

SUMÁRIO

1. INTRODUÇÃO	4
2. PLANEJAMENTO DA REVISÃO SISTEMÁTICA	5
2.1 ELABORAÇÃO DAS QUESTÕES	5
2.1.1 <i>Objetivos da Questão</i>	5
2.1.2 <i>Qualidade e Amplitude da Questão</i>	5
2.2 SELEÇÃO DAS FONTES	7
2.2.1 <i>Critério de Definição das Fontes</i>	7
2.2.2 <i>Idioma do Estudo</i>	7
2.2.3 <i>Identificação das Fontes</i>	7
2.3 SELEÇÃO DOS ESTUDOS	7
2.3.1 <i>Definição dos Estudos</i>	7
2.4 EXTRAÇÃO DOS RESULTADOS	8
3. CONDUÇÃO DA REVISÃO SISTEMÁTICA – PROCESSO DE BUSCA	9
3.1 BUSCA NA BIBLIOTECA DIGITAL DA ACM	9
3.2 BUSCA NO IEEE	10
3.3. BUSCA NO SPRINGERLINK	12
3.4 BUSCA NO GOOGLE SCHOLAR	13
4. CONDUÇÃO DA REVISÃO SISTEMÁTICA – SELEÇÃO PRELIMINAR.....	15
4.1 SELEÇÃO PRELIMINAR NA BIBLIOTECA DIGITAL DA ACM	15
4.2 SELEÇÃO PRELIMINAR NO IEEE	15
4.3 SELEÇÃO PRELIMINAR NA SPRINGERLINK	15
4.4 SELEÇÃO PRELIMINAR NO GOOGLE SCHOLAR	15
5. CONDUÇÃO DA REVISÃO SISTEMÁTICA - EXTRAÇÃO DE DADOS	16
5.1 ESTUDOS DA ACM	16
5.1.1 <i>Estudo 01 - Critérios de Avaliação para o Desenvolvimento de LP</i>	16
5.2 ESTUDOS DA IEEE	20
5.2.1 <i>Estudo 02 - PuLSE-Eco V2.0: Uma Abordagem de Avaliação para a Análise de Benefícios e Riscos de LP</i>	20
5.2.2 <i>Estudo 03 - Recuperação e Incorporação de Artefatos em LP</i>	25
5.2.3 <i>Estudo 04 - Avaliação de Arquitetura de Software com o Método QADA</i>	28

5.2.4 Estudo 05 - <i>Influência de Fatores Organizacionais em Qualidade de Arquitetura de LP</i>	30
5.2.5 Estudo 06 - <i>Métricas de Utilização de Serviços para Avaliação da Estrutura de Arquiteturas de LP</i>	34
5.3 ESTUDOS DO GOOGLE SCHOLAR	39
5.3.1 Estudo 07 - <i>Prometheus: Uma Abordagem para a Modelagem de Qualidade em LP</i>	39
5.3.2 Estudo 08 - <i>O Modelo BAPO para Avaliação de LP</i>	43
5.3.3 Estudo 09 - <i>Avaliação Prática de Arquiteturas de LP</i>	45
5.4 ANÁLISE DOS RESULTADOS	48
6. CONSIDERAÇÕES FINAIS SOBRE A REVISÃO SISTEMÁTICA	53
REFERÊNCIAS	54

1. Introdução

A demanda por qualidade tem motivado a comunidade de software para o desenvolvimento de modelos para qualidade de software. Estes modelos estão orientados por duas visões: visão de processo e visão de produto. A visão de processo trata da avaliação e melhoria dos processos utilizados para o ciclo de vida de software. A visão de produto trata da avaliação de um produto de software, para verificação de sua qualidade.

Nas últimas décadas, vários padrões de qualidade de produto e processo de software foram propostos e estão disponíveis na literatura. Dessa forma, este trabalho apresenta o planejamento e a condução de uma revisão sistemática sobre padrões de qualidade e avaliação em linha de produto de software (LP). A revisão visa à identificação de padrões de qualidade e de avaliação de LP existentes e apresenta como resultado final uma análise, a partir do material selecionado, sobre o estado da arte e da prática no que diz respeito à qualidade de LP.

Esta monografia está organizada da seguinte maneira: a seção 2 apresenta o planejamento da revisão sistemática; a seção 3 apresenta o processo de busca dos estudos nas fontes selecionadas; a seção 4 apresenta a seleção preliminar dos estudos da revisão; a seção 5 apresenta a extração de dados dos textos selecionados e uma análise dos resultados; e a seção 6 apresenta as considerações finais a respeito da revisão.

2. Planejamento da Revisão Sistemática

Este trabalho está relacionado aos padrões e normas existentes sobre qualidade e avaliação em linha de produto de software (LP). O objetivo deste estudo é relacionar tais padrões identificando para cada um deles suas principais características, bem como suas limitações.

As seções a seguir apresentam o planejamento da revisão sistemática, de acordo com o protocolo proposto por Biolchini et al. (2005).

2.1 Elaboração das Questões

2.1.1 Objetivos da Questão

Questão Primária:

- Quais os padrões e normas de qualidade e avaliação existentes para LP?

2.1.2 Qualidade e Amplitude da Questão

Intervenção:

- Padrões e normas de qualidade e avaliação em LP

Controle:

- Nenhuma *baseline* ou conjunto de dados disponíveis para este estudo

Efeito:

- Análise sobre o estado da arte e da prática sobre qualidade e avaliação de LP, com base nos materiais selecionados durante a revisão
- Estudo inicial para a proposta de um modelo de qualidade para LP

Métricas:

- $\sum LP$ = Quantidade de estudos selecionados sobre qualidade/avaliação de LP como um todo (incluindo arquitetura de LP)
- $\sum Arq$ = Quantidade de estudos selecionados somente sobre qualidade/avaliação de arquitetura de LP

População:

- Arquitetura de LP
- LP como um todo

Resultados:

- Padrões e/ou normas de qualidade para LP

Aplicação:

- Academia e indústria interessadas em avaliação de qualidade de LP

Projeto Experimental:

- Nenhum método estatístico será usado
- Análise quantitativa e qualitativa com base nas métricas definidas para o estudo
- Apresentação de histogramas com os valores calculados para as métricas

Palavras-Chaves:

- Qualidade/avaliação de linha de produto (família de produto)

Termos Relacionados:

- **Qualidade/avaliação de linha de produto (família de produto):**
 - qualidade de linha de produto/família de produto (*product line/ product-line/product family/product-family quality*)
 - avaliação de linha de produto/família de produto:
 - (*product line/ product-line/product family/product-family evaluation*)
 - (*product line/ product-line/product family/product-family assessment*)
 - padrão/norma de qualidade (*quality standard*)
 - modelo de qualidade (*quality model*)
 - avaliação de qualidade (*quality evaluation/assessment*)
 - maturidade de qualidade (*quality maturity*)
 - atributo de qualidade (*quality attribute*)
 - fator de qualidade (*quality factor*)

2.2 Seleção das Fontes

2.2.1 Critério de Definição das Fontes

As fontes selecionadas para o estudo devem ter como característica principal a sua ampla utilização e indexação, bem como um vasto acervo disponível para consulta. Assim, foram definidas as seguintes fontes de estudos primários:

- bases de dados eletrônicas indexadas;
- máquinas de busca eletrônica.

2.2.2 Idioma do Estudo

O idioma definido leva em consideração à ampla disponibilidade dos estudos nas fontes. Assim, somente o idioma inglês foi escolhido para a busca nas fontes.

2.2.3 Identificação das Fontes

As buscas serão realizadas por meio da submissão de strings nas bases de dados eletrônicas e nas máquinas de buscas conforme a lista a seguir:

- bases de dados eletrônicas indexadas:
 - ACM
 - IEEE
 - SpringerLink
- Máquina de busca eletrônica:
 - Google Scholar

A string geral que será submetida à busca (adaptada para cada uma das fontes) é apresentada a seguir, no forma de uma expressão lógica com os operadores OR e AND:

((“product line” OR “product family” OR “product-line” OR “product-family”) AND quality AND (standard OR model OR evaluation OR assessment OR maturity OR attribute OR factor))

2.3 Seleção dos Estudos

2.3.1 Definição dos Estudos

Critérios de Inclusão dos Estudos:

- Questão Primária: padrões/normas de qualidade ou avaliação de LP

Critérios de Exclusão dos Estudos:

- Questão Primária: padrões/normas de qualidade ou avaliação que não se referem à LP; textos que não foram escritos em inglês; textos não disponíveis por completo (menos de 03 páginas), textos muito extensos (mais de 20 páginas), e textos no formato diferente de PDF.

Definição do Tipo dos Estudos:

- qualquer tipo de estudo, por exemplo:
 - qualitativos ou quantitativos;
 - observacionais;
 - de caracterização;
 - de viabilidade.

Procedimentos para Seleção Preliminar:

A seleção acontecerá por meio da preparação de strings de busca a partir das palavras-chaves e dos termos relacionados, apresentados na Seção 2.1.2 e da submissão de tais strings às fontes.

Para saber se um determinado estudo é relevante à pesquisa, serão lidos os resumos (*abstracts*) e, em caso positivo, estes serão selecionados para serem lidos por completo segundo os critérios de inclusão e exclusão.

Procedimentos para Seleção Final:

Para a seleção final serão lidos todos os artigos relevantes identificados na seleção preliminar.

2.4 Extração dos Resultados

Ao término da seleção final, será realizada uma síntese de cada estudo, considerando alguns itens como: identificação, proponente(s), caracterização e limitações de cada padrão/norma de qualidade e avaliação de LP.

3. Condução da Revisão Sistemática – Processo de Busca

Para a realização desta etapa as strings de busca elaboradas foram submetidas às fontes previamente selecionadas. Porém, o que se nota é que cada máquina de busca possui a sua forma própria de submeter às strings para a busca.

A seguir é apresentada a descrição da execução das buscas em cada uma das fontes.

3.1 Busca na Biblioteca Digital da ACM

A busca na biblioteca digital da ACM foi realizada no dia 07/11/2006, por meio do endereço eletrônico <http://portal.acm.org/advsearch.cfm>.

A figura a seguir ilustra o mecanismo de busca da ACM.

The screenshot shows the ACM Digital Library's advanced search interface. At the top, there's a header with the 'PORTAL' logo and navigation links: 'Subscribe (Full Service)', 'Register (Limited Service, Free)', and 'Login'. Below this, there's a search bar with a 'SEARCH' button. The main section is titled 'THE GUIDE TO COMPUTING LITERATURE' and 'Advanced Search'. It includes a 'Search Tips' link. The 'Desired Results' section has three options: 'must have all of the words or phrases' (selected), 'must have any of the words or phrases', and 'must have none of the words or phrases'. The 'Name or Affiliation' section has three dropdown menus for 'Authored', 'Edited', and 'Reviewed' by, each with 'all', 'any', and 'none' options. The 'Only search in:' section has four radio buttons: 'Title', 'Abstract' (selected), 'Review', and 'All Information'. A 'SEARCH' button is at the bottom right. A note at the bottom states: '*Searches will be performed on all available information, including full text where available, unless specified above.'

Figura 1: Mecanismo de Busca da Biblioteca Digital da ACM.

A página de busca avançada da ACM foi utilizada, sendo que a pesquisa foi realizada somente no resumo dos artigos, como mostra o campo *Only search in:* (Figura 1). Os seguintes campos foram preenchidos, conforme a seguir:

- ***must have all of the words or phrases:*** quality
- ***must have any of the words or phrases:*** "product line " "product family" "product-line" "product-family"

Assim, a string final utilizada na busca foi (o “+” significa obrigatório):

+abstract:quality abstract:"product line" abstract:"product family" abstract:"product-line" abstract:"product-family"

Desse modo, a pesquisa retornou 25 resultados, sendo que:

- 01 texto estava inacessível;
- 24 textos estavam disponíveis.

Outras 04 strings de busca foram utilizadas para pesquisa. Elas foram submetidas separadamente umas das outras, sendo elas:

1. *+abstract:"product line" +abstract:quality abstract:standard abstract:model abstract:evaluation abstract:assessment abstract:maturity abstract:attribute abstract:factor*
2. *+abstract:product-line +abstract:quality abstract:standard abstract:model abstract:evaluation abstract:assessment abstract:maturity abstract:attribute abstract:factor*
3. *+abstract:"product family" +abstract:quality abstract:standard abstract:model abstract:evaluation abstract:assessment abstract:maturity abstract:attribute abstract:factor*
4. *+abstract:product-family +abstract:quality abstract:standard abstract:model abstract:evaluation abstract:assessment abstract:maturity abstract:attribute abstract:factor*

O mecanismo de busca da ACM possui restrição quanto ao tamanho da string de busca. Dessa forma, uma busca para cada combinação dos termos *product line*, *product-line*, *product family* e *product-family* foi realizada. A primeira string retornou 12 textos, a segunda retornou 13 textos, a terceira retornou 01 texto e a quarta retornou 01 texto. Porém, percebeu-se que os textos encontrados com as 04 strings foram os mesmos encontrados com a string final utilizada para a ACM e, dessa maneira, optou-se pela string mais simplificada.

3.2 Busca no IEEE

A busca na biblioteca digital do IEEE foi realizada no dia 07/11/2006, por meio do endereço eletrônico <http://ieeexplore.ieee.org/search/advsearch.jsp>.

Assim como na ACM, a busca avançada foi realizada no resumo e também no título dos textos, sendo, para isso, utilizada a opção *option 2* do IEEE (Figura 2). No campo *Enter keywords, phrases or a Boolean expression* foi utilizada a string de busca:

((product line <or> product family <or> product-line <or> product-family) <and> quality <and> (standard <or> model <or> evaluation <or> assessment <or> maturity <or> attribute <or> factor)) <in> (ab,ti, metadata)

Outras opções foram definidas para esta busca (Figura 2), sendo elas:

- **Maximum:** 100, o valor máximo de resultados da busca
- **Display:** 100, o valor máximo de resultados que devem ser mostrados a partir de *Maximum*

A figura a seguir ilustra o mecanismo de busca da IEEE.

The screenshot displays the IEEE Xplore search interface. At the top, there is a navigation bar with links for Home, Login, Logout, Access Information, Alerts, Sitemap, and Help. Below this is a header with the IEEE Xplore logo and the text 'RELEASE 2.1'. The main navigation bar includes 'Advanced Search', 'BROWSE', 'SEARCH', 'IEEE XPLORE GUIDE', and 'SUPPORT'.

The search interface is divided into two main sections: 'OPTION 1' and 'OPTION 2'. 'OPTION 1' is titled 'Enter keywords or phrases, select fields, and select operators' and features three search boxes with 'in All Fields' dropdowns and 'AND' operators. 'OPTION 2' is titled 'Enter keywords, phrases, or a Boolean expression' and features a large text area containing a complex Boolean search query. Both options have 'Run Search' and 'Reset' buttons.

