elas forem pressionadas;
- ajustar e personalizar os toques, sons de alerta e outros sons do aparelho;

Analisando as funcionalidades identificadas pelos participantes para o documento “LG Optimus 2X P990”, também foram identificados alguns problemas em algumas funcionalidades do Participante 1. Funcionalidades vagas, talvez escritas de forma incompleta ou identificadas incorretamente como “Voltar”, “Mostrar”, “Visualizá-las”, “Salvar” e “Compartilhar” estão na lista. Entretanto, a situação identificada no documento anterior se repete neste: enquanto o Participante 2 identificou as funcionalidades “Gravar voz” e “Enviar gravação de voz via bluetooth, email, gmail ou mensagens”, o Participante 1 identificou:

- gravador de voz para gravar notas de voz;
- iniciar a gravação;
- parar a gravação;
- escutar a gravação;
- escutar a gravação salva;
- enviando a gravação de voz;

As mesmas situações identificadas na análise dos dois primeiros documentos se repetem nos outros 4. Ou seja, o Participante 1 demonstrou ser mais detalhista ao descrever funcionalidades: ao invés de descrever “Configurar volume”, por exemplo, este participante descreveria todas as formas de realizar essa configuração como, por exemplo: “Aumentar volume da música”, “Diminuir volume da música”, “Aumentar volume do rádio” e etc.

Entretanto, em alguns momentos funcionalidades vagas foram descritas por este mesmo Participante, como “Voltar”, “Salvar” e “Compartilhar”. Estas funcionalidades, quando descritas em uma lista que não identifica, por exemplo, o capítulo no qual foram encontradas, não podem ser consideradas relevantes pois não estão claras: “Voltar” para onde? “Salvar” e “Compartilhar” o que?

Na segunda etapa do experimento, a abordagem proposta para criação da LPS foi utilizada. As pessoas foram instruídas na utilização da abordagem, com explicação sobre seu funcionamento e foram direcionadas a, primeiro, analisar as funcionalidades identificadas com o objetivo de excluir as que não eram funcionalidades realmente, e só depois identificar funcionalidades sinônimas.

O tempo gasto na segunda etapa do experimento também foi anotado pelos participantes. Esta anotação é relacionada ao tempo que a pessoa levou para analisar os resultados apresentados, excluir e criar funcionalidades sinônimas. O tempo que o algoritmo levou para extrair as informações dos documentos foi gravado pelo próprio algoritmo em sua log de execução. Também foi gravada na log de execução do algoritmo as funcionalidades excluídas pelo usuário e as funcionalidades que o algoritmo ignorou e não apresentou ao usuário porque estavam na lista de *stop features* pois já tinham sido excluídas anteriormente pelo próprio usuário. A Tabela 4-12 apresenta os resultados do Participante 1 e a Tabela 4-13 apresenta os resultados do Participante 2. Ambos apresentam o resumo desta segunda etapa.

Tabela 4-12. Resumo etapa automática – Participante 1 (Fonte: o Autor)

Ciclo	Funcionalidades ignoradas – *stop features*	Funcionalidades apresentadas	Tempo gasto pelo algoritmo	Funcionalidades excluídas pelo usuário
1	-	1349	00:00:60	79
2	2	2284	00:00:18	122
3	23	2813	00:00:16	40
4	24	3322	00:00:18	8
5	26	3876	00:00:18	1
6	32	4085	00:00:18	3

A coluna ciclo representa o ciclo de processamento do algoritmo. No ciclo 1 o primeiro arquivo é processado, portanto a lista de *stop features* está vazia e

nenhuma funcionalidade é ignorada automaticamente. As funcionalidades são apresentadas ao usuário (coluna 3), o usuário analisa e exclui as que deseja (última coluna). Quando o usuário finaliza a análise dos dados apresentados no primeiro ciclo, o algoritmo inicia o segundo ciclo, ou seja, processa o segundo arquivo, reúne os resultados com os resultados do primeiro arquivo, ignora automaticamente as funcionalidades que estão na lista de *stop features* e apresenta novamente o resultado para o usuário. Estes ciclos se repetem até que as informações de todos os arquivos sejam processadas, apresentadas e analisadas pelo usuário.

A primeira linha da coluna que representa o tempo de processamento gasto pelo algoritmo (coluna 4) também contempla o tempo que o algoritmo levou para realizar a anotação linguística em todos os documentos que serão analisados, por isso o primeiro ciclo é o mais demorado que os demais.

Tabela 4-13. Resumo etapa automática – Participante 2 (Fonte: o Autor)

Ciclo	Funcionalidades ignoradas – *stop features*	Funcionalidades apresentadas	Tempo gasto pelo algoritmo	Funcionalidades excluídas pelo usuário
1	-	1349	00:00:36	668
2	20	1629	00:00:08	623
3	139	1645	00:00:07	48
4	103	2177	00:00:06	845
5	202	1789	00:00:08	3
6	247	2031	00:00:05	628

O Participante 1 excluiu no total 253 funcionalidades. Todas elas foram incluídas na lista de *stop features*, resultando em 107 funcionalidades ignoradas automaticamente (não apresentadas ao usuário). Ao final dos 6 ciclos, o Participante 1 gerou uma LPS com 4065 funcionalidades, sem considerar as funcionalidades marcadas como sinônimas durante a análise feita pelo participante.

O segundo participante excluiu 2815 funcionalidades, resultando em 711 funcionalidades ignoradas automaticamente (não apresentadas ao usuário), por terem sido incluídas na lista de *stop features*. Cerca de 20% do número total de funcionalidades excluídas foram excluídas automaticamente sem a necessidade de avaliação do usuário.

Ao final dos 6 ciclos, o Participante 2 gerou uma LPS com 1051 funcionalidades, também desconsiderando as funcionalidades marcadas como sinônimas durante a análise feita pelo participante. Com relação aos ciclos, o ciclo 3 foi finalizado por engano pelo participante, por isso o número de funcionalidades excluídas neste ciclo é menor que dos anteriores e do imediatamente posterior. Já no ciclo 5, o participante resolveu finalizar o ciclo para avaliar de uma só vez o resultado de todos os arquivos, já que estava no penúltimo, por isso neste ciclo houveram apenas 3 funcionalidades excluídas.

Com relação ao tempo, o Participante 1 finalizou toda a análise em um dia, levando 06 horas. O Participante 2 levou 11 horas para finalizar as análises, porém esse tempo foi distribuído em 6 dias. A distribuição do tempo de análise em mais dias provavelmente beneficiou a análise do segundo participante, já que esta atividade é morosa, requer concentração e foco para bons resultados.

Analisando as funcionalidades que fazem parte da LPS dos participantes, criada nesta etapa do experimento, novamente identifica-se várias funcionalidades vagas ou incoerentes nos resultados do Participante 1. Exemplos: “permite acordo luz”, “usar acordo sua”, “reproduzir adic lista”, “picasa sns”, “dependendo software seu”, “arquivo srt mesmo”, “saber informações determinada”, etc. Como o Participante 1 utilizou cerca de 80% a menos de tempo em relação ao Participante 2, isto pode ter afetado a qualidade do resultado produzido, uma vez que esta tarefa é manual e requer atenção e concentração.

O segundo participante excluiu cerca de 1000% mais funcionalidades que o Participante 1, e levou cerca de 80% a mais de tempo. Como levou mais tempo para analisar os resultados, estima-se que a análise foi feita com mais profundidade e por isso mais funcionalidades foram excluídas e mais funcionalidades foram identificadas como sinônimas, diminuindo o resultado final do número de funcionalidades da LPS em cerca de 74%.

Comparando os resultados desta etapa com os resultados da etapa anterior, conclui-se que, nos dois casos, o Participante 1 identificou um número maior de funcionalidades. Porém, isto não significa que as funcionalidades identificadas estão corretas, pois várias inconsistências nos resultados foram encontradas posteriormente pelo pesquisador. Com relação ao tempo gasto, o Participante 1 gastou praticamente o mesmo tempo nas duas etapas. A Tabela 4-14 apresenta um resumo desta comparação.

