



ENGENHARIA E GERENCIAMENTO DE ECOSSISTEMAS DE SOFTWARE

Rodrigo Pereira dos Santos

Exame de Qualificação apresentado ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Sistemas e Computação.

Orientadora: Cláudia Maria Lima Werner

Rio de Janeiro

Março de 2013

Índice

Índice de Figuras.....	i
Índice de Tabelas	iv
Capítulo 1 – Introdução.....	1
1.1 Preâmbulo	1
1.2 Motivação	3
1.3 Problema	6
1.4 Contexto da Proposta de Tese	8
1.5 Objetivos e Visão Geral da Proposta.....	16
1.6 Organização	19
Capítulo 2 – Técnicas e Ambientes para Reutilização de Software	20
2.1 Introdução	20
2.2 Engenharia de Domínio	22
2.2.1. Linhas de Produtos de Software	23
2.2.2. Ciclo de Desenvolvimento	24
2.2.3. Modelo de Características e Variabilidade	25
2.2.4. Engenharia de Domínio e Ontologias	30
2.3 Desenvolvimento Baseado em Componentes	32
2.3.1. Classificação.....	33
2.3.2. Repositórios.....	34
2.3.3. Arquiteturas, Estilos e Padrões.....	36
2.3.4. Mercados.....	37
2.4 Critérios para Apoiar o Desenvolvimento Baseado em Componentes no Contexto da Engenharia de Domínio.....	39
2.5 Biblioteca Brechó	40
2.6 Considerações Finais	44
Capítulo 3 – A Evolução da Indústria Orientada aos Ecossistemas de Software.....	47
3.1 Introdução	47

3.2	Da Engenharia de Software aos Ecossistemas: As “4 Gerações” da Reutilização.....	49
3.2.1.	Primeira Geração: Sistemas Monolíticos.....	52
3.2.2.	Segunda Geração: Sistemas Baseados em Componentes	53
3.2.3.	Terceira Geração: Sistemas Baseados em Linhas de Produtos	54
3.2.4.	Quarta Geração: Sistemas Baseados em Ecossistemas	55
3.3	Visão Geral de Ecossistemas de Software	57
3.3.1.	Terminologia e Definição	60
3.3.2.	Estruturas de Classificação	63
3.3.3.	Características	75
3.3.4.	Benefícios e Dificuldades	76
3.3.5.	Sinais de Teorização	77
3.3.6.	Impactos na Engenharia de Software	78
3.3.7.	Grupos e Tópicos de Pesquisa.....	80
3.4	Desafios em Ecossistemas de Software	81
3.4.1.	Desafios 2009-2010.....	82
3.4.2.	Desafios 2011-2012.....	82
3.4.3.	Desafios sob o Ponto de Vista Nacional	84
3.5	Considerações Finais	84
Capítulo 4 –	Uma Abordagem para Engenharia e Gerenciamento de Ecossistemas de Software Baseada em Reutilização	87
4.1	Introdução	87
4.2	Um <i>Framework</i> para Engenharia e Gerenciamento de Ecossistemas de Software	91
4.2.1.	Dimensão #1: Técnica	93
4.2.2.	Dimensão #2: Transacional.....	106
4.2.3.	Dimensão #3: Social	113
4.3	Proposta de Ecossistemas de Software Baseada em Reutilização	120
4.3.1.	Domínios em Ecossistemas de Software	121
4.3.2.	Componentes em Ecossistemas de Software	123

4.3.3. Definição de Ecossistemas em Engenharia de Software	126
4.3.4. Requisitos da Abordagem <i>ReuseSEEM</i>	128
4.3.5. Infraestrutura da Solução	129
4.3.6. Projeto e Implementação.....	133
4.4 Trabalhos Relacionados	135
4.5 Considerações Finais	140
Capítulo 5 – Conclusão	143
5.1 Contribuições	143
5.2 Metodologia da Pesquisa	145
5.2.1. Atividades Realizadas.....	145
5.2.2. Atividades Previstas.....	146
5.3 Resultados Preliminares.....	148
5.4 Resultados Previstos	149
5.5 Cronograma Proposto	150
Referências Bibliográficas	151

Índice de Figuras

Figura 1.1 – Cenários da Reutilização de Software em direção aos ECOSs	9
Figura 1.2 – Ambiente de ECOSs baseado em reutilização.....	12
Figura 1.3 – Estratégia de pesquisa, estendida originalmente de (SPÍNOLA <i>et al.</i> , 2010) e adaptada de (SANTOS <i>et al.</i> , 2011; 2011d).	14
Figura 1.4 – Processo da abordagem <i>ReuseSEEM</i>	18
Figura 2.1 – Atividades para o desenvolvimento da LPS. Fonte: (NORTHROP, 2004)	24
Figura 2.2 – Atividades do ciclo de desenvolvimento de uma LPS. Fonte: (FERNANDES, 2009).....	25
Figura 2.3 – Exemplo de modelo de características na notação Odyssey-FEX. Fonte: (FERNANDES, 2009).....	30
Figura 2.4 – Processo de produção-gerência-consumo em DBC. Fonte: (ALMEIDA <i>et al.</i> , 2007).....	35
Figura 2.5 – Tela inicial da biblioteca Brechó. Fonte: (SANTOS, 2010).....	41
Figura 2.6 – Organização interna da biblioteca <i>Brechó</i> . Fonte: (MARINHO <i>et al.</i> , 2009b).....	42
Figura 3.1 – Trajetória da Reutilização de Software em “4 gerações”	51
Figura 3.2 – Distribuição das publicações sobre ECOSs ao longo dos últimos anos, extraído de dois estudos secundários. Fontes: (BARBOSA <i>et al.</i> , 2013) e (MANIKAS & HANSEN, 2013).....	60
Figura 3.3 – Níveis de escopo dos ECOSs, onde OS = organização de software; FS = fornecedor de software e C = cliente. Fonte: (JANSEN <i>et al.</i> , 2009a).....	66
Figura 3.4 – Perspectivas dos ECOSs. Fonte: (JANSEN <i>et al.</i> , 2009b).....	67
Figura 3.5 – Visão do ECOSs em três dimensões. Fonte: (CAMPBELL & AHMED, 2010).....	68
Figura 3.6 – Ciclo de vida social dos ECOSs. Fonte: (SANTOS <i>et al.</i> , 2012a)	70
Figura 3.7 – Fontes de inovação fechada em um sistema tradicional, onde as linhas pontilhadas são relações que não estão estabelecidas claramente, dado que a organização de software centraliza as relações. Fonte: (SANTOS <i>et al.</i> , 2013b) ..	73