On the right side, there are several filter sections: 'Publications' with checkboxes for IEEE Periodicals, IEEE Conference Proceedings, and IEEE Standards; 'Other Resources (Available for Purchase)' with a checkbox for IEEE Books; 'Select date range' with a radio button for 'Search latest content update (04 Dec 2006)' and a 'From year' dropdown set to 'All'; 'Display Format' with radio buttons for 'Citation' and 'Citation & Abstract'; and 'Organize results' with dropdowns for 'Maximum' (100), 'Display' (100), 'Sort by' (Relevance), and 'In' (Descending).

Figura 2: Mecanismo de Busca da IEEE.

A busca retornou 49 resultados, sendo todos eles disponíveis.

3.3. Busca no SpringerLink

A busca no SprilgerLink foi realizada no dia 07/11/2006 por meio da página principal de busca <http://www.springerlink.com>, em que foi utilizado o mecanismo de composição de strings (Figura 3). Em tal mecanismo é possível combinar termos de busca com os operadores AND, OR e NOT, além de se poder escolher em que local do texto tal string pode ser buscada (título, resumo, autor, ISSN, ISBN).

A figura a seguir ilustra o mecanismo de busca da SpringerLink.

The screenshot shows the SpringerLink homepage. On the left, there is a sidebar with the SpringerLink logo and a list of recognized institutions: Athens Authentication Point, Sistema Integrado de Bibliotecas/USP (140-82-472), SIBI (312-25-383), Brazil-CAPES Springer (679-31-148), 667125 CAPES Consortia, Coordenacao de (716-56-436), Universidade de Sao Paulo (956-49-051), and a 'Welcome!' message with login/register links. The main content area has a 'Welcome to SpringerLink!' banner, a description of the database, and a search section titled 'Find content by keyword' with a 'more options' link. The search input field contains the query: 'su:("product line" OR "product-line" OR "product family" ...)' and a 'Go' button. Below the search bar, there are two columns of results: 'Content type' and 'Subject'. The 'Content type' column lists: All (3,089,218), Publications (17,118), Journals (1,581), Book Series (719), Books (14,818), and Reference Works (68). The 'Subject' column lists various fields with their respective counts: Architecture, Design and Arts (30), Behavioral Science (73,733), Biomedical and Life Sciences (819,030), Business and Economics (69,159), Chemistry and Materials Science (568,937), Computer Science (266,048), Earth and Environmental Science (284,456), Engineering (352,980), Humanities, Social Sciences and Law (181,401), Mathematics and Statistics (244,469), Medicine (651,246), and Physics and Astronomy (426,747). There is also a 'Featured library' section listing the Chinese Library of Science (10,468) and the Russian Library of Science (425,995).

Figura 3: Mecanismo de Busca da SpringerLink.

Assim, a string final de busca utilizada foi (su – significa sumário):

```
su:("product line" OR "product-line" OR "product family" OR "product-family") AND su:(quality)
AND su:(standard OR model OR evaluation OR assessment OR maturity)
```

Dessa forma, a busca retornou 20 resultados. Porém, no dia 21/11/2006 ao tentar efetuar o *download* dos textos, o mecanismo de busca da SpringerLink apresentou um erro ao buscar pela string definida. A Figura 4 apresenta o erro na avaliação da string ao restringir a busca ao resumo dos textos. O mecanismo não aceitou o índice de busca “su” relativo ao resumo de textos.

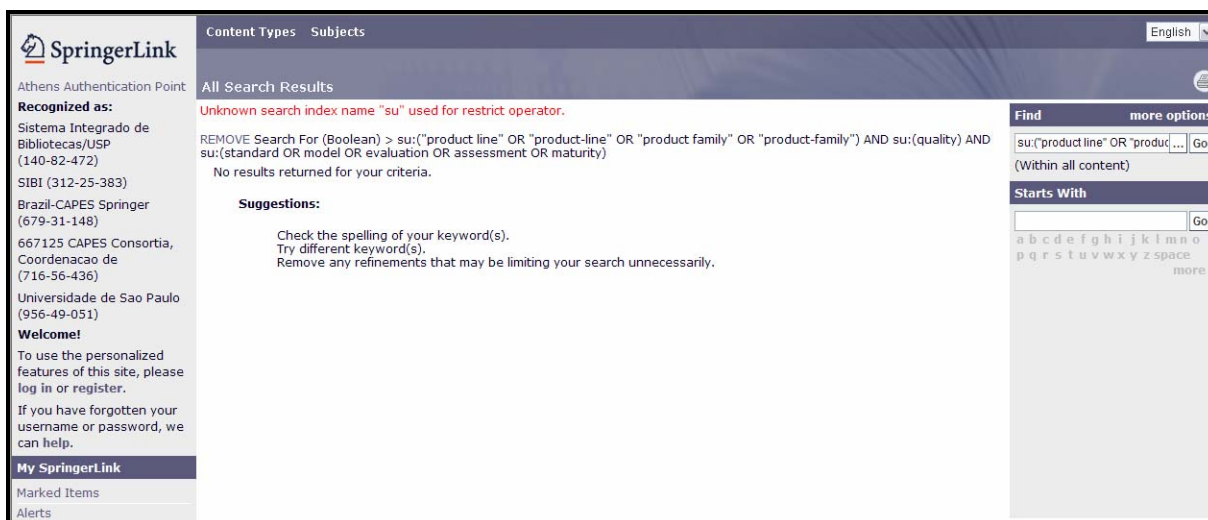


Figura 4: Erro no Mecanismo de Busca da SpringerLink.

Ao entrar em contato, a equipe responsável pelo SpringerLink reconheceu ser um erro do software de busca do mecanismo. Porém, o mecanismo encontra-se ainda indisponível para a consulta em resumos, o que impossibilitou a análise dos textos encontrados.

3.4 Busca no Google Scholar

A busca no Google Scholar foi realizada no dia 07/11/2006, de modo avançado por meio do endereço eletrônico http://scholar.google.com/advanced_scholar_search?hl=en&lr=.

Para tanto, alguns campos da busca foram preenchidos como segue (Figura 5):

- ***With all the words:*** quality software
- ***With at least one of the words:*** standard model evaluation assessment maturity attribute factor
- ***Where my words occur:*** in the title of the article

A String final de busca utilizada foi:

allintitle: quality ("product line" OR "product-line" OR "product family" OR "product-family")
 (standard OR model OR evaluation OR assessment OR maturity OR attribute OR factor)
 filetype:pdf

Google Scholar BETA **Advanced Scholar Search** [Advanced Search Tips](#) | [About Google Scholar](#)

Find articles with **all** of the words
 with the **exact phrase**
 with **at least one** of the words
without the words
 where my words occur

quality software
 standard model evaluation asse
 anywhere in the article ▼

100 results ▼ **Search Scholar**

Author Return articles written by
 e.g., "PJ Hayes" or McCarthy

Publication Return articles published in
 e.g., J Biol Chem or Nature

Date Return articles published between
 e.g., 1996

Subject Areas
☒ Return articles in all subject areas.
☐ Return only articles in the following subject areas:

- ☐ Biology, Life Sciences, and Environmental Science
- ☐ Business, Administration, Finance, and Economics
- ☐ Chemistry and Materials Science
- ☐ Engineering, Computer Science, and Mathematics
- ☐ Medicine, Pharmacology, and Veterinary Science
- ☐ Physics, Astronomy, and Planetary Science
- ☐ Social Sciences, Arts, and Humanities

Figura 5: Mecanismo de Busca do Google Scholar.

Dessa forma, a busca retornou 01 texto. Para tentar contornar a falta de resultados, foi realizada a busca novamente, porém no campo *Where my words occur* optou-se pela opção *anywhere in the article*. Assim, o mecanismo de busca retornou 11.600 resultados. Como este é um número inviável para a realização desta revisão sistemática, somente os 10 primeiros foram considerados, estando todos eles disponíveis.

4. Condução da Revisão Sistemática – Seleção Preliminar

Conforme o planejamento da revisão sistemática, nesta etapa os resumos de todos os textos coletados no processo de busca foram lidos.

As seções a seguir apresentam a seleção preliminar dos textos candidatos à leitura completa.

4.1 Seleção Preliminar na Biblioteca Digital da ACM

O seguinte texto foi previamente selecionado para sua completa leitura:

1. *Product Line Software Engineering of Embedded Systems*

4.2 Seleção Preliminar no IEEE

Os seguintes textos foram previamente selecionados para sua completa leitura:

1. *An Assessment Approach To Analyzing Benefits and Risks of Product Lines*
2. *Asset Recovery and Their Incorporation into Product Lines*
3. *Evaluating the Portability and Maintainability of Software Product Family Architecture: Terminal Software Case Study*
4. *The Influence of Organizational Factors on the Success and Quality of a Product-Line Architecture*
5. *Using Service Utilization Metrics to Assess the Structure of Product Line Architectures*

4.3 Seleção Preliminar na SpringerLink

Nenhum texto selecionado por causa do problema apresentado pelo mecanismo de busca da SpringerLink.

4.4 Seleção Preliminar no Google Scholar

Os seguintes textos foram previamente selecionados para sua completa leitura:

1. *Practical Evaluation of Software Product Family Architectures*
2. *Quality Modeling for Software Product Lines*
3. *Software Product Family Evaluation*

5. Condução da Revisão Sistemática - Extração de Dados

Nesta etapa da revisão sistemática foi realizada a extração dos dados a partir da leitura dos textos previamente selecionados na seção anterior.

Para tanto, foi necessária a elaboração de um formulário de extração de dados com o objetivo de organizar as informações obtidas na leitura dos textos. Assim, a Figura 6 apresenta o formulário elaborado.

Identificação: _____
Proponente(s): _____
Referenciado Por: _____
Características e Limitações: _____

Figura 6: Formulário de Extração de Dados.

O formulário é composto por quatro campos principais, sendo eles:

- **Identificação:** texto descritivo do item em questão (padrão ou norma);
- **Proponente(s):** indica a pessoa, o grupo ou a organização que criou o padrão/norma ou item em questão;
- **Referenciado Por:** texto(s) que referencia(m) o item em questão;
- **Características e Limitações:** uma síntese do item em questão apresentando as suas características principais, bem como as suas limitações.

Dessa forma, todos os textos selecionados na seção anterior foram lidos por completo. As seções a seguir apresentam uma síntese de cada estudo e ao final é apresentada uma análise acerca dos resultados.

5.1 Estudos da ACM

A extração dos dados dos textos da ACM resultou no item apresentado a seguir.

5.1.1 Estudo 01 - Critérios de Avaliação para o Desenvolvimento de LP

Identificação:

Critérios de Avaliação para o Desenvolvimento de LP

Proponente(s):

NIEMELÄ, E. e IHME, T. (*VTT Electronics - Embedded Software - Finland*)

Referenciado Por:

Niemelä, E., Ihme, T. 2001. *Product Line Software Engineering of Embedded Systems*. ACM SIGSOFT Software Engineering Notes, Vol. 26, No 3 (May 2001). pp. 118-125.

Niemelä, E.; Matinlassi, M.; Taulavouri, A. 2004. *Practical Evaluation of Software Product Family Architectures*, Proceedings of the 3rd International Conference on Software Product Lines, pp. 130-145.

Características e Limitações:

A primeira etapa no desenvolvimento de LP é a análise do negócio e do escopo da LP, que visa estimar os benefícios da abordagem para a organização e os domínios potenciais de reutilização.

Um dos grandes problemas em LP é o conhecimento do domínio, bem como: qual o tipo de conhecimento que existe sobre o domínio? Como ele é representado? Qual a sua qualidade?

Assim, ao se iniciar o desenvolvimento de LP, 03 questões devem ser respondidas, sendo elas:

- A abordagem de LP é propícia à área de negócio da organização?
- Onde, quando e como a abordagem de LP poderia ser aplicada?
- Que práticas devem ser desenvolvidas para alcançar um custo efetivo com LP?

A identificação dos *business drivers* e dos *stakeholders*, bem como uma estimativa dos possíveis consumidores dos produtos ajudam a responder a primeira questão. Uma análise de custo-benefício deve ser realizada investigando o número de possíveis produtos que podem ser gerados pela LP. A primeira questão equivale à elaboração de um modelo econômico para LP.

A análise dos *business drivers* é realizada segundo 03 pontos de vista: processos e cultura da organização; arquitetura de LP; e gerenciamento de artefatos (Figura 1).

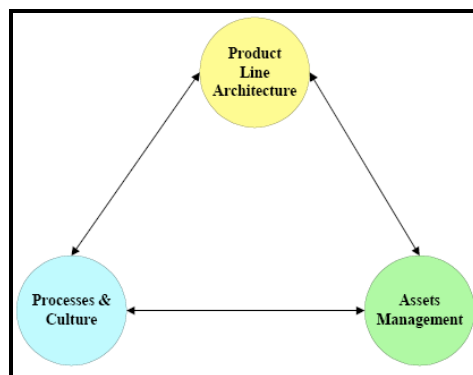


Figura 1: Visões do Desenvolvimento de LP (NIEMELÄ, 2001).

A primeira visão analisa o estado do desenvolvimento e da conformidade de software entre as práticas de desenvolvimento de software que estão descritas e as que são aplicadas. A segunda visão estima as propriedades existentes e desejáveis e a qualidade dos produtos inseridos na LP. A terceira visão estima o potencial de manutenção dos artefatos por meio do desenvolvimento de mecanismos de suporte como, por exemplo, gerenciamento e configuração de software. A Tabela 1 apresenta as 03 visões e os elementos para cada visão.

Tabela 1: Identificação dos *Business Drivers* de uma LP (NIEMELÄ, 2001).

<i>Elements</i>	<i>Organisation culture and processes</i>	<i>Product line architecture</i>	<i>Assets Management</i>
Product assortment Expected sales The number of products The value of reuse for products Similarity and diversity of products Product features	Cultural differences in multi-site software development Current state of development processes	Commonality and variability of the functional features of the products Quality requirements of the products	Mechanisms to manage variability inside the product line
Business manners In-house components Commercial off-the-shelf components Original component manufactures Contracted software Appropriateness of the product -line approach	Component development, acquisition and utilisation processes	Artefacts and their representations Quality of artefacts Documentation of artefacts Business rationale: cost, time-to-market, partnerships	Mechanisms to share the properties of the product line with business stakeholders

A análise dos *business drivers* estima os benefícios da adoção de LP, sendo uma maneira de manter os gerentes comprometidos com a abordagem.

A análise de domínio ajuda a responder a segunda questão. O escopo, as semelhanças, as variabilidades e a qualidade é o foco principal, assim como o comprometimento do pessoal, o conhecimento do domínio e as habilidades técnicas são analisadas para se saber se uma organização possui as habilidades profissionais exigidas para o desenvolvimento de LP. A Tabela 2 apresenta os elementos e as pré-condições na definição do domínio.