Tabela 4-14. Comparação resultados etapas (Fonte: o Autor)

Participante	Tempo gasto etapa 1	Funcionalidades identificadas etapa 1	Tempo gasto etapa 2	Funcionalidades identificadas etapa 2
1	05:31:00	1796	06:00:00	4065
2	02:41:00	697	11:00:00	1051

É importante destacar que na primeira etapa apenas foram identificadas as funcionalidades, separadamente, para cada documento analisado. A análise de funcionalidades sinônimas e a análise de quais funcionalidades foram identificadas em vários documentos (para identificar as variabilidades da LPS) não foram realizadas. Portanto, o resultado final da primeira etapa não é o mesmo da segunda: na primeira etapa os participantes geraram uma lista de funcionalidades identificadas para cada documento analisado, ou seja, 6 listas separadas. Na segunda etapa, com a utilização da abordagem proposta, os participantes geraram uma LPS com suas funcionalidades classificadas como comum, variável ou opcional, conforme Seção 4.3 deste documento.

A Tabela 4-15 apresenta os dados da LPS criada pelo Participante 1 nesta etapa. Na tabela é possível verificar o total de funcionalidades encontradas em cada documento, separadas pelo seu tipo, e também a porcentagem em relação ao total de funcionalidades de cada documento.

Tabela 4-15. Dados da LPS criada – Participante 1 (Fonte: o Autor)

Nome do documento	Comum	Variável	Opcional
LG Hotmail Phone C570	20 (1,75%)	117 (10,22%)	1008 (88,03%)
LG Optimus 2X P990	20 (2,11%)	613 (64,8%)	313 (33,09%)
LG Optimus GT540	20 (1,97%)	431 (42,55%)	562 (55,48%)
LG Optimus L3 E400	20 (1,95%)	480 (46,74%)	527 (51,31%)
LG Optimus Pro C660	20 (2,68%)	517 (69,30%)	209 (28,02%)
LG Optimus Black P970	20 (2,03%)	576 (58,6%)	387 (39,37%)

Analisando a tabela, é possível identificar que em três documentos a maioria das funcionalidades identificadas é do tipo variável, nos outros três a maioria é do tipo opcional. É importante lembrar que as funcionalidades classificadas como

comuns estão presentes em todos os documentos, as classificadas como variáveis estão presentes em mais de um documento e as classificadas como opcionais estão apenas em um documento.

Portanto, pode-se considerar que as duas primeiras colunas representam funcionalidades que devem fazer parte da maioria das funcionalidades presentes nos produtos de uma LPS. Analisando novamente o quadro com esta perspectiva, verifica-se que três documentos teriam, nessa visão, mais funcionalidades do tipo opcional (que estão presentes apenas neles mesmos), do que comuns e variáveis (LG Hotmail Phone C570, LG Optimus GT540 e LG Optimus L3 E400).

É importante destacar que a correta classificação do tipo das funcionalidades também depende da ação do usuário ao analisar os resultados apresentados durante o processo, pois ao sinalizar que duas funcionalidades, classificadas como opcionais, são sinônimas, o algoritmo faz a criação de uma só funcionalidade e altera o tipo para variável (ou comum, se estivermos lidando com apenas dois documentos).

A Tabela 4-16 apresenta os mesmos dados da tabela anterior, porém com relação à LPS criada pelo Participante 2:

Tabela 4-16. Dados da LPS criada – Participante 2 (Fonte: o Autor)

Nome do documento	Comum	Variável	Opcional
LG Hotmail Phone C570	41 (9,47%)	104 (24,02%)	288 (66,51%)
LG Optimus 2X P990	41 (12,17%)	254 (75,37%)	42 (12,46%)
LG Optimus GT540	41 (12,09%)	192 (56,64%)	106 (31,27%)
LG Optimus L3 E400	41 (11,17%)	219 (59,67%)	107 (29,16%)
LG Optimus Pro C660	41 (14,14%)	202 (69,66%)	47 (16,2%)
LG Optimus Black P970	41 (12,69%)	239 (73,99%)	43 (13,32%)

Analisando os resultados da linha criada pelo Participante 2, verifica-se que apenas em um dos documentos a maioria das funcionalidades identificadas (66,51%) é do tipo opcional (que está presente apenas nele mesmo): o documento LG Hotmail Phone C570. Em todos os outros 5 documentos, a maioria das funcionalidades é do tipo variável, que são as que estão presentes em mais de um documento da LPS.

Nos dois resultados, o mesmo documento apresentou mais funcionalidades opcionais do que comuns e variáveis: “LG Hotmail Phone C570”. Na Figura 4-8 foram identificados 3 padrões diferentes na estrutura e no estilo de escrita dos documentos. Este documento, assim como o “LG Optimus GT540”, foram os únicos

que tinham um padrão totalmente diferente dos demais. Este padrão diferente pode ter influenciado a identificação das funcionalidades e originado estes resultados.

4.5 Fase 5 – Implementar a avaliação de um produto em relação a uma família de produtos pré-existente

Nesta fase foi implementado o algoritmo que avalia se um novo produto faz parte ou não de uma LPS já criada. Nesta fase são utilizados os três arquivos que são gravados no momento em que a criação da LPS é finalizada usando a abordagem proposta neste trabalho.

Para iniciar a avaliação, o usuário informa o diretório onde estão os três arquivos citados anteriormente. O algoritmo localiza o arquivo XML, faz sua leitura e apresenta a LPS ao usuário, como na Figura 4-10. Nesta tela, ao final da listagem de funcionalidades atuais, o usuário deve informar o diretório onde se encontra o documento do novo produto que será avaliado.

O mesmo processamento que é feito para criação de uma nova linha é feito neste momento para a avaliação de um produto novo: o arquivo é transformado para o formato TXT caso não esteja, recebe as anotações linguísticas, em seguida as funcionalidades são extraídas de acordo com os padrões pesquisados, são organizadas e apresentadas ao usuário para análise. São realizados praticamente todos os mesmos passos da Figura 4-9. Neste processo também é avaliada a lista de *stop features* da LPS existente, excluindo automaticamente funcionalidades do produto que estejam nesta lista.

Porém, depois que o usuário finaliza a análise do resultado das funcionalidades identificadas no novo produto, excluindo e tornando sinônimas as que deseja, o processo de avaliação propriamente dito é realizado. É feita uma busca nas funcionalidades da LPS existente, para verificar quais funcionalidades do novo produto já existem na LPS. Isto é realizado para identificar quais funcionalidades seriam consideradas comuns ou variáveis caso o produto fosse incorporado à LPS.

FUNCIONALIDADES IDENTIFICADAS

Substantivo	Funcionalidade raiz (Funcionalidade original)	Classificação	Produto
ABAR	deslizar abar notificação (deslize aba notificação)	VARIÁVEL	LG Optimus 2X P990 LG Optimus L3 E400 LG Optimus Pro C660
ACESSIBILIDADE	gerenciar acessibilidade (gerenciar acessibilidade)	OPCIONAL	LG Optimus GT540
ACESSO	configurar acesso fio (configurar acesso fio)	VARIÁVEL	LG Optimus 2X P990 LG Optimus Black P970 LG Optimus L3 E400
ACESSO	fornecer acesso remoto (fornece acesso remoto)	OPCIONAL	LG Hotmail Phone C570
ACESSO	renomeando acesso (renomeando acesso)	VARIÁVEL	LG Optimus 2X P990 LG Optimus Black P970 LG Optimus L3 E400

Figura 4-10. Lista das funcionalidades de uma LPS existente (Fonte: o Autor).

Em seguida, é calculada a porcentagem de funcionalidades que seriam classificadas como comuns ou variáveis caso o produto fosse incorporado à LPS, em relação ao total de funcionalidades do produto. Este é o mesmo cálculo feito pelo algoritmo para todos os produtos ao finalizar a criação de uma LPS e gravar o arquivo “SF.txt”.