Tabela 2: Identificação do Domínio: Pré-condições de uma LP (NIEMELÄ, 2001).

<i>Elements</i>	<i>Organisation culture and processes</i>	<i>Product line architecture</i>	<i>Assets Management</i>
Domain analysis Domains and their scopes Commonalities and variabilities within domains Stability and maturity of domains Representations of reusable domain knowledge Quality of these representations Evolutionary or revolutionary approach	Support for domain analysis Collecting and ordering domain knowledge Representations for knowledge management Individual and organisational readiness in product line development	Quality requirements of the application domains Diversity of domains and implementation technology Standards Mapping domains to a product line architecture Domain rationale	Conceptual classification of the requirements and product features Models for representing and storing requirements and architecture into the core assets repository Models for storing documents of core assets
Personnel Management Domain expertise Technical skills	Commitment Decision making Roles, rights and responsibilities Expertise of software processes and metrics	Domain analysts Domain and product architects Dissemination of architecture Expertise of design and analysis of software architectures and components	Asset managers Distributed and virtual assets management infrastructure Expertise of data modelling and management

A análise de modelos, métodos e ferramentas, bem como artefatos pré-existentes e práticas aplicadas ajuda a responder a terceira questão. Para tanto, é necessário definir quais as partes de software que são valiosas para fazerem parte do núcleo de artefatos. O desenvolvimento do núcleo de artefatos inicia com uma análise do estado atual da definição dos requisitos, da arquitetura de software e da descrição dos componentes. Os artefatos que possuem um maior potencial de reutilização serão escolhidos para fazerem parte do desenvolvimento do núcleo de artefatos (Tabela 3).

Tabela 3: Desenvolvimento do Núcleo de Artefatos (NIEMELÄ, 2001).

Elements	Organisation culture and processes	Product line architecture	Assets Management
Core assets engineering Definition of core assets of a domain Evaluation of the current state of core assets The characteristics of subsequent versions of the product line architecture The product line architecture Architecture styles and component models Design and analysis methods and tools of requirements, architecture and components A validated product line	Adaptation of methods and tools to organisation settings Practices and metrics of product line processes (domain engineering process, component acquisition process and product engineering processes)	Adopted architecture design and analysis methods and techniques for the domain Architectural vision of a domain Architectural styles and patterns Views of a product line architecture Product line architecture Stable design centres with required variant aspects Adequacy of component models Design rationale, principles, concepts and rules	Initiating the core assets repository Principles, concepts and rules to store and retrieve core assets
Evolution of core assets Management of product line development and product development (variability in space) Change management: changes of customers and end-users' needs, problem domains, technologies and standards (variability in time)	Practices and metrics for the SCM process of the product line and the PDM process of the product line	Mapping variability in space and time to the product line architecture Principles and rules for the evolution of the product line architecture Reusability, modifiability and maintainability of a product line architecture	Mechanisms to manage variability in space and time

Uma análise da evolução do núcleo de artefatos é desejável para as 03 visões, uma vez que alguns produtos podem reutilizar somente a arquitetura de LP e mudar totalmente o código para cada produto gerado.

Um estudo com base no modelo de 03 visões foi desenvolvido na forma de entrevistas em empresas de desenvolvimento de software embarcado segundo o *framework* da Figura 2.

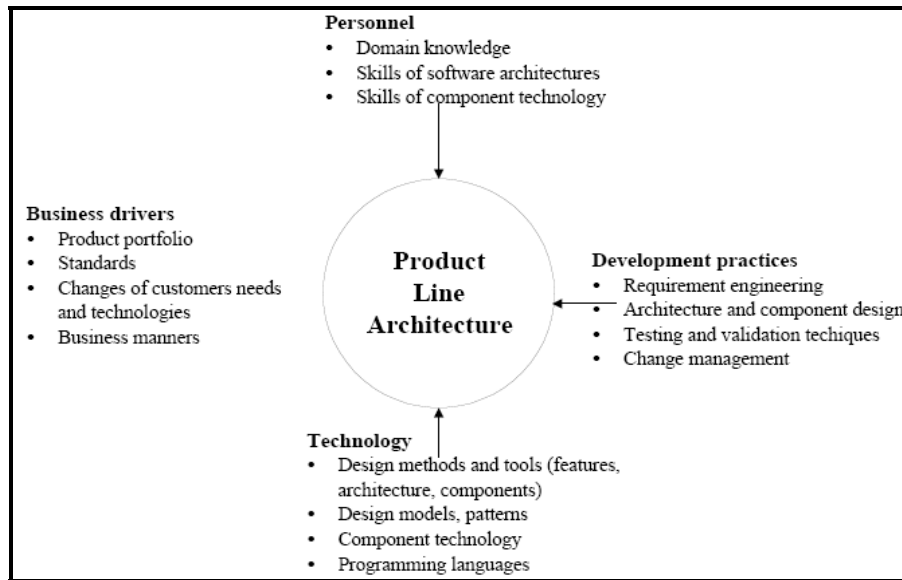


Figura 2: *Framework* de Entrevista (NIEMELÄ, 2001).

Um dos resultados do estudo mostrou que somente 30% das empresas entrevistadas possuem uma visão e um entendimento claro de quais são as propriedades mais importantes em seus produtos.

A grande limitação do modelo encontra-se em como avaliar, com base nos critérios apresentados, se a organização está apta a adotar a abordagem de LP. O estudo realizado pelos autores mostra somente como as 03 visões foram definidas. Porém, a partir da definição destas como saber se a empresa está preparada para iniciar o desenvolvimento de uma LP para o seu domínio? Acredita-se que algumas métricas ou pesos para cada critério seja uma forma de medir quantitativamente se a empresa se beneficiará da abordagem de LP.

5.2 Estudos da IEEE

A extração dos dados dos textos da IEEE resultou nos itens apresentados a seguir.

5.2.1 Estudo 02 - PuLSE-Eco V2.0: Uma Abordagem de Avaliação para a Análise de Benefícios e Riscos de LP

Identificação:

PuLSE-Eco V2.0: Uma Abordagem de Avaliação para a Análise de Benefícios e Riscos de LP

Proponente(s):

SCHMID, K. (*Fraunhofer Institute for Experimental Software Engineering- Germany*)

Referenciado Por:

Schmid, K. 2001a. *An Assessment Approach to Analyzing Benefits and Risks of Product Lines*, Proceedings of the 25th International Computer Software and Applications Conference on Invigorating Software Development, pp. 525-530.

Características e Limitações:

Grande parte das atividades técnicas de LP concentra-se no desenvolvimento da arquitetura de LP e na análise de domínio. Porém, uma análise de riscos e benefícios é fundamental para as empresas que desejam seguir a abordagem de LP. Algumas questões que precisam ser respondidas como parte da definição do escopo de LP:

- A abordagem de LP deve focar somente uma linha particular de produtos (domínio)?
- Todas as áreas técnicas (sub-domínios) devem ser tratadas da mesma maneira, ou deve-se priorizar somente algumas?
- Quais as partes de software que devem ser desenvolvidas para reutilização?
- Quais os benefícios que se pode esperar no desenvolvimento de LP?

Assim, uma metodologia que busca responder tais perguntas está em desenvolvimento como parte da abordagem PuLSE (*Product Line Software Engineering*), chamada PuLSE-Eco V2.0. Esta metodologia visa cobrir todas as questões levantadas, uma vez que a abordagem original (PuLSE) encontrou problemas em sua aplicação na prática industrial.

O PuLSE-Eco V2.0 baseia-se em vários padrões de qualidade e de avaliação de processo como CMM, Bootstrap, SPICE e FAME. No contexto de reutilização de software, tais padrões já vêm sendo usados para cobrir dois aspectos: a dimensão do processo (para reutilização) e a dimensão do domínio (conjunto de produtos). Dessa forma, a abordagem PuLSE-Eco V2.0 avalia os domínios individualmente ao invés da LP como um todo.

A abordagem PuLSE-Eco V2.0 é formada por três fases:

- **Mapeamento da LP:** descrição em alto nível da LP em análise e dos domínios relevantes a ela. Isto fornece uma base de comunicação para as próximas fases da abordagem;
- **Definição do Escopo do Domínio:** uma avaliação dos benefícios e dos riscos do desenvolvimento de LP é realizada em nível de LP e para cada sub-domínio. Como

resultado, as áreas mais promissoras são identificadas e alguns domínios já podem ser removidos da análise detalhada. Isso reduz o esforço na definição de escopo dos artefatos da LP;

- **Definição do Escopo dos Artefatos:** é uma fase baseada em análise quantitativa.

A abordagem PuLSE-Eco V2.0 está organizada de forma que as duas primeiras fases podem ser aplicadas sem a terceira. Assim, a avaliação dos benefícios e riscos consiste nas duas primeiras fases. Quando a abordagem foi desenvolvida, foi possível se basear em abordagens de avaliação de processos como o FAME, que é uma instanciiação do SPICE fazendo uso de toda a experiência adquirida pela comunidade ao longo dos anos. Porém, existe uma diferença entre a avaliação de maturidade de um processo e a avaliação de potenciais LPs. Um mapeamento entre as duas formas de avaliação é apresentado na Tabela 1.

Tabela 1: Mapeamento dos Conceitos de Avaliação (SCHMID, 2001a).

Process Assessment Concept	Product Line Assessment Concept
Process framework [11] + process existence indicators	The product line mapping method (cf. Section 4) is used for developing a reference description of the domains relevant to the product line [20]
Standardized capability indicators are provided [11]	Assessment indicators have been specifically developed for this approach using a goal-driven methodology (cf. Section 5)
A standardized process for performing assessments is used [2]	A variant of the FAME-process is used for performing the assessment (cf. Section 6)

Estas diferenças são traduzidas em três problemas que devem ser resolvidos no desenvolvimento de um método de avaliação de LP (PuLSE B&R):

- um framework de referência para LP teve de ser desenvolvido;
- medidas de avaliação para benefícios e riscos de LP foram identificadas;
- um processo teve de ser definido para a execução da avaliação de LP.

Em avaliação de LP não é usada uma estrutura de referência fixa, como áreas-chaves de processo do CMM, mas é necessário identificar uma estrutura de referência que deve ser modelada para cada LP. Assim, abordagem PLM (*Product Line Mapping*) foi desenvolvida para tratar as questões sobre modelos de referência. Este método serve para dois propósitos: identificar a estrutura de alto nível de uma LP em termos das principais funcionalidades e como esta se relaciona aos produtos da linha e aos domínios técnicos - assim, a abordagem pode ser usada como um ponto de início para a análise de domínio; e fornecer a base de comunicação para a realização da avaliação.

A abordagem PLM não foca diretamente nos produtos particulares que podem ser desenvolvidos, mas modela a intersecção dos domínios e das funcionalidades dos produtos. Esta abordagem é dividida em várias etapas, sendo elas:

1. identificar os produtos existentes e futuros que podem ser relevantes a LP;

2. desenvolver um plano de apresentação destes produtos;
3. identificar as maiores funções/características que são relevantes as funcionalidades destes sistemas;
4. agrupar tais funções em áreas funcionais maiores;
5. desenvolver um plano de apresentação dos domínios que mostram as suas inter-relações;
6. analisar a informação para a existência de domínios internos adicionais;
7. garantir consistência dos sistemas existentes;
8. desenvolver uma matriz inicial de produto/função.

Além da abordagem PLM, um *framework* de referência para realizar a avaliação é necessário. Este *framework* é usado para determinar a avaliação atual dos domínios ao longo de varias dimensões. O início da avaliação se dá usando uma abordagem estruturada para derivar as questões de avaliação, similar ao GQM (*Goal Question Metric*). Como entrada, são fornecidas as dimensões de avaliação, resultantes de um *survey* de avaliações no contexto de engenharia de domínio. Tais dimensões são apresentadas na figura a seguir.

Viability dimensions	Benefit dimensions
Maturity	Market potential – External
Stability	Market potential – Internal
Resource constraints	Commonality and Variability
Organizational constraints	Coupling and Cohesion
	Existing Assets

Figura 2: Dimensões de Avaliação (SCHMID, 2001a).

Cada uma das dimensões é dividida em questões principais (atributos) e questões detalhadas (indicadores). Um exemplo de arranjo hierárquico entre as dimensões e questões é apresentado na figura a seguir.

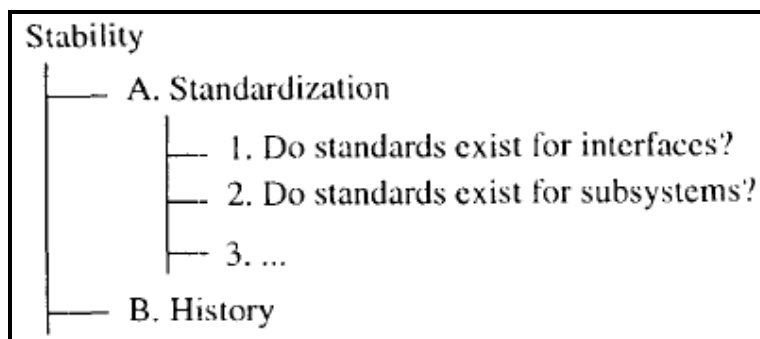


Figura 3: Exemplo de Decomposição de Avaliação (SCHMID, 2001a).

O *framework* atualmente é composto de 09 dimensões de avaliação, 25 atributos (questões principais) e 93 indicadores (questões detalhadas). Durante a execução da avaliação, as respostas são contabilizadas de acordo com uma escala de quatro níveis: *F=Fully*, *L=Largely*, *P=Partially* e *N=None*.

Dessa forma, um processo de avaliação foi proposto com base na abordagem FAME, e apresentado na figura a seguir.

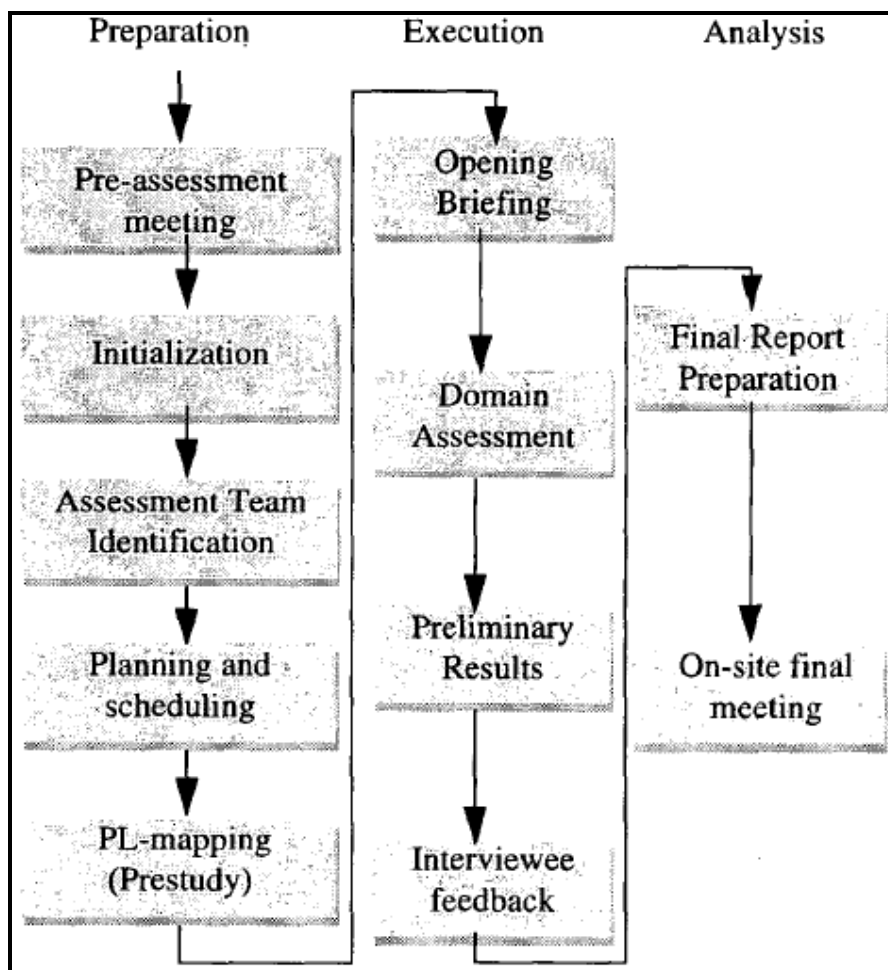


Figura 4: Processo de Avaliação (SCHMID, 2001a).

O processo consiste de 3 etapas:

- **preparação:** a equipe de avaliação é formada e a finalidade da avaliação é definida. A maior tarefa desta etapa é o desenvolvimento do mapeamento da LP gerando uma descrição detalhada da LP que serve de entrada para a avaliação;
- **execução:** a avaliação é executada como um conjunto de entrevistas estruturadas, que são dirigidas por questionários que descrevem o *framework* de avaliação. Resultados preliminares são apresentados aos entrevistados para possíveis correções;

- **análise:** um relatório de avaliação é preparado e o julgamento final dos benefícios e riscos de reutilização nos vários domínios é desenvolvido. Como maior resultado, são elaboradas recomendações para quais domínios deve ser aplicado a reutilização.

A abordagem proposta apresenta a descrição explícita de LP como um modelo de referência similar a um modelo de referência de processo usado pelas abordagens de avaliação de processos. Além disso, a abordagem é fortemente baseada em trabalhos existentes de avaliação de maturidade de processos e tem como vantagens o grande conhecimento e experiência no campo de avaliação de processos. A abordagem tem a vantagem de reduzir o esforço na análise de sub-domínios que já podem ter sido excluídos em etapas anteriores da abordagem.

Acredita-se que este trabalho esteja limitado com relação à falta de um estudo de caso ou de um projeto piloto para ilustrar as etapas de avaliação de uma potencial LP. Talvez uma versão estendida do artigo ou um relatório técnico possam elucidar a abordagem e a sua aplicação.

5.2.2 Estudo 03 - Recuperação e Incorporação de Artefatos em LP

Identificação:

Recuperação e Incorporação de Artefatos em LP

Proponente(s):

KNODEL, J.; JOHN, I.; DHARMALINGAM, G. (*Fraunhofer Institute for Experimental Software Engineering- Germany*)

PINZGER, M. (*Institute of Informatics – Switzerland*)

USERO, F. (*Telvent – Spain*)

ARCINIEGAS, A. L. (*Departamento de Ingeniería de Sistemas Telemáticos – Spain*)

RIVA, C. (*Software Architecture Group – Nokia Research – Finland*)

Referenciado Por:

Knodel, J.; John, I.; Dharmalingam, G.; Pinzger, M.; Arciniegas, A. L.; Riva, C. 2005. *Asset Recovery and their Incorporation into Product Lines*, Proceedings of the 12th Workshop Conference on Reverse Engineering, pp. 120-129.

Características e Limitações:

Raramente uma LP é desenvolvida a partir do zero. Na maioria das vezes a LP é concebida quando já existe um número suficiente de produtos que tornam o domínio consolidado.

Muitas vezes é necessário adicionar novos artefatos ao núcleo de artefatos de uma LP. Tais artefatos podem ter origem dos mais variados tipos. Para tanto, é necessário estabelecer a forma como estes serão incorporados ao núcleo de artefatos da LP, sendo que as possíveis formas são:

- **reutilizar o artefato na forma em que ele se encontra:** incorporar o artefato na forma como foi encontrado ou com poucas alterações, sendo que a qualidade é boa o suficiente para migrar o artefato para o núcleo com esforço limitado;
- **reutilizar o artefato com modificações:** incorporar o artefato após mudanças substanciais;
- **recuperar e adaptar o artefato:** recuperar e adaptar um artefato existente com técnicas de engenharia reversa;
- **re-implementação do artefato:** desenvolver um novo artefato baseado no já existente e descartar o antigo artefato.

Assim, um processo é proposto para realizar a incorporação dos artefatos ao núcleo de artefatos de uma LP. O processo utiliza o *Software Process Engineering Meta-model* (SPEM) da OMG como notação para a incorporação de artefatos à infra-estrutura da LP. O processo é focado principalmente nas atividades de engenharia de domínio, como mostra a Figura 1.

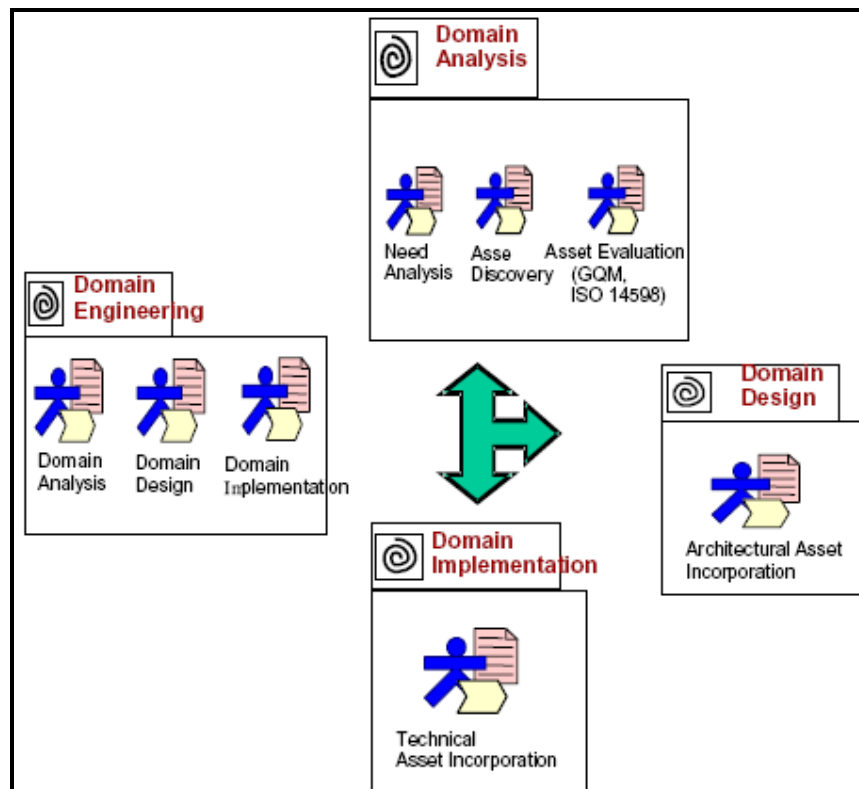


Figura 1: Atividade de Engenharia de Domínio (KNODEL et al., 2005).

Durante a análise de domínio, uma análise de necessidades é realizada e tem como saída uma descrição genérica dos artefatos mostrando os requisitos necessários para que o artefato possa ser incorporado ao núcleo de artefatos da LP. Em seguida, a atividade de descoberta de artefatos visa identificar os artefatos existentes que são candidatos a suprir as necessidades já identificadas. A atividade seguinte visa a avaliação do impacto, do custo e da qualidade necessária para a incorporação do artefato na LP. A proposta é utilizar algum padrão para a avaliação como, por exemplo, a ISO 14598 ou GQM (*Goal-Question-Metric*).

Ao incorporar o artefato ao núcleo de artefatos da LP é necessário incluí-lo também na arquitetura de LP, o que é difícil de conseguir por causa de potenciais efeitos colaterais e restrições técnicas.

A Figura 2 apresenta todas as atividades do processo de incorporação de artefatos.

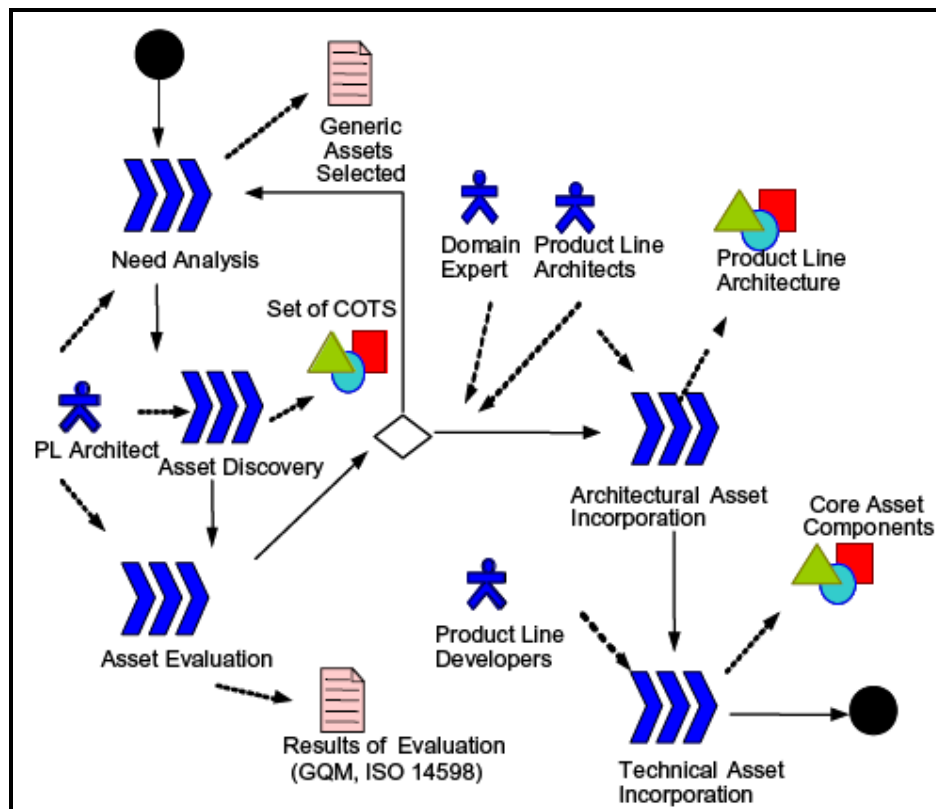


Figura 2: Cargos e Atividades na Incorporação de Artefatos (KNODEL et al., 2005).

A grande limitação do trabalho reside na avaliação de qualidade do núcleo de artefatos da LP. Avaliar o artefato que está sendo incorporado de forma isolada – como a avaliação de um produto específico - não garante que a qualidade da LP continue a mesma ou melhore. Muito pelo contrário, a existência de uma atividade de análise técnica no processo de incorporação é uma evidência de que a qualidade da LP como um todo deve ser avaliada.

5.2.3 Estudo 04 - Avaliação de Arquitetura de Software com o Método QADA

Identificação:

Avaliação de Arquitetura de Software com o Método QADA (*Quality-driven Architecture Design and quality Analysis*)

Proponente(s):

MATINLASSI, M. (*VTT Technical Research Centre of Finland*)

Referenciado Por:

Matinlassi, M. 2004. *Evaluating the Portability and Maintainability of Software Product Family Architecture: Terminal Software Case Study*, Proceedings of the 4th Working IEEE/IFIP Conference on Software Architecture, pp. 295-298.

Características e Limitações:

O trabalho apresenta um estudo de caso na avaliação de uma arquitetura de LP para terminais de coleta públicos. A LP possui dois tipos de produtos: terminais fixos e terminais portáteis. A idéia é utilizar o método QADA (*Quality-driven Architecture Design and quality Analysis*) para avaliar a portabilidade e manutenibilidade da arquitetura de LP após a inclusão de um novo membro da LP.

Antes de iniciar a avaliação, todos os produtos da LP foram estudados por meio da documentação existente e a arquitetura foi documentada usando o QADA. O QADA permite a descrição da arquitetura em 4 pontos de vista (estrutural, comportamental, implantação e desenvolvimento) e 2 níveis de abstração (conceitual e concreto).

O processo de avaliação de portabilidade e manutenibilidade inclui as seguintes etapas:

1. **descrição da arquitetura de LP:** de acordo com as visões arquiteturais;
2. **criação de categorias de cenários:** criar categorias de mudança para o domínio do problema e identificar categorias de tarefas de manutenção.
3. **identificação de cenários:** cenários de manutenção e mudança são criados baseados nas categorias definidas na etapa anterior;
4. **associar um peso para cada cenário:** pode ser, por exemplo, a probabilidade de ocorrência em um certo período de tempo (somente para avaliação de manutenibilidade);
5. **avaliar o efeito dos cenários em elementos da arquitetura;**
6. **revelar interações de cenário:** quais cenários afetam o mesmo componente.

A descrição da arquitetura foi feita com base nas visões do QADA, nos diagramas de classe e de interface de cada componente, bem como entrevistas com os desenvolvedores. Ainda, as variabilidades da arquitetura foram capturadas.

No estudo de caso, 4 categorias de portabilidade e 4 categorias de manutenção foram definidas, sendo elas:

- **Categorias de Portabilidade (Mudança):**
 - Mudanças de Interface de HW/SW;
 - Mudanças de Plataforma de HW;
 - Mudanças de Plataforma de SW;
 - Mudanças em Requisitos de Segurança.
- **Categorias de Manutenção:**
 - Extensões de *Features*;
 - Exclusão de Funcionalidades não-desejadas;
 - Entrada para Novos Dispositivos;
 - Melhorias na Qualidade (Segurança e Desempenho).

Assim, foram definidos 20 cenários de portabilidade e 17 cenários de manutenção. Medidas qualitativas foram associadas à ocorrência dos cenários. Com relação a período de tempo, algumas medidas como “dentro dos próximos 2 anos” foram utilizadas.