Os valores do arquivo “SF.txt” da LPS existente são lidos e os percentuais são comparados. Se o percentual de funcionalidades comuns ou variáveis do produto novo for maior ou igual ao percentual de referência da LPS existente, o produto é considerado parte da LPS e é automaticamente incorporado a ela. Como resultado final, as funcionalidades são classificadas novamente e um novo arquivo “linha.xml” é gravado. A nova LPS é apresentada ao usuário na tela. Caso o produto não seja considerado parte da LPS existente, uma mensagem informativa é apresentada conforme Figura 4-11 e o processo é finalizado.

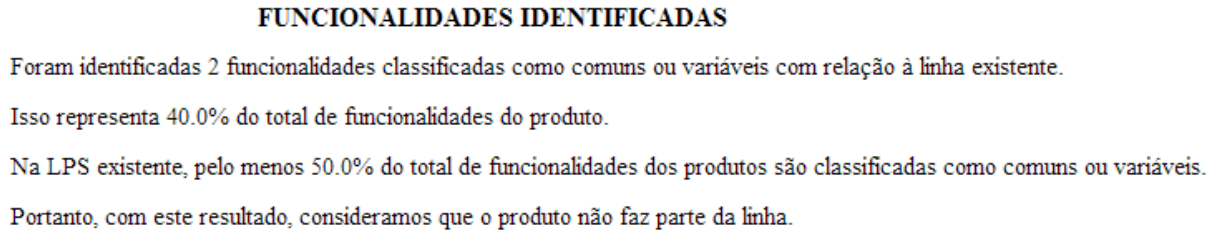

Figura 4-11. Mensagem informativa reprovando novo produto (Fonte: o Autor).

4.6 Fase 6 – Avaliar a classificação de um produto em relação à família pré-existente

Analisando os resultados apresentados na Fase 4 desta pesquisa, a LPS criada pelo Participante 2 foi a LPS selecionada para ser utilizada nesta fase como a família pré-existente, pois foi o resultado que demonstrou que esta LPS é a mais estável das duas criadas no experimento. Para essa LPS, de acordo com a Tabela 4-16, a porcentagem de referência para avaliação de um novo produto é 33,49%. Este valor é correspondente à soma da porcentagem de funcionalidades comuns e variáveis do produto “LG Hotmail Phone C570”, que é o menor valor encontrado nos produtos que compõem essa LPS.

Nesta Fase 3 os seguintes experimentos foram realizados:

- avaliação de um celular da família *Smartphones*, porém de outra marca;
- avaliação de um celular convencional da mesma marca dos utilizados na criação da LPS; e,
- avaliação de uma TV LCD.

Para estes experimentos, os arquivos foram pré-processados manualmente para conter apenas os capítulos que descrevem funções dos aparelhos (capítulos como índice, garantias, endereços, instalação e acessórios foram excluídos) e também foram “limpos” de acordo com os itens informados anteriormente:

- deve conter uma frase por linha (sem necessidade de incluir pontuação);
- não deve possuir palavras separadas incorretamente por espaços em branco ou hifenização;

Os experimentos foram realizados pelo próprio pesquisador, pois o objetivo neste caso é verificar se o comportamento do algoritmo que avalia se um produto faz parte ou não da LPS existente está correto, portanto, não gerando nenhum viés

indesejado na pesquisa. A Tabela 4-17 apresenta um resumo dos experimentos realizados.

Tabela 4-17. Resumo avaliação produto novo (Fonte: o Autor)

Documento	Funcionalidades ignoradas – *stop features*	Funcionalidades apresentadas	Funcionalidades restantes após análise	Funcionalidades comuns ou variáveis com relação à LPS
Motorola MB502	47	1128	314	30
TV LCD	9	328	110	2
LG-A180	20	234	81	6
LG-E405	387	673	344	145

A primeira linha da tabela é de um celular da família *smartphone* e a terceira é um celular da família de celulares convencionais. Para o *smartphone*, 30 funcionalidades identificadas também foram encontradas na LPS existente. Isso representa 9,55% do total de funcionalidades desse produto, portanto ele não foi considerado parte da LPS. Da mesma forma, os outros dois produtos também não foram considerados parte da LPS, pois tiveram poucas funcionalidades encontradas em comum.

Com este resultado, buscou-se mais um manual da família *smartphones* da marca LG, para validar o comportamento do algoritmo com um produto que deve ser considerado parte da LPS. Buscando na *web*, foi encontrado o manual do celular LG-E405 e um novo experimento foi feito. Da mesma forma que os experimentos anteriores, o arquivo foi “limpo” e pré-processado manualmente para conter apenas os capítulos que descrevem funções do aparelho.

No caso do manual do celular LG-405, 145 funcionalidades identificadas também foram encontradas na LPS existente. Isso representa 42,15% do total de funcionalidades desse produto, portanto ele foi considerado parte da LPS. A Figura 4-12 apresenta a tela que o usuário visualizou com as informações sobre a comparação realizada pelo algoritmo e com a lista das funcionalidades que compõe a LPS atualizada com o novo produto.

FUNCIONALIDADES IDENTIFICADAS

Foram identificadas 145 funcionalidades classificadas como comuns ou variáveis com relação à linha existente.

Isso representa 42.151162790697676% do total de funcionalidades do produto.

Na LPS existente, pelo menos 33.49% do total de funcionalidades dos produtos são classificadas como comuns ou variáveis.

Com este resultado, consideramos que o produto faz parte da linha.

As funcionalidades do produto novo foram adicionadas à linha.

Substantivo	Funcionalidade raiz (Funcionalidade original)	Classificação	Produto
ABAR	deslizar abar notificação (deslize aba notificação)	VARIÁVEL	LG Optimus 2X P990 LG Optimus L3 E400 LG Optimus Pro C660 LG-E405
ACESSIBILIDADE	gerenciar acessibilidade (gerenciar acessibilidade)	OPCIONAL	LG Optimus GT540

Figura 4-12. Mensagem informativa aprovando novo produto (Fonte: o Autor).

Na Tabela 4-17 pode-se verificar que para este produto que foi considerado parte da LPS existente, 387 funcionalidades foram ignoradas automaticamente (não apresentadas ao usuário), por estarem na lista de *stop features*. Isto mostra que cerca de 54% do número total de funcionalidades excluídas foram excluídas automaticamente sem a necessidade de avaliação do usuário. Isso diminui muito o tempo que o usuário gasta para analisar os resultados, realizar as exclusões e marcação de sinônimos. Neste caso foram gastos 29 minutos. Caso a lista de *stop features* não tivesse sido implementada, pode-se estimar que o tempo de análise manual aumentaria mais de 50%, cerca de 43 minutos.

CAPÍTULO 5 - DISCUSSÃO DOS RESULTADOS

"Tornou-se chocantemente óbvio que a nossa tecnologia excedeu a nossa humanidade."
- Albert Einstein

Este capítulo tem o objetivo de discutir detalhadamente os resultados da pesquisa apresentados no capítulo anterior.

5.1 Reflexões acerca dos resultados obtidos

No Capítulo 1, o objetivo geral da pesquisa foi definido como: "desenvolver uma abordagem semiautomática para auxiliar a definição de escopo de LPS" e os seguintes objetivos específicos foram definidos para atender o objetivo geral:

1- Classificar automaticamente funcionalidades de sistemas existentes;
2- Definir o escopo da LPS por meio da análise manual das funcionalidades classificadas automaticamente;
3- Avaliar automaticamente novos produtos em relação ao escopo definido da LPS;
4- Avaliar a abordagem por meio da comparação com a definição de escopo de forma manual.

Para atender aos objetivos, a pesquisa foi dividida em 6 fases, conforme Capítulo 3, Figura 3-1. Para atender ao primeiro objetivo específico, foram realizadas as Fases 1, 2 e 3. Para atender ao segundo e ao quarto objetivos específicos, foi realizada a Fase 4. Para atender o terceiro objetivo específico, foram realizadas as Fases 5 e 6.