Após a criação dos cenários, estes foram exercitados com relação a quais componentes da arquitetura foram afetados e que tipos de mudança foram necessárias visando estimar o efeito de cada cenário na arquitetura. Um exemplo de cenário de portabilidade e uma pequena avaliação dos efeitos na arquitetura são apresentados a seguir.

Cenário: Conexão da impressora mudou de local para *wireless (bluetooth)*.

Efeito na Arquitetura: A impressão é feita por uma API. Uma classe *wrapper* teve de ser criada para o componente *User Interface* para habilitar a impressão no novo ambiente.

Resultado: Mudança moderada em um componente.

Ao final da execução dos cenários, foram contabilizados quantos componentes foram afetados por um cenário e quantos cenários afetaram um só componente. Por exemplo, um componente foi afetado por 07 cenários. Assim, a interação de cenários pode ser traduzida como uma fraca separação de interesses do componente. Por outro lado, um cenário afetou 5 componentes diferentes, o que pode ser visto como um risco para a arquitetura.

Como lições aprendidas, a avaliação de portabilidade deve ser feita por um método baseado em cenários em nível arquitetural. A descrição da arquitetura tem papel fundamental em sua

avaliação de qualidade. Os pontos de vista arquiteturais do QADA são projetados especialmente em atributos de qualidade. A avaliação de manutenibilidade é bastante parecida com a de portabilidade. Cenários de manutenção incluem modificações em funcionalidade e os diagramas na visão comportamental são usados como fonte de informação.

A avaliação realizada revelou novas soluções para a arquitetura existente ser usada em um novo produto operando em um PDA, além de fornecer um *feedback* em como melhorar as descrições do método QADA para um maior suporte à avaliação de qualidade.

Acredita-se que o estudo seja limitado pelo fato da avaliação ser somente qualitativa. Talvez o processo de avaliação usando o método QADA possa ser complementado com um conjunto de métricas relacionadas aos cenários e aos componentes, dando subsídio para a realização de vários estudos experimentais com relação a portabilidade e manutenção da arquitetura ao se desenvolver um novo membro da LP.

Outra limitação, já que o trabalho foi realizado em uma LP, é a necessidade de se avaliar não somente a arquitetura de LP, mas também um subconjunto de todas as possíveis arquiteturas derivadas da LP para seus membros, o que não é uma tarefa simples. Talvez o uso de cenários possa ajudar na avaliação de cada arquitetura do subconjunto de produtos da LP.

5.2.4 Estudo 05 - Influência de Fatores Organizacionais em Qualidade de Arquitetura de LP

Identificação:

Influência de Fatores Organizacionais em Qualidade de Arquitetura de LP

Proponente(s):

JAKTMAN, C. B. (*University of Technology – Sidney*)

Referenciado Por:

Jaktman, C. B. 1998. *The Influence of Organisational Factors on the Success and Quality of a Product-line Architecture*, Proceedings of the Software Engineering Conference, pp. 02-11.

Características e Limitações:

Normalmente, as organizações medem o sucesso de suas LP por meio das vendas de seus produtos no mercado, ajudando a organização a determinar a aceitação de seus produtos no mercado. Porém, tais medidas na maioria das vezes não refletem a qualidade da LP e, em particular, não refletem diretamente a qualidade da arquitetura de LP. A facilidade com que a arquitetura pode evoluir é um dos fatores que determinam a qualidade da LP.

Uma organização precisa entender os fatores que afetam tanto o sucesso quanto a qualidade da arquitetura para garantir a capacidade da LP de estar apta às oportunidades de negócio futuras de maneira eficiente e efetiva de custo.

Uma arquitetura de LP pode ser afetada por vários fatores durante a sua manutenção, desenvolvimento e implantação. Os três maiores fatores que podem influenciar a qualidade da arquitetura de LP são:

- o ambiente de negócio da LP;
- o processo de software e as ferramentas usadas para criar e evoluir a arquitetura;
- a estrutura, o gerenciamento, as práticas e o pessoal da organização.

Contudo, os fatores organizacionais são tão difíceis de monitorar quanto os fatores ambientais e de processo. Todos os aspectos de manutenção e evolução da LP devem ser estudados para se entender o efeito de um fator organizacional. Os fatores organizacionais podem influenciar as mudanças em um cronograma de um projeto e este pode afetar a qualidade do código desenvolvido. Por exemplo, mudanças no prazo de entrega podem restringir o tempo disponível para teste.

Um grande problema em entender o impacto dos fatores organizacionais é que estes não são documentados em termos de como e quanto estes influenciam na qualidade de uma LP. Vários estudos de casos têm documentado os fatores organizacionais, porém estudos experimentais são necessários para relacionar os fatores organizacionais com processo de software e o ambiente de negócio.

Dois estudos de caso foram realizados em empresas australianas de desenvolvimento de software com o objetivo de mostrar como os fatores organizacionais afetam a qualidade e a manutenibilidade de uma LP. Além disso, é apresentado como o impacto dos fatores organizacionais variou com relação ao tamanho, ao estilo de gerenciamento e à volatilidade da LP.

Os fatores organizacionais são estudados segundo 04 perspectivas:

- **a estrutura organizacional:** a hierarquia e a estrutura das áreas envolvidas com a arquitetura de LP;
- **o domínio de negócio:** reputação e presença de mercado da LP;
- **as práticas de gerenciamento:** políticas e procedimentos na evolução da LP;
- **as características do pessoal:** perfil pessoal de quem dá manutenção ao software – conhecimento do domínio, experiência com LP e preocupação com a qualidade da arquitetura.

Essas perspectivas foram escolhidas, pois ajudam a entender a demanda do mercado, os procedimentos para a evolução, e como o pessoal trabalha na evolução dos produtos.

Cada fator organizacional inclui propriedades que podem ter impacto direto em como a manutenção é realizada, o que pode influenciar na qualidade do código. Estas propriedades e seus objetivos são apresentados na Tabela 1.

Tabela 1: Fatores Organizacionais e seus Objetivos (JAKTMAN, 1998).

Organisational Factors	Properties Studied
Organisational Structure	Objective: To determine the source of product-line decisions and architectural ownership. <ul style="list-style-type: none"> reporting structure and flow of communication; geographic location of people in the reporting structure; and communication with maintainers and developers for product-modifications, customer requests, and domain changes.
Business Domain	Objective: To determine the organisation's experience, knowledge and market presence in the product-line domain.
Management Practices	Objective: To determine management's approach to product quality, product maintenance, and project scheduling. <ul style="list-style-type: none"> the source of funding to pay for product-line development (ie. R&D money available or whether new product-line contracts are fixed price or cost-plus); deployment (marketing) of the product-line; focus on customer values (eg. safety, price, availability, functionality, reliability);
	<ul style="list-style-type: none"> customer support and feedback; software release schedule and future planning; focus on product and process quality practices (eg. how reusable assets are handled and managed).
Staffing Characteristics	Objective: To determine the profile of the staff and their ability to understand and modify the product-line. <ul style="list-style-type: none"> domain experience and customer usage knowledge mentoring, training reward structure, morale staff turnover use of contractors and consultants roles (ie. architect) quality focus

Os estudos de caso foram realizados em duas organizações, sendo que o primeiro foi realizado em 1997 – durou 3 meses, pois não possuía documentação da arquitetura de LP - em uma empresa de telecomunicação e o segundo em 1998 (3 semanas de duração) em uma empresa de telecomunicação multinacional. As empresas se diferenciaram pelo tamanho dos produtos e os estilos de gerenciamento.

As técnicas aplicadas para entender os fatores organizacionais foram: gráfico organizacional, diagrama de comunicação para o fluxo de informação entre a equipe de software e

gerenciamento e os entrevistados envolvidos com LP. A abordagem seguiu um ciclo de entrevistas, documentação, *feedback* e revisão pela equipe de software e gerentes.

A Tabela 2 apresenta o perfil organizacional de cada empresa.

Tabela 2: Perfil Organizacional das Empresas (JAKTMAN, 1998).

Case Study A	Case Study B
Profile: <ul style="list-style-type: none"> • Australian telecommunications manufacturer. • Founded: 1978 	<ul style="list-style-type: none"> • Multinational communications manufacturer • Founded in 1980.
Organisation Size <ul style="list-style-type: none"> • Australian Staff: 208 • R&D Staff Directly related to the Product Line: 10 • Maintenance Software Engineers: 2 	<ul style="list-style-type: none"> • Australian Staff within division: 400 • R&D Staff Directly related to the Product Line: 8 • Maintenance Software Engineers: 11
Product Line Size: <ul style="list-style-type: none"> • Product variants: 3 • Approx. 180K SLOC 	<ul style="list-style-type: none"> • Product variants: 6 • Approx. 180K SLOC
Product-line first released in: 1993	Product-line first released in: 1995

Os dados capturados a respeito da localização geográfica do centro de desenvolvimento de software e sua matriz em ambos os estudos de caso não afetaram a arquitetura.

Os resultados são apresentados na forma de tabelas de acordo com os fatores organizacionais (JAKTMAN, 1998). Por exemplo, no primeiro estudo de caso, o fator Características do Pessoal identificou que a mudança freqüente do líder da equipe de software resultou em: mudanças no projeto do software e decisões técnicas de implementação; mudanças nos prazos e prioridades; falta de tempo para as atividades de qualidade de software, entre outros.

Assim, os fatores organizacionais que mais contribuem para a qualidade da arquitetura identificados em ambos os estudos de caso foram:

- experiência organizacional com o domínio permite um planejamento para as mudanças de mercado;
- baixa rotatividade de pessoal significa manter o conhecimento da LP, permitindo mudanças no software melhor planejados e mais consistentes;
- consciência no entendimento da qualidade pelo gerente ajuda a planejar atividades de melhoria de software.

Os fatores que mais contribuem com o sucesso da arquitetura em ambos os estudos de caso foram:

- abordagem dirigida ao cliente significa entender as mudanças futuras e novos requisitos e planejar a evolução do produto;
- o tamanho da LP e a taxa de mudança podem ser uma indicação do sucesso da arquitetura.

Acredita-se que a limitação do trabalho está na dificuldade de relacionar as habilidades do pessoal (motivação e treinamento, por exemplo) com a qualidade dos produtos, uma vez que é difícil obter tais medidas.

5.2.5 Estudo 06 - Métricas de Utilização de Serviços para Avaliação da Estrutura de Arquiteturas de LP

Identificação:

Métricas de Utilização de Serviços para Avaliação da Estrutura de Arquiteturas de LP

Proponente(s):

VAN DER HOEK, A. (*Institute for Software Research – University of California - USA*)

DINCEL, E. e MEDVIDOVIC, N. (*University of Southern California - USA*)

Referenciado Por:

van der Hoek, A.; Dincel, E.; Medvidovic, N. 2003. *Using Service Utilization Metrics to Assess the Structure of Product Line Architectures*, Proceedings of the 9th International Software Metrics Symposium, pp. 298-308.

Características e Limitações:

Este trabalho foca a avaliação de qualidade de arquitetura de LP por meio de métricas para medir o atributo de qualidade chamado *structure soundness*. A estrutura criada pelos componentes forma a abstração central na qual todas as outras informações a respeito da arquitetura de LP se baseiam.

Existem algumas métricas como, por exemplo, *fan-in/fan-out* e complexidade ciclomática, mas são difíceis de serem aplicadas diretamente em arquiteturas de LP. Isto se deve por causa de duas características herdadas das arquiteturas de LP:

- **opcionalidade:** mesmo que alguns componentes formem o núcleo da arquitetura e estejam presentes em todos os membros da linha, outros componentes podem ou não ser incluídos em um produto final, dependendo, por exemplo, da preferência do cliente;
- **variabilidade:** existem variações no núcleo da arquitetura de LP que capturam alternativas lógicas como um conjunto de componentes dos quais somente um pode ser selecionado para fazer parte de um produto final.

Apesar de estas características serem teoricamente simples por natureza, elas violam as suposições críticas a respeito de métricas de software, nas quais o sistema a ser medido é estruturado de forma fixa.

Baseado no conceito de utilização de serviço, que mede o percentual de serviços fornecidos ou requeridos de um componente, as métricas propostas neste trabalho são capazes de identificar potenciais deficiências estruturais em arquiteturas de LP e guiar o arquiteto na avaliação de diferentes soluções para corrigir tais problemas. As métricas não revelam se a arquitetura está “boa” ou “ruim”, mas identificam problemas com base em valores relativos. Como exigem a interpretação humana, as métricas são informativas.

As métricas são aplicadas a três arquiteturas de LP: a primeira é uma arquitetura obtida por engenharia reversa de um curso da Universidade da Carolina do Sul; a segunda é uma arquitetura desenvolvida pelo grupo de estudo dos autores; e a terceira é uma arquitetura real fornecida de um estudo de caso de uma biblioteca.

As métricas baseiam-se no conceito de serviço, que inclui: métodos públicos ou funções, estruturas de dados de acesso direto, e qualquer outro tipo de recurso com acesso público. O termo serviço é utilizado para evitar a utilização das métricas em uma determinada ADL. Pelo contrário, fazer um mapeamento das construções das ADLs para o conceito de serviço permite aplicar as métricas em qualquer ADL em que uma arquitetura de LP possa ser modelada.

Assim, as métricas definidas são:

- $$PSU_X = \frac{P_{actual}}{P_{total}}$$

• **Provided Service Utilization (PSU):** definida como
- $$RSU_X = \frac{R_{actual}}{R_{total}}$$

• **Required Service Utilization (RSU):** definida como

P_{actual} é o número de serviços oferecidos por um componente X que é usado por outro componente da arquitetura e P_{total} é o número total de serviços oferecidos pelo componente X. R_{actual} e R_{total} tem significado análogo aos serviços requeridos.

As métricas RSU e PSU são dependentes de contexto. Um determinado componente pode ter valores diferentes de PSU e RSU dependendo da configuração particular da arquitetura da qual este faz parte. Por exemplo, considere a arquitetura da Figura 1a. O componente B fornece serviços ao componente A1 e requer serviços do componente C1. Dos 3 serviços fornecidos pelo componente B, somente 1 é utilizado pelo componente A1. Assim, $PSU_B = 1/3$. Devido ao componente C1 oferecer somente 1 dos 3 serviços requeridos pelo componente B, o $RSU_B = 1/3$.

Tais valores são baixos se comparados aos valores da arquitetura da Figura 1b, que são $PSU_B = 1$ e $RSU_B = 2/3$.