5.1.1 Classificação automática das funcionalidades

Para atender este objetivo específico, as seguintes fases foram realizadas:

- fase 1 – Identificar quais artefatos dos sistemas de software existentes serão analisados;
- fase 2 – Identificar a técnica que será utilizada para identificação e classificação automática das funcionalidades;

- fase 3 – Implementar a identificação e classificação automática das funcionalidades;

As Fases 1 e 2 foram executadas para que a Fase 3 pudesse ser efetivamente realizada. Esta gerou os resultados que atenderam este objetivo e estão descritos no Capítulo 4.

Para atender este objetivo, ou seja, classificar automaticamente funcionalidades de sistemas existentes, um algoritmo foi implementado. A avaliação do algoritmo foi realizada na Fase 4.

5.1.2 Definir o escopo da LPS e avaliá-lo

Para atender este objetivo específico, a seguinte fase foi realizada:

- fase 4 - Avaliar a identificação e classificação automática das funcionalidades;

Nesta fase foram realizados três experimentos com o objetivo de avaliar a identificação e classificação automática das funcionalidades, comparando seus resultados com a identificação feita totalmente de forma manual. Todos os resultados estão descritos no Capítulo 4.

Os resultados do **pré-experimento** mostraram que o algoritmo implementado para identificar automaticamente as funcionalidades teve um desempenho melhor que a identificação totalmente manual. Concluiu-se que o tempo gasto para identificação das funcionalidades diminuiu mais de 40%, enquanto o número de funcionalidades relevantes encontradas aumentou mais de 48%.

Neste experimento, algumas funcionalidades encontradas manualmente não foram encontradas automaticamente. Conforme descrito no Capítulo 4, Quadro 4-2, elas não foram encontradas porque o estilo de escrita das frases não se encaixou em nenhum dos padrões procurados pelo algoritmo.

Apesar do resultado final deste experimento ter sido positivo, conclui-se que é possível melhorar ainda mais a performance do algoritmo revisando os padrões e também revisando a forma pela qual o algoritmo busca os padrões.

Por exemplo, a funcionalidade “Enviar contatos via bluetooth” não foi encontrada neste experimento. De acordo com as informações do Quadro 4-2, a frase onde esta funcionalidade poderia ter sido encontrada, com suas anotações linguísticas, é a seguinte: “(V enviar)(ADJ todos)(DET os)(NOM contatos)(PRP por)(NOM bluetooth).”

O padrão da frase, identificado pelo algoritmo é VERBO+ADJETIVO+SUBSTANTIVO+SUBSTANTIVO (apenas verbos, adjetivos e substantivos são levados em consideração). Relembrando, os padrões pesquisados pelo algoritmo são: VERBO+SUBSTANTIVO+ADJETIVO, VERBO+ADJETIVO+SUBSTANTIVO, VERBO+SUBSTANTIVO+SUBSTANTIVO OU VERBO+SUBSTANTIVO.

Os padrões são buscados pelo algoritmo na ordem em que estão descritos no parágrafo acima. Logo que um padrão é identificado, o padrão da frase é atualizado considerando as palavras que sobraram. Ou seja, nesta frase foi identificado o segundo padrão, gerando a funcionalidade “enviar todos contatos” e deixando na frase apenas o último substantivo, que é a palavra *bluetooth*.

O resultado deste experimento sugere que se pode revisar esta forma de buscar os padrões para que mais funcionalidades possam ser identificadas na mesma frase. No exemplo citado, caso o terceiro padrão tivesse sido identificado, a funcionalidade “enviar contatos *bluetooth*” seria gerada.

O algoritmo poderia ser alterado para não descartar as palavras que formaram um dos padrões procurados na frase, e buscar todas as combinações possíveis entre o verbo da palavra e os adjetivos e substantivos existentes. Com essa alteração, para o exemplo anterior, seriam identificadas as funcionalidades "enviar todos contatos" (segundo padrão), "enviar contatos bluetooth" (terceiro padrão), "enviar contatos" (quarto padrão) e "enviar bluetooth" (quarto padrão).Porém, com essa alteração no algoritmo o número de funcionalidades identificadas aumentaria consideravelmente. No exemplo anterior, aumentou de 1 para 4 funcionalidades. A primeira e a terceira (“enviar todos contatos” e “enviar contatos”) provavelmente seriam marcadas por um usuário (de forma manual) como sinônimas. A última provavelmente seria descartada, pois não deixa claro o que será enviado por *bluetooth*.

Portanto, esta possível alteração ajudaria a aumentar o número de funcionalidades relevantes encontradas, porém, da mesma forma, aumentaria o número de funcionalidades irrelevantes. Consequentemente, o tempo de análise manual dos resultados seria maior.

Além dessa alteração, poderia ser feita uma análise nos padrões para verificar a porcentagem de funcionalidades relevantes que foram identificadas

através deles. Esta análise auxiliaria a identificar se algum dos padrões poderia ser excluído, reduzindo o número de funcionalidades irrelevantes identificadas.

O **segundo experimento** realizado foi parecido com o anterior, porém realizado por mais pessoas, sem a presença do pesquisador. Os resultados também estão descritos no Capítulo 4, e mostram que, na média, foram identificadas 124,77% a mais de funcionalidades utilizando a abordagem proposta, e gasto 24,53% a menos de tempo, em relação à abordagem manual. Também concluiu-se que a representatividade do número de funcionalidades encontradas automaticamente, em relação ao total de funcionalidades relevantes (somatório das funcionalidades encontradas manualmente e automaticamente) é grande: varia de 88,41% à 96,85%.

Neste experimento, as funcionalidades que não foram identificadas pelo processamento automático também foram analisadas e listadas no Quadro 4-9. Nesta análise, mais um motivo, além dos padrões, foi identificado: em alguns casos a palavra foi anotada incorretamente pelo algoritmo utilizado nesta abordagem. Por exemplo: a palavra “chamada” em determinada frase foi anotada como adjetivo e não substantivo, fazendo com que o padrão pesquisado não fosse encontrado.

Para melhorar os resultados, poderia ser feita uma análise do benefício da inclusão de novos padrões. Por exemplo, o padrão VERBO+ADJETIVO não existe mas poderia ser incrementado no algoritmo. Esse padrão poderia identificar funcionalidades que tiveram palavras anotadas incorretamente pelo tagueador, como o exemplo do id 1 do Quadro 4-9.

Outra alteração que poderia ser avaliada é a inclusão de padrões que levem em consideração outras tags do TreeTagger (Quadro 4-1). Caso existisse um padrão, por exemplo, com a tag de palavra estrangeira, a funcionalidade do id 3 do Quadro 4-9 poderia ser encontrada.

Além disso, pode-se avaliar a possibilidade de trocar o algoritmo que é utilizado para realizar a anotação linguística, com o objetivo de melhorar ainda mais os resultados. Para isso seria necessário pesquisar um novo algoritmo de anotação linguística e realizar testes para garantir a melhoria nos resultados.

Além do problema com a anotação de algumas palavras, 3 funcionalidades foram encontradas apenas no processo manual porque foram geradas a partir da interpretação da leitura de alguns parágrafos do manual analisado. Neste caso não foi identificada nenhuma alteração que possa ser feita no algoritmo, pois se trata do

resultado da leitura e interpretação de um humano. Portanto esta é uma limitação da abordagem proposta. O resultado desse segundo experimento também gerou melhorias no processo a ser seguido pelo usuário, apresentando os resultados gradativamente à medida que os manuais são processados, e não tudo de uma só vez. Além disso, implementou-se uma lista de "*stop features*", populada cada vez que o usuário exclui funcionalidades irrelevantes durante o processo já existente e que é utilizada para ignorar automaticamente funcionalidades já excluídas quando da análise de manuais anteriores. Essas melhorias foram implementadas e o terceiro experimento foi realizado.