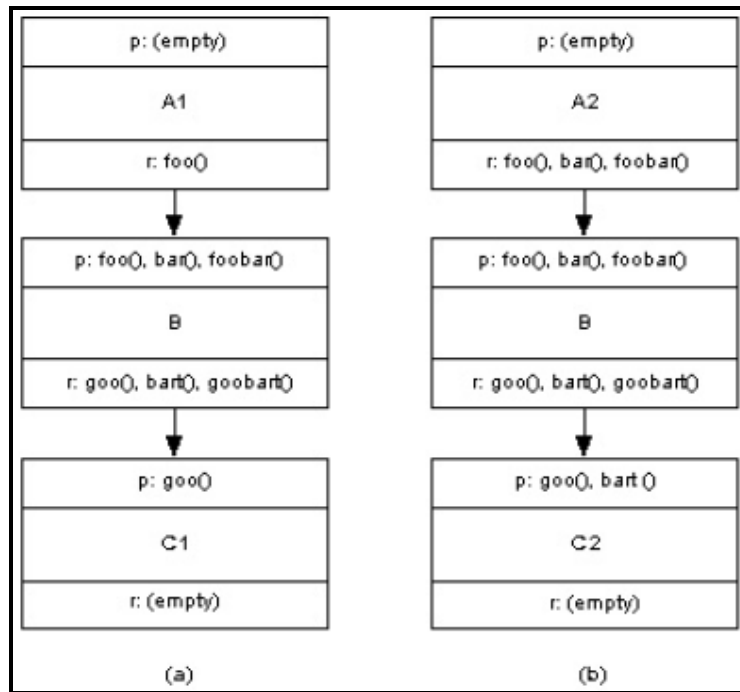


Figura 1: Exemplos de Arquiteturas (VAN DER HOEK; DINCEL; MEDVIDOVIC, 2003).

Apesar dos valores absolutos de PSU e RSU poderem indicar potenciais problemas – componente com RSU tendendo a 0 pode não funcionar bem na arquitetura; e componente com PSU tendendo a 0 significa um monte de funcionalidades extras que não são usadas por outros componentes - valores relativos são mais úteis ainda na comparação de duas arquiteturas relacionadas, mas diferentes. PSU_B e RSU_B , por exemplo, mostram que o componente B se encaixa melhor na arquitetura da Figura 1b do que na arquitetura da Figura 1a.

Para determinar o papel de cada componente na arquitetura, os valores de PSU e RSU devem ser calculados e analisados. Analisando os componentes da Figura 1, percebe-se que alguns possuem um conjunto vazio de serviços requeridos ou fornecidos.

As métricas PSU e RSU estão relacionadas com o tamanho do componente. Valores altos podem ser atingidos com componentes pequenos ou grandes, contanto que seus serviços requeridos e fornecidos sejam realmente usados.

Outro ponto importante a considerar é a análise da arquitetura como um todo em função de seus componentes e de suas métricas PSU e RSU. Para tanto, as métricas CPSU (*Compound PSU*) e CRSU (*Compound RSU*) são definidas como segue:

$$\boxed{\text{CPSU} = \frac{\sum_{i=1}^n P_{actual}^i}{\sum_{i=1}^n P_{total}^i} \quad \text{CRSU} = \frac{\sum_{i=1}^n R_{actual}^i}{\sum_{i=1}^n R_{total}^i}}$$

P_{actual} , P_{total} , R_{actual} e R_{total} são definidos da mesma forma que em PSU e RSU, enquanto n é o número de componentes da arquitetura. Nos exemplos da Figura 1a e Figura 1b, os valores de CPSU são, respectivamente 1/2 e 1, e os valores de CRSU são, respectivamente, 1/2 e 5/6.

As métricas CPSU e CRSU são usadas para avaliar a coesão interna de uma arquitetura. Valores próximos de 1 significa que a arquitetura é auto-contida, arquiteturas completamente funcionais. Valores baixos indicam uma arquitetura desbalanceada. Valor baixo de CPSU combinado com valor alto de CRSU implica que a arquitetura é mais ampla do que é necessário, ou seja, os componentes fornecem mais serviços do que realmente precisam para o funcionamento correto do sistema. Valor alto de CPSU com valor baixo de CRSU indica que a arquitetura está funcionalmente degradada, ou seja, são necessários novos componentes.

Com relação a arquiteturas de LP, a opcionalidade deve ser vista de acordo com: o próprio componente opcional; os outros componentes não-opcionais; e a arquitetura como um todo.

O cálculo de PSU e RSU para um componente opcional só é interessante se este fizer parte do produto final, pois pode ser tratado como um componente não-funcional. Porém, quando o componente é incluído, a análise dos componentes não-funcionais deve ocorrer para cada possibilidade de produtos finais. Por exemplo, a Figura 2 apresenta uma arquitetura com 2 componentes opcionais, D1 e D2. Assim, existem 4 possíveis produtos finais:

1. componentes D1 e D2 não são incluídos na arquitetura final do produto;
2. somente o componente D1 é incluído na arquitetura final do produto;
3. somente o componente D2 é incluído na arquitetura final do produto;
4. ambos os componentes D1 e D2 são incluídos na arquitetura final do produto;

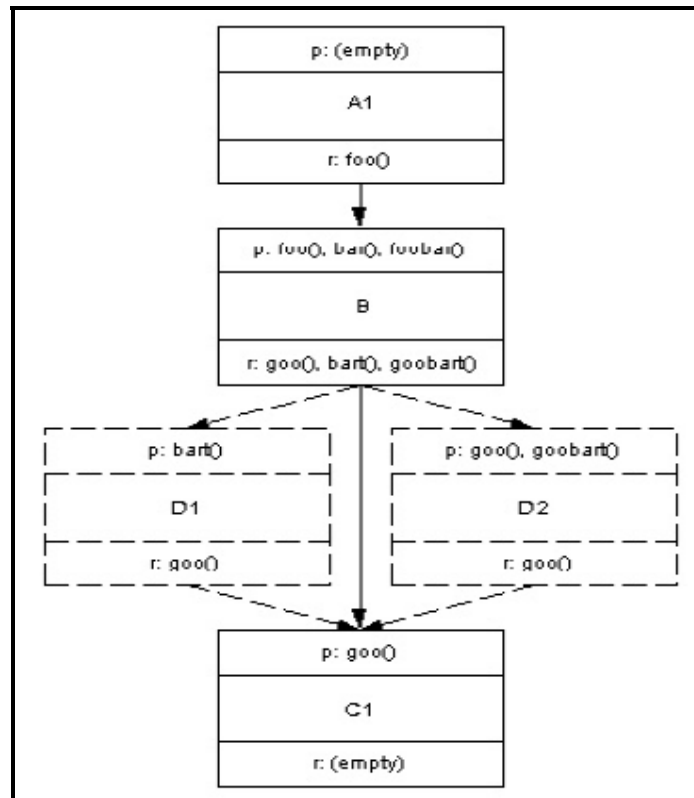


Figura 2: Exemplo de Arquitetura com Componentes Opcionais (VAN DER HOEK; DINCEL; MEDVIDOVIC, 2003).

Dessa forma, os valores PSU e RSU para cada componente não-opcional devem ser calculados para cada possibilidade de produto final. É possível, assim, analisar se a inclusão de cada componente e o impacto na arquitetura como um todo.

Com relação à arquitetura de LP, a variabilidade deve ser vista na perspectiva: do componente variante; de outros componentes; e da arquitetura como um todo.

A inclusão de um componente variante representa um desafio no cálculo de PSU e RSU dos componentes não-variantes. A Figura 3 apresenta um exemplo de arquitetura com componentes variantes. Da mesma forma, a análise de PSU e RSU deve ser feita para cada possibilidade.

Para analisar o impacto da variabilidade em toda a arquitetura é usado o valor médio de CPSU e CRSU. As médias destas métricas devem ser mantidas altas, indicando uma arquitetura altamente coesa.

Uma limitação é a falta de orientações (*guidelines*) absolutas para saber se uma arquitetura está estruturada de forma correta.

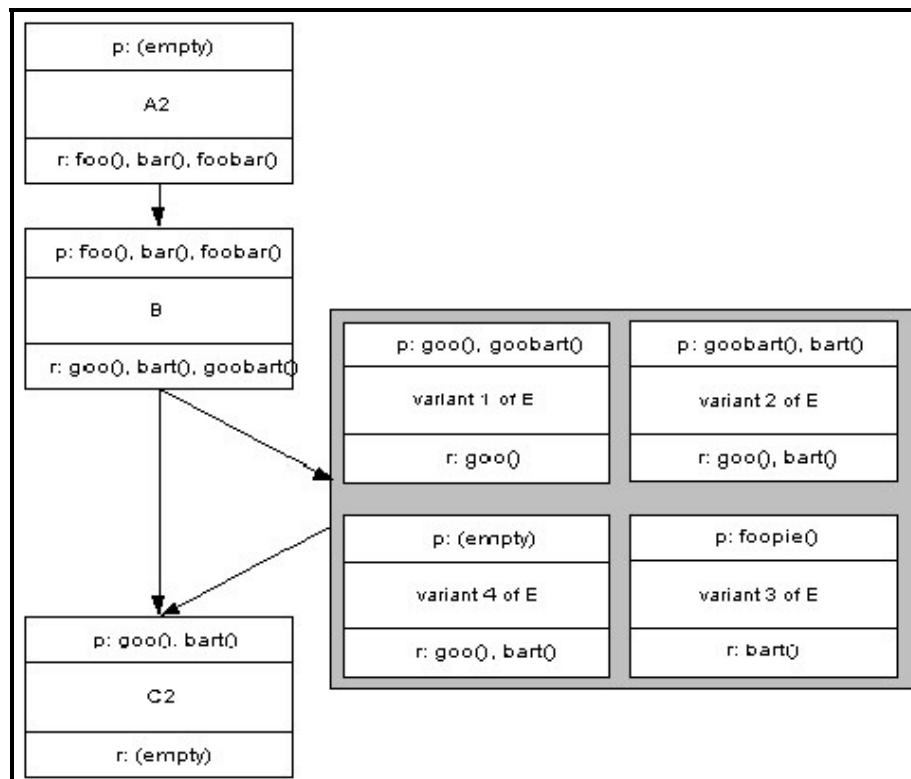


Figura 3: Arquitetura com Componentes Variantes (VAN DER HOEK; DINCEL; MEDVIDOVIC, 2003).

5.3 Estudos do Google Scholar

A extração dos dados dos textos do Google Scholar resultou nos itens apresentados a seguir.

5.3.1 Estudo 07 - Prometheus: Uma Abordagem para a Modelagem de Qualidade em LP

Identificação:

Prometheus: Uma Abordagem para a Modelagem de Qualidade em LP

Proponente(s):

TRENDOWICZ, A. e PUNTER, T. (*Fraunhofer Institute for Experimental Software Engineering - Germany*)

Referenciado Por:

Trendowicz, A; Punter, T. 2003. *Quality Modeling for Software Product Lines*, Proceedings of the 7th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering.

Características e Limitações:

O trabalho investiga qual o modelo existente que facilita alcançar a qualidade de LP em termos de requisitos não-funcionais. Assim, três requisitos para a modelagem de qualidade são definidos:

- **flexibilidade:** um modelo de qualidade deve ser flexível, por causa da dependência de contexto da qualidade. Existem vários contextos de qualidade:
 - **empresa:** inclui as características da empresa em que o modelo é usado. Um modelo de qualidade deve ser aplicável em diferentes empresas;
 - **projeto:** combina as características de um projeto em particular como o seu domínio ou diferentes visões de qualidade, representadas pelos diferentes *stakeholders*;
 - **processo:** reflete as características de um processo de desenvolvimento de software como sua estabilidade e disponibilidade para medir objetos em diferentes fases do processo.
- **reusabilidade:** modelos de qualidade devem permitir a reutilização de experiências de qualidade empacotadas em modelos de qualidade existentes em outros projetos. Dependendo do nível de similaridade dos projetos, o modelo permite a reutilização de dados de medidas, bem como as características de qualidade e suas relações;
- **transparência:** um modelo de qualidade deve fornecer a descrição de como certas características estão relacionadas a outras e como identificar sub-características. O modelo deve permitir que o gerente de projeto intervenha para modificá-lo se necessário.

Existem dois tipos de abordagem para modelar qualidade em LP:

- ***fixed-model:*** fornece um conjunto fixo de qualidades tal que a identificação de características específicas do cliente resulta em um subconjunto daquelas publicadas no modelo. Para medir as características de qualidade, as características, medidas e relações associadas com o modelo fixo são usadas. O problema de modelos fixos é que estes são limitados a dados quantitativos e relacionados ao produto.
- ***define-your-own-model:*** em contraste, as características não são definidas de forma fixa, mas com a ajuda do cliente um consenso entre as características de qualidade mais relevantes para um determinado sistema. Estas características são decompostas em características mensuráveis de qualidade e métricas relacionadas. Este tipo de modelo abrange tanto flexibilidade quanto transparência.

Apesar da existência de vários modelos de qualidade como ISO 9126 e o modelo de McCall, muitos deles dificilmente cobrem os requisitos de flexibilidade, reusabilidade e transparência.

Assim, o modelo SQUID pode ser utilizado, porém GQM deve ser usado para melhorar a fase de especificação do SQUID, assim como *Bayesian Belief Nets* pode ser usada para estimativas qualitativas e quantitativas.

Prometheus é uma abordagem de modelagem de qualidade e combina as características de vários modelos existentes com o objetivo de abranger os requisitos de flexibilidade, reusabilidade e transparência. A abordagem consiste de 3 fases:

- **especificação:** um modelo de qualidade é desenvolvido, conforme a Figura 1;
- **aplicação:** o modelo é aplicado para avaliar requisitos não-funcionais específicos;
- **empacotamento:** coleta as experiências adquiridas durante a aplicação do modelo para melhorá-lo este e ser reutilizado em outros projetos.

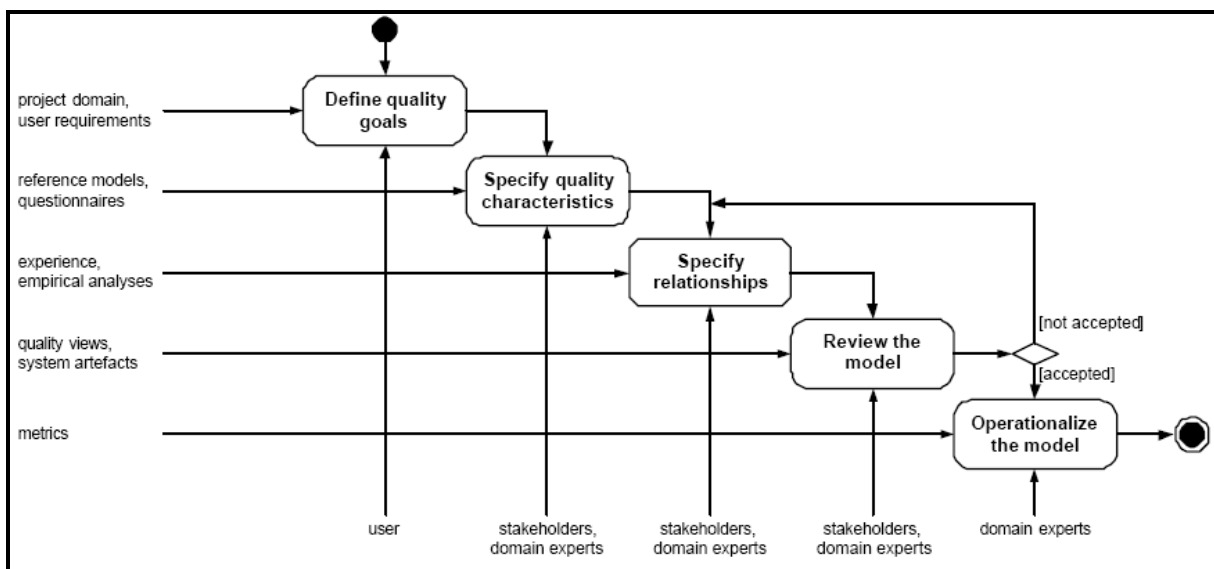


Figura 1: Atividades da Fase de Especificação da Abordagem Prometheus (TRENDOWICZ; PUNTER, 2003).