O **terceiro experimento** foi maior que os anteriores, e também realizado semiautomaticamente por meio da abordagem proposta, e manualmente, por especialistas do domínio. Neste experimento, seis manuais completos foram analisados. Todos os resultados estão descritos no Capítulo 4.

Os resultados deste experimento confirmaram os anteriores. O número de funcionalidades identificadas foi maior, e com relação ao tempo gasto manualmente e automaticamente, para um participante do experimento foi praticamente o mesmo, para o outro, o tempo gasto no processamento automático foi mais de 300% maior.

Entretanto, é importante destacar que, no processamento manual, apenas foram identificadas as funcionalidades. A análise de funcionalidades sinônimas e a identificação das variabilidades, para criação de uma LPS, não foi realizada no processamento manual. Com a utilização da abordagem proposta, os participantes deste experimento geraram uma LPS com suas funcionalidades classificadas como comum, variável ou opcional.

Sobre a implementação das *stop features*, para um dos participantes deste experimento a redução do número de funcionalidades chegou a mais de 20%, diminuindo ainda mais o tempo gasto na análise das funcionalidades. Conclui-se, portanto, que a implementação das *stop features* foi benéfica à abordagem proposta.

Com relação à correta classificação automática das funcionalidades em comum, variável ou opcional, alguns problemas no algoritmo foram identificados, quando relacionados às funcionalidades marcadas como sinônimas. Eles estão detalhados no Capítulo 6 deste documento como trabalhos futuros.

Além desses problemas, a correta classificação das funcionalidades também depende da identificação de funcionalidades sinônimas que o usuário pode fazer durante a análise dos resultados apresentados. Quando o usuário identifica que

duas funcionalidades são sinônimas, o algoritmo reúne-as em uma só e atualiza sua classificação na LPS.

5.1.3 Avaliar automaticamente novos produtos

Para atender este objetivo específico, as seguintes fases foram realizadas:

- fase 5 – Implementar a avaliação de um produto em relação à uma família de produtos pré-existente;
- fase 6 – Avaliar a classificação de um produto em relação à família pré-existente;

Na Fase 5 o algoritmo para avaliar se um novo produto faz parte ou não de uma LPS pré-existente foi implementado. Os detalhes da implementação estão descritos no Capítulo 4. Na Fase 6, uma LPS criada durante a Fase 4 foi selecionada para ser utilizada como família pré-existente e alguns experimentos foram realizados com o objetivo de avaliar se a classificação de um produto em relação à família pré-existente estava correta, atendendo a este objetivo. Os resultados estão detalhados no Capítulo 4.

Quatro experimentos foram realizados e os resultados mostram que a avaliação feita pelo algoritmo está correta, atingindo o objetivo específico. Nos três primeiros experimentos, o algoritmo indicou de forma correta que os produtos não faziam parte da LPS criada. No último, a indicação também foi correta: o produto foi indicado como parte da LPS.

Neste experimento, confirmou-se mais uma vez a efetividade da utilização das *stop features*: em um dos produtos analisados, cerca de 58% do número total de funcionalidades excluídas foram excluídas automaticamente sem a necessidade de avaliação do usuário.

5.1.4 Objetivo geral

Com os resultados apresentados, todos os objetivos específicos foram atingidos, portanto conclui-se que o objetivo geral também foi atingido: “desenvolver uma abordagem semiautomática para auxiliar na definição de escopo de LPS”. A abordagem foi desenvolvida e mostrou-se que auxilia no processo de identificação de funcionalidades para definição de escopo de uma LPS.

Desta forma, pode-se considerar também que a questão principal da pesquisa, “É possível semiautomatizar a identificação e classificação das funcionalidades de sistemas existentes - diminuindo a necessidade da presença constante do especialista de domínio durante a definição de escopo realizada - ao se criar uma LPS de maneira extrativa?”, foi respondida positivamente.

Os resultados mostraram que a abordagem semiautomática proposta neste trabalho auxiliou na identificação e classificação das funcionalidades. Os resultados das comparações feitas entre a utilização da abordagem e a identificação manual mostram que o número de funcionalidades identificadas utilizando a abordagem foi maior. Além disso, a abordagem não só identifica funcionalidades nos produtos como as classifica. Em todas as comparações, foi realizada manualmente apenas a identificação das funcionalidades, e não sua classificação.

Portanto, pode-se afirmar que a identificação e classificação de funcionalidades foi semiautomatizada com sucesso, e contribuiu para diminuir o tempo gasto com análises totalmente manuais. Além disso, o número de funcionalidades identificadas foi sempre maior, resultando na definição mais completa do escopo de um LPS se comparado com o escopo que seria definido de forma totalmente manual.

5.2 Validade e confiabilidade da pesquisa

Todos os experimentos foram realizados com usuários do produto escolhido, não com especialistas de domínio. Em todos os casos os resultados que utilizaram a abordagem proposta foram melhores que os que utilizaram a abordagem manual. Espera-se que, no caso de um especialista de domínio, os resultados sejam ainda melhores, pois a análise dos resultados deve, neste caso, ser feita com mais agilidade e rapidez.

Ainda, os resultados mostram que mesmo pessoas leigas no assunto de LPS conseguiram utilizar a abordagem proposta e gerar resultados positivos, quando comparados à abordagem totalmente manual.

A abordagem proposta neste trabalho procurou semi-automatizar a identificação e classificação de funcionalidades, porém, ainda possui etapas manuais que dependem dos usuários, pois os resultados precisam ser validados por

um usuário humano. Entretanto, os resultados mostraram que o tempo gasto pelos usuários nessas atividades manuais diminuiu.

O processo proposto sofreu alterações durante a realização dos experimentos, para facilitar a utilização dos usuários. Em sua primeira concepção, o usuário precisava analisar todos os resultados de uma só vez e se ocorresse algum problema, como falta de energia elétrica, por exemplo, todo o trabalho seria perdido. O processo foi alterado para mostrar os resultados de forma gradativa e salvar os resultados em arquivo, possibilitando a continuação da análise posteriormente. Com isso, a análise dos resultados pôde ser realizada com mais tranquilidade pelos participantes dos experimentos.

Além do quesito usuários, a qualidade da recuperação das funcionalidades na abordagem proposta está diretamente relacionada à qualidade dos arquivos processados para que os padrões sejam identificados. Por este motivo, em todos os experimentos os arquivos foram pré-processados manualmente para que apenas os capítulos que continham funcionalidades fossem analisados (capítulos como índice, acessórios e garantias foram excluídos). Além disso, também foram “limpos” de acordo com os critérios informados anteriormente no capítulo 4.

Essa necessidade de “limpar” os arquivos foi identificada porque todos os experimentos foram realizados a partir de manuais em PDF, transformados em formato TXT por meio de um algoritmo existente. No capítulo 3, na figura 3-2 foi demonstrado um exemplo dos problemas que podem ocorrer nessa conversão.

Esse processo de “limpeza” dos arquivos TXT também demanda tempo porém ele não foi descrito nos resultados, pois caso os manuais originais já estivessem nesse formato, este processo não seria necessário. Caso os manuais estejam em formato DOC, por exemplo, é possível salvar o documento em formato TXT sem causar os problemas identificados na conversão do PDF.

Com relação à porcentagem de referência utilizada para avaliar um novo produto em relação à uma LPS existente, levou-se em consideração que essa porcentagem poderia ser retirada da LPS criada por meio desta abordagem. Essa porcentagem utilizada é retirada de um produto que já faz parte desta LPS, portanto é uma referência válida.

CAPÍTULO 6 - CONSIDERAÇÕES FINAIS

Neste capítulo serão descritas as considerações finais deste trabalho com relação à relevância do estudo, contribuições da pesquisa, limitações da pesquisa e trabalhos futuros.

6.1 Relevância do estudo

Todas as abordagens de definição de escopo identificadas no Capítulo 2 deste trabalho necessitam da presença do especialista do domínio no qual será criada a LPS, porém apenas o trabalho proposto em (JOHN, 2010) procurou minimizar o tempo gasto por este profissional.