A fase de especificação consiste das seguintes atividades:

- **definição das metas de qualidade:** o usuário do sistema especifica as metas de qualidade com base em um *template* de metas de medição (Tabela 1);
- **especificação das características de qualidade:** refinamento das metas de qualidade em um conjunto de características e sub-características. Uma característica é mensurável quando é possível associar algum componente da LP e definir uma ou mais métricas. Entrevistas com quem tem conhecimento do domínio são realizadas;
- **especificação dos relacionamentos:** a importância relativa das características é determinado e contradições e redundâncias entre eles são detectadas. Dois tipos de relações devem ser distinguidos:

- **decomposição:** especifica como as características de alto nível podem ser decompostas em sub-características mais detalhadas;
- **influência:** define quais sub-características influenciam no valor de outras características.
- **revisão do modelo:** o modelo é revisado com relação a implementação;
- **operacionalização do modelo:** consiste na quantificação do modelo. Apesar do termo “quantificação”, as métricas associadas às características podem ter caráter quantitativo. Para combinar relacionamentos quantitativos com qualitativos são usadas *Bayesian Belief Nets* (BBN).

Tabela 1: *Template* de Metas de Medição (TRENDOWICZ; PUNTER, 2003).

Dimension	Definition	Example
Object	Which artefact is analysed?	<i>Analyze the software system</i>
Purpose	Why is the object analyzed?	<i>for the purpose of evaluation</i>
Quality Focus	Which characteristics of the object are analyzed?	<i>with respect to maintainability</i>
Viewpoint	Who will use the data collected?	<i>from the viewpoint of the system user</i>
Context	In which environment does the analysis take place?	<i>in the context of company X and development project Y</i>

Após a revisão de vários modelos de qualidade de software observa-se que nenhum deles abrange os requisitos não-funcionais de flexibilidade, reusabilidade e transparência para LP.

Com base nos modelos existentes, foi proposta a abordagem Prometeus para a modelagem de qualidade em LP. A abordagem é propícia às organizações que desejam realizar avaliações em estágios iniciais do desenvolvimento de LP, mesmo não tendo dados anteriores.

A limitação está na falta de detalhes em como analisar os atributos não-funcionais. O trabalho apresenta somente a abordagem de modo geral e as etapas da especificação, mas não entra em detalhes de como realizar a avaliação usando a abordagem proposta.

5.3.2 Estudo 08 - O Modelo BAPO para Avaliação de LP

Identificação:

O Modelo BAPO para Avaliação de LP

Proponente(s):

VAN DER LINDEN, F. (*Philips Medical Systems*)

BOSCH, J. (*University of Groningen*)

KAMSTIES, E. (*University of Duisburg-Essen*)

KÄNSÄLÄ, K. (*Nokia Research Center*)

OBBINK, H. (*Philips Medical Systems*)

Referenciado Por:

van der Linden, F.; Bosch, J.; Kamsties, E.; Käsälä, K.; Obbink, H. 2004. *Software Product Family Evaluation*, Proceedings of the 3rd International Conference on Software Product Lines, pp. 110-129.

Niemelä, E.; Martinassi, M.; Taulavouri, A. 2004. *Practical Evaluation of Software Product Family Architectures*, Proceedings of the 3rd International Conference on Software Product Lines, pp. 130-145.

Características e Limitações:

Neste trabalho um *framework* de avaliação de LP é apresentado, bem como um estudo de caso.

Quatro interesses no desenvolvimento de software são identificados (BAPO), sendo eles:

- **Business:** como se beneficiar de seus produtos;
- **Architecture:** meios técnicos de construir software;
- **Process:** cargos, responsabilidades e relacionamentos no desenvolvimento de software;
- **Organisation:** mapeamento de cargos e responsabilidades da estrutura organizacional.

A Figura 1 apresenta o relacionamento de cada um dos interesses.

Cada organização terá um nível de avaliação diferente para cada interesse, uma vez que estes são independentes.

Por meio do *framework* BAPO, a avaliação coleta e estrutura as características da unidade de produção, divisão ou companhia de software. O propósito do modelo é servir como um *benchmark* efetivo de LP; apoiar a avaliação de LP em termos das unidades, divisões ou

companhias de desenvolvimento de software; e apoiar a melhoria de LP, que envolve a avaliação e melhorias de planos.

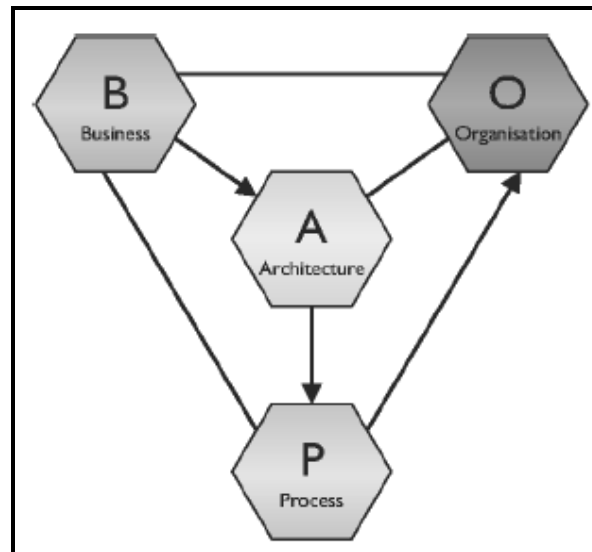


Figura 1: Interesses BAPO.

Assim, quatro dimensões de avaliação são desejáveis: *Business*, *Architecture*, *Process* e *Organisation*.

A **dimensão de negócio** (*business*) trata da habilidade de uma organização em gerenciar, prever e estimar os custos e os benefícios do desenvolvimento. O custo depende da arquitetura e da organização escolhidas. A principal preocupação está na forma como a organização pode determinar o custo de uma LP.

Uma das questões a considerar na dimensão de negócio é o escopo, ou seja, quais produtos devem estar sujeitos à LP baseado nas expectativas do mercado. No *framework* de avaliação, 4 aspectos são considerados, sendo eles: identidade, visão, objetivos e planejamento estratégico. Estes aspectos são parcialmente dependentes uns dos outros.

Na **dimensão de arquitetura** (*architecture*) é essencial detectar, projetar e modelar as partes variáveis e comuns da LP. Alguns aspectos são considerados na dimensão de arquitetura, sendo eles: a própria arquitetura, a qualidade dos produtos, os níveis de reutilização, e o gerenciamento de variabilidades.

A **dimensão de processo** (*process*) enfatiza os cargos, os produtos, e as responsabilidades e relacionamentos do desenvolvimento de software. Nessa dimensão, CMMI é utilizado para avaliar o processo na abordagem BAPO.

A **dimensão organizacional** (*organisational*) trata a maneira com que a organização está apta a lidar com as relações e responsabilidades complexas. Alguns aspectos influenciam a

avaliação da organização, sendo eles: a distribuição geográfica, a cultura, os cargos e responsabilidades, e o ciclo de vida dos produtos.

Um estudo de caso é apresentado para ilustrar a aplicação da abordagem BAPO em scanners de ressonância em tecidos humanos.

A abordagem BAPO apresenta melhorias com relação ao modelo de van der Linden et al. (2004), uma vez que existem quatro grupos de interesse (BAPO). Uma organização pode usar o *framework* para saber se possui um perfil adequado ao tipo de produtos que desenvolve. No caso de não possui o perfil adequado, o *framework* ajuda na escolha das ações a serem tomadas para a sua adequação.

A limitação do trabalho é a falta de detalhes na apresentação das dimensões e como realmente avaliar cada uma delas. Falta uma descrição mais clara de cada fase da abordagem BAPO, pois está descrito em muito alto nível comparado com outras abordagens e modelos.

5.3.3 Estudo 09 - Avaliação Prática de Arquiteturas de LP

Identificação:

Avaliação Prática de Arquiteturas de LP

Proponente(s):

NIEMELÄ, E.; MATINLASSI, M.; TAULAVUORI, A. (*VTT Technical Research Centre of Finland*)

Referenciado Por:

Niemelä, E.; Martinlassi, M.; Taulavouri, A. 2004. *Practical Evaluation of Software Product Family Architectures*, Proceedings of the 3rd International Conference on Software Product Lines, pp. 130-145.

Características e Limitações:

O trabalho investiga os interesses relacionados à avaliação de arquiteturas de LP, e apresenta um método de avaliação que usa informações empíricas coletadas de empresas por meio de entrevistas estruturais ao longo de revisões de artefatos da arquitetura de LP.

Os critérios de avaliação de LP propostos por Niemelä e Ihme (2003) representam os elementos categorizados a serem investigados nas práticas de adoção de LP, bem como estão baseados no conhecimento obtido de empresas por meio de um conjunto de entrevistas e do desenvolvimento de várias arquiteturas baseadas em componentes para sistemas embarcados.

A abordagem BAPO (BAN DER LINDEL et al. 2004) identifica as relações lógicas entre negócio e arquitetura, e arquitetura e processo. A abordagem FEF dá suporte às quatro dimensões de BAPO e sugere cinco níveis arquiteturais.

O método FAE (*Family Architecture Evaluation*) foi desenvolvido para ser usado por empresas e arquitetos de LP que desejam ter um *benchmark* de sua arquitetura de LP. Ele define como usar as abordagens citadas na avaliação de arquitetura de LP.

O método FAE inclui quatro passos:

- **Framework de Avaliação para Entrevistas:** empresas que adotam LP nos mais variados contextos são escolhidas para as entrevistas. As questões são avaliadas com base nos critérios de avaliação de Niemelä e Ihme (2003). O número de questões é limitado ao tempo de 3 horas. Os critérios são usados, pois o *framework* suporta a classificação das questões em 3 categorias: negócio, domínio e tecnologia relacionadas ao desenvolvimento e manutenção de arquiteturas de LP;
- **Revisão/Inspeção dos Artefatos da Arquitetura:** entender o tipo de abordagem de LP que a empresa adota. Se a arquitetura é documentada e encontra-se atualizada, descrições arquiteturais fornecem a melhor fonte para a sua avaliação. Dependendo do tamanho da arquitetura, a revisão/inspeção pode levar duas semanas;
- **Análise dos Resultados:** os resultados das entrevistas são coletados e reformulados para um *framework* de comparação, que foi criado para reorganizar as questões do *framework* de entrevistas de tal forma que eles vão de encontro aos aspectos da dimensão de arquitetura da abordagem BAPO;
- **Framework de Avaliação para a Realização de Benchmark:** baseado no *framework* de comparação e nos níveis e aspectos definidos na dimensão arquitetural do *framework* de avaliação de arquitetura, um *framework* de avaliação foi construído (Tabela 1).

Tabela 1: *Framework* de Avaliação de Arquiteturas de LP (NIEMELÄ; MATINLASSI; TAULAVUORI, 2004).

Relation/PFA	Aspect	Item
Business to PFA	Domain	<ul style="list-style-type: none"> • Driving force: technology, application, both • Main functional properties • Quality requirements • Change frequency in a domain • Life-cycle or customer-specific features • Trend in customer needs
	Openness	<ul style="list-style-type: none"> • Number of third-party components • Number of subcontractors/component providers
	Variability	<ul style="list-style-type: none"> • Number of members in a product family • Different parts; %/features; number of functional subsystems/components, %/total software • Reasons for differentiation • Assembling is based on existing components, (re)used/modified and new components • Variable quality requirements • Variability management • Customized third-party components
PFA	Context	<ul style="list-style-type: none"> • Identified domains • Reasons for their importance • Visibility of domains in the PFA
	Domain knowledge	<ul style="list-style-type: none"> • Common architecture for family members • Platforms • Shared components
	Variability	<ul style="list-style-type: none"> • Common properties • Binding time
	Core assets	<ul style="list-style-type: none"> • What are they? • Architectural styles and patterns • Design patterns • Component model • Appropriateness of the component model • Separation of hardware-related variability
PFA to process		<ul style="list-style-type: none"> • Methods in architecting • Modeling languages • Validation practices • Selection of third-party components • Stakeholders of the PFA • Guidelines • Guidelines in practice
PFA to organization		<ul style="list-style-type: none"> • Number of domain analysts • Applied standards • Open standards • Constraints set by other authorities • PFE familiarity • Model-driven architecture (MDA) familiarity • Effort used for sharing domain/design knowledge • Importance of documentation in knowledge sharing

Três estudos de caso são apresentados para a dimensão arquitetural de FEF. Alguns pontos são considerados na realização dos estudos de caso: o negócio das empresas entrevistadas, pré-condições para a arquitetura de LP, o conhecimento do domínio e as variabilidades, e a ligação entre arquitetura e o processo e a organização.

Alguns pontos considerados na discussão a respeito dos resultados dos estudos de caso: o tamanho das empresas, os cargos de negócio, e o tipo dos produtos. Duas empresas eram pequenas e a outra era média. O tamanho da empresa mostrou-se ter grande influência em como as relações da arquitetura para negócio, processo e organização são gerenciadas. Assim, a abordagem FEF categoriza as relações entre arquitetura e as outras dimensões de acordo com o tamanho da empresa (por exemplo, pequena, média, entre outros).

5.4 Análise dos Resultados

Com base nos dados extraídos foi possível realizar uma análise com relação aos estudos selecionados considerando:

- as métricas estabelecidas no protocolo da revisão sistemática;
- a relação, por fonte, entre a quantidade de estudos primários encontrados no processo de busca e quantidade de estudos primários da seleção preliminar;
- a porcentagem, por fonte, entre a quantidade de estudos primários encontrados e o total de estudos primários.