No trabalho proposto em (JOHN, 2010), a análise dos documentos pode ser realizada por outros profissionais e não necessariamente especialistas no domínio em questão, porém o especialista precisa validar esta análise executada, e como ela é totalmente manual, está sujeita a erros humanos e retrabalho.

Um dos trabalhos futuros citados em (JOHN, 2010) é a automação dos padrões para prover análise automática de documentos e identificação de artefatos para a LPS, pois a tarefa de aplicar manualmente padrões em documentos grandes é uma tarefa tediosa.

A abordagem proposta neste trabalho mostrou que é possível minimizar ainda mais o tempo gasto, pois mostrou ser possível recuperar as funcionalidades a partir de documentos, como por exemplo, manuais de usuário, diminuindo o tempo gasto na análise de documentos.

6.2 Contribuições da pesquisa

As principais contribuições da abordagem proposta se resumem em:

- auxiliar empresas que desejam migrar para a abordagem de linhas de produto de software a iniciar o mapeamento de seus produtos, e visualizá-los como famílias de um mesmo domínio que compartilham componentes vistos como funcionalidades em comum. A partir desta visualização é

possível iniciar o planejamento da arquitetura da linha, e identificar o reuso proporcionado pela linha.

- proporcionar uma maneira única de apresentar a família de produtos de uma forma visível a humanos e máquinas, possibilitando a realização de processamentos automáticos.
- auxiliar a tomada de decisão quanto à inserção ou não de um novo produto na família existente, mapeando suas funcionalidades em relação às funcionalidades disponíveis para serem reutilizadas na família. A partir da visualização das comunalidades e variabilidades existentes entre o futuro novo membro e a família existente, é possível decidir alterar funcionalidades da família para atender ao novo membro, incluí-las ou decidir implementá-las de maneira exclusiva ao novo produto.

6.3 Limitações da pesquisa

A principal limitação desta pesquisa está relacionada ao fato dos experimentos terem sido realizados com usuários do produto selecionado e não com especialistas do domínio. O ideal é que os experimentos tivessem sido realizados por especialistas de domínio, e comparados com uma LPS criada por eles de forma manual e extrativa.

Outra limitação da pesquisa é que os resultados dependeram da avaliação de humanos. Podem ter ocorrido erros durante as análises feitas nos experimentos e isso pode ter alterado os resultados, pois todos os experimentos feitos foram trabalhosos.

Esta abordagem prevê a utilização de manuais do produto para servirem como base, tanto para a construção da LPS quanto para a identificação de um novo produto pertencente à LPS. Isto pode ser uma limitação em casos de produto de software que não possuam manual do usuário. Uma possível abordagem, neste caso, seria partir dos documentos de Casos de Uso, que são artefatos razoavelmente comuns de se encontrar no desenvolvimento de software.

6.4 Trabalhos futuros

Diversas melhorias e correções no algoritmo implementado foram identificadas durante a realização dos experimentos e estão listadas a seguir. As

alterações propostas visam melhorar ainda mais os resultados obtidos, e também ampliar a utilização da abordagem proposta:

- revisar os padrões e também a forma pela qual o algoritmo trata as frases após a identificação de um padrão;
- analisar novas possibilidades de algoritmos existentes que realizam a anotação linguística com o objetivo de diminuir os erros e melhorar os resultados;
- corrigir o comportamento do algoritmo quando avalia funcionalidades sinônimas. Problemas encontrados:
    - funcionalidades sinônimas não são levadas em consideração ao classificar as funcionalidades, quando o resultado já analisado pelo usuário está sendo unido ao resultado do próximo produto. Por exemplo, a Figura 6-1 mostra uma funcionalidade principal “iniciar apresentação slide”, que possui 2 funcionalidades sinônimas: “visualizar apresentação de slide” e “ver apresentação e slide”. Na última linha é possível ver, novamente, a funcionalidade “ver apresentação de slide”, porém classificada incorretamente como opcional, presente em um único produto.

APRESENTAÇÃO	iniciar apresentação slide (iniciar apresentação slides) Funcionalidades sinônimas: 1. visualizar apresentação slide (visualiza apresentações slide) 2. ver apresentação slide (ver apresentação slides)	VARIÁVEL
APRESENTAÇÃO	parar apresentação slide (parar apresentação slides)	OPCIONAL
APRESENTAÇÃO	selecionar apresentação (selecione apresentação)	OPCIONAL
APRESENTAÇÃO	ver apresentação slide (ver apresentação slides)	OPCIONAL

Figura 6-1. Problema na classificação automática (Fonte: o Autor).

O algoritmo deveria ter identificado e unido as funcionalidades automaticamente, classificando a funcionalidade principal como

comum. Neste caso se o usuário não visualiza esse problema e não marca as duas como sinônimas, o resultado fica incorreto. Além disso, este problema aumenta o tempo de análise dos resultados, e também ocorre no processo de avaliação de um produto novo;

- reavaliar a implementação da identificação de funcionalidades sinônimas por meio da lista de verbos sinônimos. Poucos casos foram vistos durante os experimentos realizados. A Figura 6-2 é um exemplo que um caso onde duas funcionalidades foram marcadas como sinônimas automaticamente;

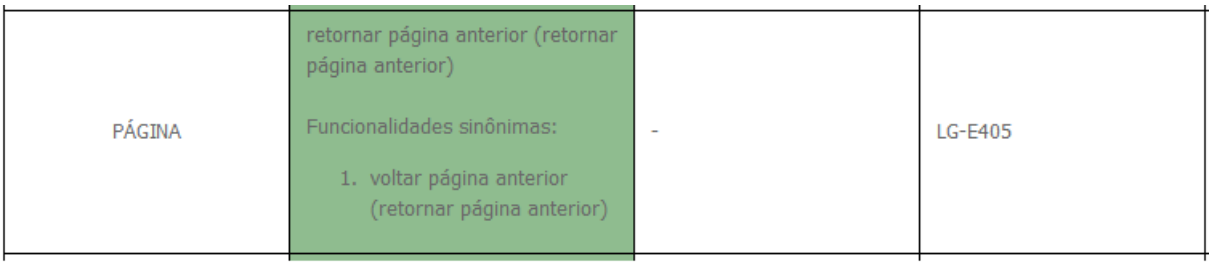

Figura 6-2. Sinônimas identificadas automaticamente (Fonte: o Autor).

- avaliar a possibilidade de permitir que funcionalidades relacionadas à termos diferentes sejam marcadas como sinônimas. Por exemplo, na implementação atual, as funcionalidades “iniciar cronometragem” e “iniciar cronômetro” não podem ser marcadas como sinônimas;

- melhorar o design de iteração com o usuário;
- avaliar a possibilidade de apresentar as funcionalidades informando os capítulos ou seções nos quais foram identificadas. Isso facilitaria a correta interpretação de funcionalidades como “inserir assunto”. Pode-se considerar que está relacionada ao envio de mensagem SMS, ou envio de email, ou criação de um evento no calendário, etc.
- implementar a atualização do arquivo de *stop functions* e o arquivo *properties* da LPS existente quando da inclusão de um novo produto;
- ajustar o processo de validação de novo produto para avaliar mais de um produto por vez;
- incluir nova funcionalidade que permita ao usuário editar uma LPS criada por meio da abordagem proposta;

- rever a abordagem para utilizar outros tipos de documento como base, como, por exemplo, documentos de Caso de Uso.

REFERÊNCIAS BIBLIOGRÁFICAS

(ÁLVARES, 2007) ÁLVARES, A. C. **Extração de informação de artigos científicos: uma abordagem baseada em indução de regras de etiquetagem**. Dissertação (Mestrado) – Universidade de São Paulo, São Paulo, 2007. 131 f.

(ALVES et al., 2010) ALVES, V.; NIU, N.; ALVES, C.; VALENÇA, G. **Requirements Engineering for Software Product Lines: A Systematic Literature Review.** Information and Software Technology, v. 52, agosto 2010.

(APACHE, 2012) The APACHE Software Foundation. **Apache PDFBox – Java PDF Library.** Disponível em: <http://pdfbox.apache.org/>. Acesso em 01. Mar. 2012.

(ARCHER et al., 2012) ARCHER, M.; CLEVE, A.; PERROUIN, G.; HEYMANS, P.; VANBENEDEN, C.; COLLET, P.; LAHIRE, P. **On extracting feature models from product descriptions.** In Proceedings of the Sixth International Workshop on Variability Modeling of Software-Intensive Systems (VaMoS '12). ACM, New York, NY, USA, pp. 45-54.

(BAEZA-YATES; RIBEIRO-NETO, 1999) BAEZA-YATES, R.; RIBEIRO-NETO, B. **Modern Information Retrieval**. In: ACM Press, Addison-Wesley, Wokingham, UK, 1999.

(BASILI; ROMBACH, 1991) BASILI, V. R.; ROMBACH, H. D. **Support for Comprehensive Reuse.** Software Engineering Journal, v. 6, n. 5, p. 306-316, março 1991.

(BOSCH, 2005) BOSCH, J. **Staged adoption of software product families**. Software Process: Improvement and Practice, Volume 10, Issue 2, pp. 125-142, Date: April/June 2005.

(CALADO, 2004) CALADO, P. P. **Utilização da Estrutura de Ligações da Web em Problemas de Recuperação de Informação**. Tese (Doutorado) – Universidade Federal de Minas Gerais, Minas Gerais, 2004. 148 f.

(CARBON et al., 2008) CARBON, R.; KNODEL, J.; MUTHIG, D.; MEIER, G. **Providing Feedback from Application to Family Engineering - The Product Line Planning Game at the Testo AG.** In: 12th International Software Product Line Conference. **Anais…** Limerick, Ireland: Ieee, 2008, p. 180-189.

(CLEMENTS, 2002) CLEMENTS, P; NORTHROP, L. **Software Product Lines: Practices and Patterns**. Boston: Addison-Wesley, 2002, 563 p.

(DeBAUD, 1999) DeBAUD, J; SCHMID, K. **A Systematic Approach to Derive the Scope of Software Product Lines**. In: Proceedings of the 21st International Conference on Software Engineering (ICSE). **Anais…** Los Angeles, CA, May 16-22, 1999. Los Alamitos, CA: IEEE Computer Society, 1999, p. 34-43.

(DUSZYNSKI, 2011) DUSZYNSKI, S. **A scalable goal-oriented approach to software variability recovery**. In: Software Product Lines - 15th International Conference, SPLC 2011, Munich, Germany, August 22-26, 2011.

(FERNEDA, 2003) FERNEDA, E. **Recuperação de Informação: análise sobre a contribuição da Ciência da Computação para a Ciência da Informação**. Tese (Doutorado) – Universidade de São Paulo, São Paulo, 2003. 147 f.

(GANESAN et al., 2006) GANESAN, D.; MUTHIG, D.; KNODEL, J.; ROSE, D. **Discovering Organizational Aspects from the Source Code History Log during the Product Line Planning Phase – A Case Study**. IEEE International Working Conference on Reverse Engineering (WCRE 2006), Villa dei Papi: 2006, p. 211 -220.

(GIMENES; TRAVASSOS, 2002) GIMENES, I. M., TRAVASSOS, G. H. **O enfoque de Linha de Produto para Desenvolvimento de Software**. In: XXI JAI - Livro Texto ed.Florianopolis : Sociedade Brasileira de Computação, 2002.

(GONZALEZ; LIMA, 2003) GONZALEZ, M.; LIMA, V. L. S. **Recuperação de Informação e Processamento da Linguagem Natural**. In: XXIII Congresso da

Sociedade Brasileira de Computação. Anais da III Jornada de Mini-Cursos de Inteligência Artificial.Campinas. v. III. SP. Campinas, 2003. p. 347-395.

(GOTTSCHALG-DUQUE, 2005) GOTTSCHALG-DUQUE, C. **SiSILiCO Uma Proposta para um Sistema de Recuperação de Informação baseado em Teorias da Linguística Computacional e Ontologia.** Tese (Doutorado) – Universidade Federal de Minas Gerais, Minas Gerais, 2005. 120 f.

(IANZEN et al., 2012) IANZEN, A.; MALUCELLI, A; REINEHR, S. **Definição de Escopo em Linhas de Produto de Software: uma abordagem semiautomática por meio de anotação lingüística.** In: Conferência Latino Americana de Informática (CLEI 2012). Medellín, Colômbia, October 1-5, 2012. In press.

(JACOBSON et al., 1997) JACOBSON, I.; GRISS, M.; JONSSON, P. **Software Reuse: Architecture, Process and Organization for Business Success.** New York: Addison Wesley, 1997. 497 p.

(JOHN, 2006) JOHN, I. **Capturing Product Line Information from Legacy User Documentation.** In: Software Product Lines. Research Issues in Engineering and Management. Springer, 2006.

(JOHN et al., 2006) JOHN, I.; KNODEL, J.; LEHNER, T.; MUTHIG, D. **A Practical Guide to Product Line Scoping**. In: Software Product Lines: Proceedings of the 10th International Software Product Line Conference (SPLC 2006). **Anais…** Baltimore, Maryland, August 21-24, 2006.

(JOHN, 2009) JOHN, I.; EISENBARTH, M. **A Decade of Scoping: A Survey**. In: Proceedings of the 13th International Software Product Line Conference, 1., 2009, Airport Marriott, San Francisco, CA, USA. **Anais…** Pittsburgh, 2009, p. 31-40.

(JOHN, 2010) JOHN, I. **Using Documentation for Product Line Scoping**. *IEEE Software*, vol. 27, 2010, p. 42 - 47.

(KRUEGER, 1992) KRUEGER, C.W. **Software Reuse**, In: ACM Computing Surveys, Vol. 24, No. 02, June, 1992.

(LEE et al., 2000) LEE, J.; KANG, S.; LEE, D. H. **A Comparison of Software Product Line Scoping Approaches.** In: International Journal of Software Engineering and Knowledge Engineering, Vol. 20, Issue 5, World Scientific, October 2010. pp. 637-663.

(LINDEN et al., 2007) LINDEN, F.; SCHIMID, K.; ROMMES, E. **Software Product Lines in Action.** Springer, 2007.

(LIU, 2010) LIU, Y.; NGUYEN, K.; WITTEN, M.; REED, K. **Cross Product Line Reuse in Component-based Software Engineering.** In: 2010 International Conference on Computer Application and System Modeling (ICCASM 2010). **Anais…** Taiyuan, China: 2010, p. 427-434.

(MANNING et al., 2009) MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. **An Introduction to Information Retrieval**. Cambridge University Press, 2009.

(MULLER, 2011) MULLER, J. **Value-Based Portfolio Optimization for Software Product Lines**. In: Software Product Line Conference (SPLC), 2011 15th International, pp.15-24, 22-26.

(NOOR et al., 2007) NOOR, M.A.; GRÜNBACHER, P.; BRIGGS, R.O. **A collaborative approach for Product Line Scoping : a case study in collaboration engineering.** In: Proceedings of the 25th IASTED International Multi-Conference. **Anais…** Innsbruck, Austria: 2007, p. 216 - 223.

(NOOR et al., 2008) NOOR, M.; RABISER, R.; GRUNBACHER, P. **Agile product line planning: A collaborative approach and a case study.** The Journal of Systems and Software, vol. 81, Jun. 2008, p. 868-882.

(OPENTHESAURUSPT, 2012) OPENTHESAURUSPT. **Dicionário de Sinônimos para a língua portuguesa.** Disponível em: <openthesaurus.caixamagica.pt>. Acesso em 01. mar. 2012.