Com relação às métricas definidas no protocolo de revisão sistemática, estas foram medidas e os seus valores são apresentados:

- $\sum \text{LP}$ = Quantidade de estudos selecionados sobre qualidade/avaliação de LP como um todo (incluindo arquitetura de LP) = **03**
- $\sum \text{Arq}$ = Quantidade de estudos selecionados somente sobre qualidade/avaliação de arquitetura de LP = **06**

A seguir são apresentados os valores de tais métricas na forma de um gráfico de pizza. O gráfico mostra que 67% dos textos analisados nesta revisão sistemática estão focados na avaliação de qualidade de arquitetura de LP. Acredita-se que isto se deva a três fatores:

- o primeiro por considerar a abordagem de LP relativamente recente aplicada ao contexto de software;

- o segundo pela arquitetura ser o artefato mais importante de uma LP, bem como a sua avaliação implicar na avaliação do conjunto de todas as possíveis arquiteturas de produtos gerados a partir da LP;
- o terceiro justificado pelo estado da arte, comprovado por esta revisão sistemática.

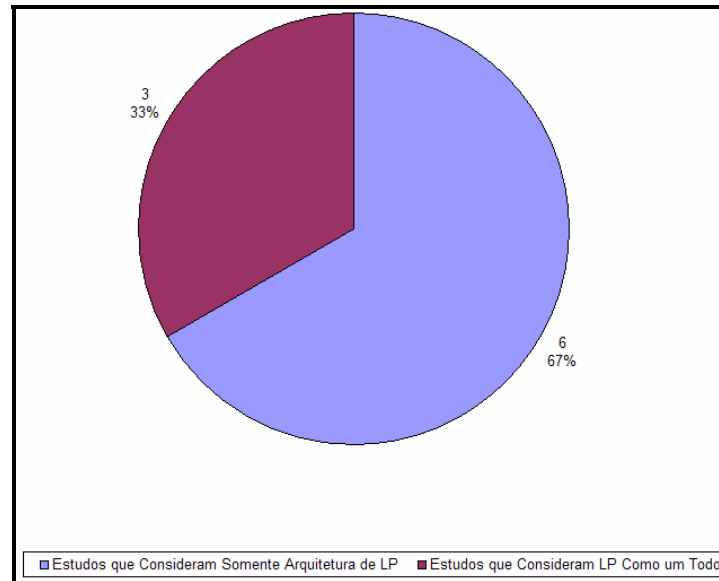


Figura 7: Valores Obtidos para as Métricas $\sum LP$ e $\sum Arq$.

Por estas razões, acredita-se que a avaliação de arquiteturas de LP seja um tema ainda em aberto e com a possibilidade de realização de vários trabalhos.

Outro ponto importante a ser analisado são as relações entre as fontes de busca dos estudos primários e a quantidade de estudos encontrados e selecionados. A Tabela 1 apresenta os números relacionados às fontes e aos estudos primários desta revisão sistemática.

Tabela 1: Números sobre os Estudos Primários e as Fontes de Busca.

Fonte	Estudos Primários Encontrados (Processo de Busca)	Estudos Primários Selecionados (Seleção Preliminar)	Porcentagem de Estudos Selecionados	Porcentagem de Estudos Encontrados
ACM	25	01	4%	29,8%
IEEE	49	05	10,2%	58,3%
Google Scholar	10	03	30%	11,9%
	84	09	10,7%	100%

Com base na Tabela 1 é possível analisar algumas relações. Uma delas é a relação entre o número de estudos primários encontrados nas fontes, durante o processo de busca, e o número de estudos primários selecionados, durante o processo de seleção preliminar. O gráfico da Figura 8 apresenta esta relação para cada fonte de busca.

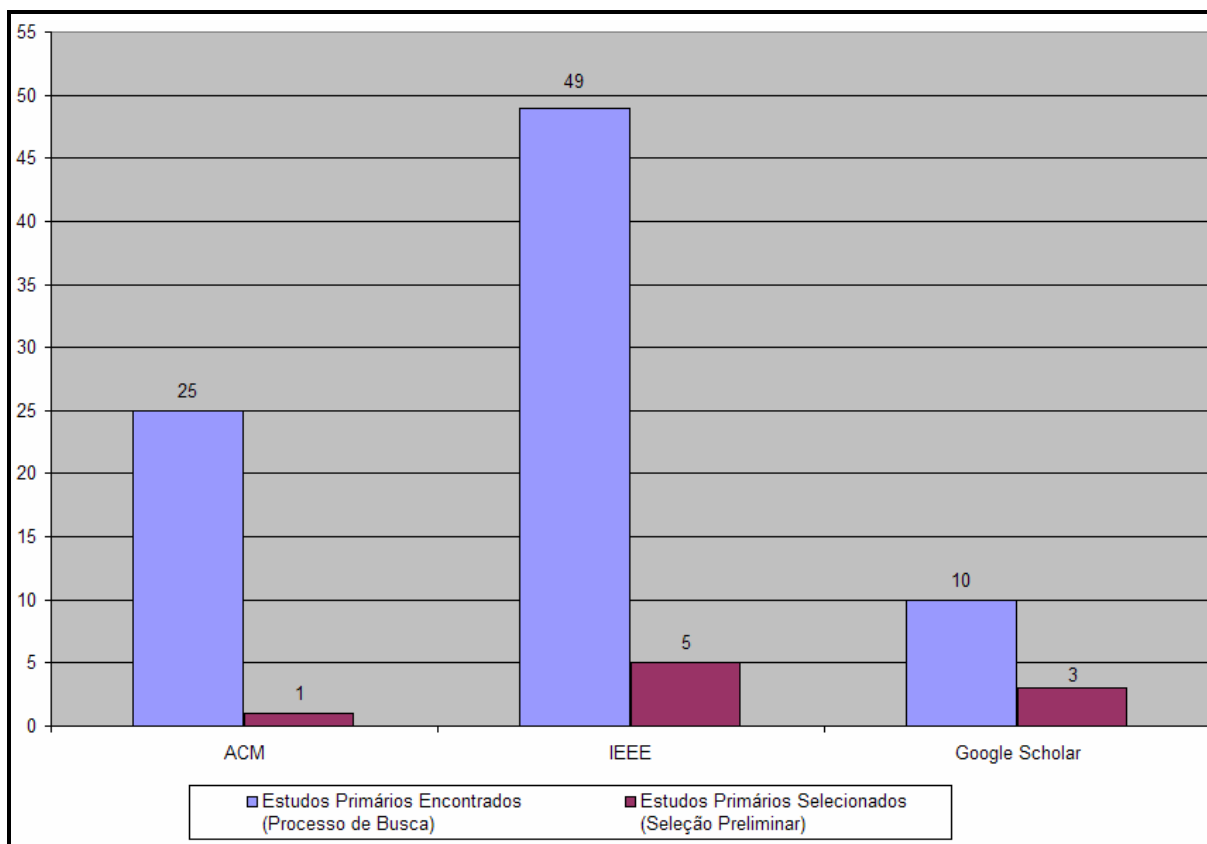


Figura 8: Gráfico da Relação entre Estudos Primários Encontrados e Selecionados.

Na Figura 8 pode-se perceber que a fonte ACM resultou em 25 estudos durante o processo de busca, porém somente 01 estudo foi selecionado para leitura completa, o que significa que somente 4% dos estudos encontrados foram selecionados. A fonte IEEE resultou em 49 estudos durante o processo de busca, mas somente 05 estudos foram selecionados, o que significa que somente 10,2% dos estudos foram selecionados. Por último percebe-se que a fonte Google Scholar obteve a melhor porcentagem em termos de estudos selecionados, pois tal fonte resultou em 10 estudos, sendo que 03 foram selecionados (30%).

Outra relação interessante é a quantidade total de estudos primários encontrados e a quantidade de estudos encontrados, para cada fonte de busca, ou seja, a porcentagem total de estudos encontrados por cada fonte de busca. Nota-se no gráfico que a fonte de busca que teve um melhor aproveitamento em relação ao número de estudos encontrados foi a IEEE. Isto se deve ao fato do amplo acervo eletrônico especializado na área.

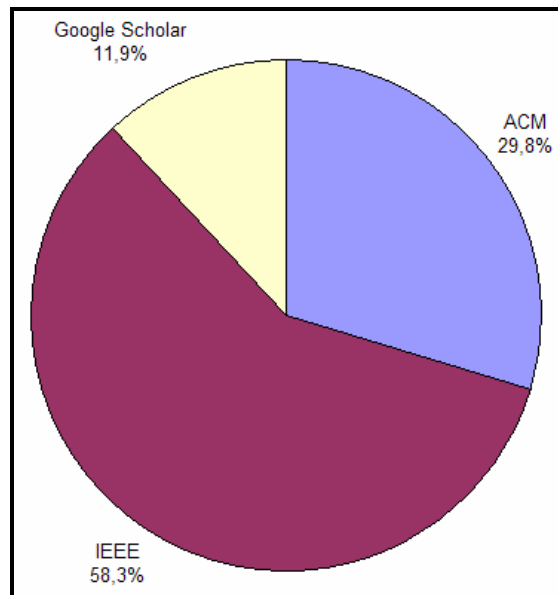


Figura 9: Porcentagem de Estudos Seleccionados por Fonte de Busca.

Com relação aos estudos seleccionados e lidos por completo é possível, a partir desta revisão sistemática, mostrar indícios de que atualmente o estado da arte em avaliação de qualidade de LP concentra-se em modelos e abordagens de descrição de arquitetura de LP, bem como a identificação de atributos de qualidade (portabilidade, manutenibilidade, flexibilidade, reusabilidade e transparência) e métricas para a medição destes atributos. A grande dificuldade em avaliação de arquitetura de LP é a necessidade de abstração de todo o conjunto de arquiteturas dos produtos gerados por uma LP. Tal conjunto depende da composição das estruturas opcionais e variantes da LP como os componentes, por exemplo.

O que se percebe é que existem várias abordagens com o propósito de avaliar qualidade em LP, ilustradas com vários estudos de caso, porém tais propostas ainda deixam a desejar com relação à sua aplicação prática.

Abordagens propostas para a avaliação de LP como um todo fazem uso de padrões de qualidade de software já consolidados na literatura como, por exemplo, a utilização da ISO 15498 combinada com a abordagem *Goal-Question-Metric*. Todas estas abordagens baseiam-se em modelo de qualidade de propósito específico ao contexto de LP. Algumas fazem uma adaptação de modelos já existentes, enquanto outras abordagens propõem seus próprios modelos. A escassez de métricas de LP é evidente com base nos estudos analisados. Métricas convencionais não podem ser aplicadas na abordagem de LP, por causa da estrutura flexível de suas arquiteturas. A opcionalidade e a variabilidade tornam a avaliação de LP um desafio para a comunidade de engenharia de software.

Um ponto interessante é que nenhum trabalho analisado nesta revisão sistemática abordou a realização de experimentos com vistas à avaliação de LP, o que se acredita ser um ponto fundamental na proposta e comprovação de novos modelos e abordagens como as apresentadas neste trabalho.

Dessa forma, acredita-se que vários trabalhos no sentido de avaliar LP e suas arquiteturas podem ainda ser desenvolvidos, principalmente no que se refere à realização de estudos experimentais para a avaliação de LP.

6. Considerações Finais sobre a Revisão Sistemática

A realização desta revisão sistemática mostrou algumas dificuldades com relação aos mecanismos de buscas disponíveis em cada uma das fontes de pesquisa.

O mecanismo da ACM, por exemplo, é restrito quanto ao tamanho da string de busca. Para contornar este problema é necessário quebrar a string de busca principal em várias sub-strings. Não foi o caso desta revisão sistemática, mas isso pode prejudicar a busca por estudos na biblioteca digital da ACM.

Outro mecanismo que também apresentou alguma dificuldade foi o do Google Scholar, pois este não permite a busca no resumo dos trabalhos. A busca pode ser feita ou no corpo todo do texto ou pelo título do trabalho. Isto é ruim, pois o Google possui um acervo muito grande de estudos científicos, além de indexar outras fontes de busca.

No caso particular desta revisão sistemática, a fonte SpringerLink não aceitou a busca por resumo, por causa de um erro interno no algoritmo de busca que rejeitou o termo “su” relativo ao resumo dos estudos (do inglês *summary*).

Com relação às 03 fontes de busca utilizadas e no trabalho desenvolvido, acredita-se que a fonte mais adequada para a realização de uma revisão sistemática seja a IEEE, levando em consideração a qualidade dos artigos encontrados - periódicos e anais de eventos internacionais muito bem conceituados - e a facilidade de uso do mecanismo de busca e de suas propriedades como, por exemplo, a string de busca.

Acredita-se que a realização desta revisão sistemática possa apoiar a comunidade acadêmica que vem trabalhando para a melhoria continua da abordagem de LP, bem como a indústria que, por meio de empresas comprometidas com os seus clientes e com a qualidade de seus produtos, adota tais abordagens e contribuem com projetos reais e relatos sobre a adoção e desenvolvimento de LP.

Referências

- Biolchini, J.; Mian, P. G.; Natali, A. C. C.; Travassos, G. H. 2005. *Systematic Reviews in Software Engineering*, Technical Report ES 679/05, 31 p.
- Jaktman, C. B. 1998. *The Influence of Organisational Factors on the Success and Quality of a Product-line Architecture*, Proceedings of the Software Engineering Conference, pp. 02-11.
- Knodel, J.; John, I.; Dharmalingam, G.; Pinzger, M.; Arciniegas, A. L.; Riva, C. 2005. *Asset Recovery and their Incorporation into Product Lines*, Proceedings of the 12th Workshop Conference on Reverse Engineering, pp. 120-129.
- Matinlassi, M. *Evaluating the Portability and Maintainability of Software Product Family Architecture: Terminal Software Case Study*, Proceedings of the 4th Working IEEE/IFIP Conference on Software Architecture, pp. 295-298.
- Niemelä, E., Ihme, T. 2001. *Product Line Software Engineering of Embedded Systems*. ACM SIGSOFT Software Engineering Notes, Vol. 26, No 3 (May 2001). pp. 118-125.
- Schmid, K. 2001a. *An Assessment Approach to Analyzing Benefits and Risks of Product Lines*, Proceedings of the 25th International Computer Software and Applications Conference on Invigorating Software Development, pp. 525-530.
- Schmid, K. 2001b. *A Framework for Product Line Quality Model Development: The PuLSE-Eco Meta Quality Model*, Technical Report No. 047.00/E, 41 p.
- Trendowicz, A; Punter, T. 2003. *Quality Modeling for Software Product Lines*, Proceedings of the 7th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering.
- van der Hoek, A.; Dincel, E.; Medvidovic, N. 2003. *Using Service Utilization Metrics to Assess the Structure of Product Line Architectures*, Proceedings of the 9th International Software Metrics Symposium, pp. 298-308.
- van der Linden, F.; Bosch, J.; Kamsties, E.; Käsälä, K.; Obbink, H. 2004. *Software Product Family Evaluation*, Proceedings of the 3rd International Conference on Software Product Lines, pp. 110-129.