(OTHERO, 2006) OTHERO, G. A. **Linguística computacional: uma breve introdução**. Revista Letras de Hoje, Porto Alegre: EDIPUCRS, v. 41, n. 2, p. 341-351.

(PRIBERAM, 2012) PRIBERAM Informática. **Dicionário Priberam da Língua Portuguesa.** Disponível em: <http://www.priberam.pt/dlpo/Conjugar.aspx?pal=ser>. Acesso em 01.mar.2012.

(SANTOS, 1999) SANTOS, A. R. **Metodologia científica: a construção do conhecimento**. 2ª ed. Rio de Janeiro: DP&A editora, 1999. 144 p.

(SCHMID, 1994) SCHMID, H. **Probabilistic Part-of-Speech Tagging Using Decision Trees.** In: Proceedings of the International Conference on New Methods in Language Processing. Anais… Manchester, UK, 1994.

(SCHMID, 2000) SCHMID, K. **Scoping software product lines — an analysis of an emerging technology.** In: Proceedings of the First Software Product Line Conference (SPLC1), 2000, Denver, Colorado, United States. **Anais…** 2000, p. 513-532.

(SEI, 2005) SEI – SOFTWARE ENGINEERING INSTITUTE. **A Framework for Software Product Line Practice, Version 5.0 - Scoping**. Disponível em: <http://www.sei.cmu.edu/productlines/frame_report/productLS.htm>. Acesso em 23 mar. 2011.

(SEI, 2005a) SEI – SOFTWARE ENGINEERING INSTITUTE. **Overview.** Disponível em: < http://www.sei.cmu.edu/productlines >. Acesso em 07 mai. 2011.

(SILVA et. al., 2007) SILVA, B. C. et. al. **Introdução ao Processamento das Línguas Naturais e Algumas Aplicações**. Núcleo Interinstitucional de Linguística Computacional. NILC - ICMC-USP. São Paulo. São Carlos, 2007.

(SILVA et. al., 2010) SILVA, J.; BRANCO, A.; CASTRO, S.; REIS, R. **Out-of-the-Box Robust Parsing of Portuguese**. In: Proceedings of the 9th International Conference on the Computational Processing of Portuguese (PROPOR'10), pp. 75–85.

(SINGHAL, 2001) SINGHAL, A. **Modern Information Retrieval: A Brief Overview**. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, Vol. 24, No. 4. (2001), pp. 35-42.

(SOMMERVILLE, 2007) SOMMERVILLE, I. **Engenharia de Software, 8ª edição**. São Paulo: Pearson Addison-Wesley, 2007. 552 p.

(ULLAH et al., 2010) ULLAH, M.I.; RUHE, G.; GAROUSI, V. **Decision support for moving from a single product to a product portfolio in evolving software systems**. The Journal of Systems and Software, vol. 83, Dec. 2010, p. 2496-2512.

(ZAHRA; MALUCELLI, 2009) ZAHRA, F.M.; MALUCELLI, A. **Poronto : ferramenta para construção semiautomática de ontologias em português**. Dissertação (Mestrado) - Pontifícia Universidade Católica do Paraná, Curitiba, 2009. 94 f.

(ZIADI et al., 2012) ZIADI, T.; FRIAS, L.; SILVA, M. M. A.; ZIANE, M. **Feature Identification from the Source Code of Product Variants**. In: Software Maintenance and Reengineering (CSMR), 2012 16th European Conference on, pp.417-422.

(YOSHIMURA et al., 2008) YOSHIMURA, K.; NARISAWA, F.; HASHIMOTO, K.; KIKUNO, T. **A Method to Analyze Variability Based on Product Release History: Case Study of Automotive System**. In: Proc. SPLC (2), 2008, pp.249-256.

APÊNDICE A – LPS criada – estrutura do XML

Exemplo da estrutura do arquivo XML gravado ao finalizar a criação de uma nova LPS utilizando a abordagem proposta. Neste exemplo, a funcionalidade principal é a “abrir contato”, e apenas uma funcionalidade está marcada como sinônima, a funcionalidade “acessar contatos”.

```
<TermoFuncionalidades>
    <termo>contato</termo>
    <funcionalidades>
        <FuncionalidadeBean>
            <funcionalidadeOriginal>abrir contato</funcionalidadeOriginal>
            <funcionalidadeStemm>abrir contato </funcionalidadeStemm>
            <arquivos>
                <string>LG Hotmail Phone C570Tagueado.txt</string>
                <string>LG Optimus 2X P990Tagueado.txt</string>
                <string>LG Optimus Black P970Tagueado.txt</string>
                <string>LG Optimus GT540Tagueado.txt</string>
                <string>LG Optimus L3 E400Tagueado.txt</string>
                <string>LG Optimus Pro C660Tagueado.txt</string>
            </arquivos>
            <verbosSinonimos>
```

```
        <string>abduzir</string>
        <string>abduzir</string>
        <string>abrir</string>
        <string>abrir</string>
        <string>afastar</string>
        <string>afastar</string>
        <string>desviar</string>
        <string>distrair</string>
</verbosSinonimos>
<sinonimos>
        <FuncionalidadeBean>
                <funcionalidadeOriginal>acessar contatos</funcionalidadeOriginal>
                <funcionalidadeStemm>acessar contato </funcionalidadeStemm>
                <arquivos>
                        <string>LG Hotmail Phone C570Tagueado.txt</string>
                </arquivos>
                <verbosSinonimos/>
                <sinonimos/>
                <verbo>acessar</verbo>
                <verboStemm>acessar</verboStemm>
                <nome>contatos</nome>
```

```
                        <nomeStemm>contato</nomeStemm>
                        <tipo>toda_linha</tipo>
                        <classe>4</classe>
                    </FuncionalidadeBean>
                </sinonimos>
                <verbo>abrir</verbo>
                <verboStemm>abrir</verboStemm>
                <nome>contato</nome>
                <nomeStemm>contato</nomeStemm>
                <tipo>toda_linha</tipo>
                <classe>4</classe>
            </FuncionalidadeBean>
        </funcionalidades>
</TermoFuncionalidades>
```

O formato do arquivo XML é explicado a seguir:

- A tag <TermoFuncionalidades> contém todas as funcionalidades identificadas relacionadas ao mesmo termo (substantivo). Contém outra tag chamada <termo> e outra chamada <funcionalidades>, que pode conter várias chamadas <FuncionalidadeBean>;
- A tag <FuncionalidadeBean> representa uma funcionalidade. Dentro dela:
    - A tag <FuncionalidadeOriginal> representa a funcionalidade identificada;

- A tag <FuncionalidadeStemm> representa a funcionalidade identificada, porém apenas com a raiz (*stemm*) das palavras;
- A tag <arquivos> contém a lista de arquivos processados nos quais esta funcionalidade foi identificada;
- A tag <verbosSinonimos> contém uma lista de verbos sinônimos ao verbo da funcionalidade;
- A tag <sinonimos> contém as funcionalidades consideradas sinônimas a esta. Pode conter várias ou nenhuma tag <FuncionalidadeBean>;
- A tag <verbo> representa o verbo da funcionalidade identificada;
- A tag <verboStemm> representa a raiz (*stemm)* do verbo da funcionalidade identificada;
- A tag <nome> representa o substantivo da funcionalidade identificada;
- A tag <nomeStemm> representa a raiz (*stemm)* do substantivo da funcionalidade identificada;
- A tag <adjetivo> representa o adjetivo da funcionalidade identificada;
- A tag <adjetivoStemm> representa a raiz (*stemm)* do adjetivo da funcionalidade identificada;
- A tag <tipo> representa a classificação da funcionalidade;
- A tag <classe> representa o padrão no qual a funcionalidade se encaixa.

O arquivo XML completo com LPS criada pelo participante 1 no experimento com 6 manuais pode ser encontrado por meiopor meio do endereço <https://docs.google.com/file/d/0BzHyutFIFv-ET1VycHlkRTRGcjQ/edit?usp=sharing>.