Figura 3.8 – Fontes de inovação aberta em um ECOSs, incorporando aspectos do ciclo de vida social (SANTOS <i>et al.</i> , 2012a). Fonte: (SANTOS <i>et al.</i> , 2013b).....	74
Figura 3.9 – Tópicos de pesquisa mais investigados em ECOS. Fonte: (BARBOSA & ALVES, 2011)	80
Figura 3.10 – Tendências de ECOSs na área de ES. Fonte: (WERNER <i>et al.</i> , 2012)....	86
Figura 3.11 – Desafios em ECOSs enquanto área de pesquisa. Fonte: (BARBOSA <i>et al.</i> , 2013).....	86
Figura 4.1 – <i>Framework</i> “3+1” para E&G de ECOSs. Fonte: (WERNER <i>et al.</i> , 2012) 93	
Figura 4.2 – Dimensão técnica do <i>ReuseECOS</i> . Fonte: (SANTOS & WERNER, 2011b)	95
Figura 4.3 – Número de linhas de código (LoC) em <i>commits</i> SVN na plataforma do ECOS Brechó, extraído da ferramenta StatSVN. Fonte: (SANTOS & WERNER, 2011b).....	97
Figura 4.4 – LoC por mudança realizada (superior) e contribuição de LoC por desenvolvedor da plataforma do ECOS Brechó (inferior). Fonte: (SANTOS & WERNER, 2011b).....	103
Figura 4.5 – Tipo de atividade (modificação/correção) (superior) e taxa de atividade por hora por dia, na plataforma do ECOS Brechó. Fonte: (SANTOS & WERNER, 2011b).....	105
Figura 4.6 – Dimensão transacional do <i>ReuseECOS</i> . Fonte: (SANTOS & WERNER, 2011d).....	107
Figura 4.7 – Cadeia de valor da indústria de software (externa) e do mercado/ambiente de DBC (interno) implementada na Brechó-VCM, analisada e marcada com elementos de gerenciamento de ECOSs. Os retângulos representam os papéis dos agentes e as setas, as suas ações. Nos retângulos mais claros, estão os elementos que adicionados pela proposta de SANTOS (2010). Fonte: (SANTOS & WERNER, 2011d).....	108
Figura 4.8 – Arquitetura da plataforma Brechó-VCM, analisada e marcada agora com elementos de gerenciamento de ECOSs. Elementos de negócio são incorporados por mecanismos que apoiam o canal de distribuição e o mercado de componentes (SANTOS, 2010). Fonte: (SANTOS & WERNER, 2011d).....	113

Figura 4.9 – Dimensão social do <i>ReuseECOS</i> . Fonte: (SANTOS & WERNER, 2012a)	114
Figura 4.10 – Modelagem de redes em ES. Os atores são representados por nós em formato de avatar e os artefatos, por nós em formato de folha de papel. Fonte: (VAHIA <i>et al.</i> , 2011)	116
Figura 4.11 – Rede social de atores e conhecimentos e seus relacionamentos em um ambiente de desenvolvimento de software com ECOSs. Fonte: (SEICHTER <i>et al.</i> , 2010)	117
Figura 4.12 – Interações entre atores e artefatos na plataforma Brechó, extraídos pela ferramenta StatSVN, em formato de lista. Fonte: (SANTOS & WERNER, 2012a)	118
Figura 4.13 – Interações entre atores e artefatos na plataforma Brechó (LoC por modificações e contribuições), extraídos pela ferramenta StatSVN. Fonte: (SANTOS & WERNER, 2012a)	119
Figura 4.14 – Arquitetura para apoiar o ciclo de vida social de ECOSs, baseado em (SANTOS & WERNER, 2012b). Fonte: (SANTOS <i>et al.</i> , 2013b)	119
Figura 4.15 – Exemplo dos conceitos de ECOSs	120
Figura 4.16 – Parte de um modelo conceitual de ECOSs preliminar	125
Figura 4.17 – Parte da RPS do ECOS <i>SWCompany</i>	125
Figura 4.18 – Variabilidade em ECOSs, onde um ator pode assumir diferentes papéis dependendo do ponto de vista do interessado. Optou-se pelo interesse sobre o ECOS <i>SWCompany</i> , cujo papel do ator SAP é de fornecedor, após uma adaptação da proposta de BOUCHARAS <i>et al.</i> (2009). O ajuste corresponde a: (1) nível de escopo interno do ECOS representado por linha tracejada pela visão externa do ECOS (ao invés de linha contínua); e (2) organização SAP representada por dois símbolos, indicando seu papel sujeito ao ponto de vista (ao invés de um símbolo)	126
Figura 4.19 – Definição de ECOSs para esta proposta de tese	128
Figura 4.20 – Infraestrutura de apoio à abordagem ReuseSEEM. Fonte: (SANTOS & WERNER, 2012b)	130

Índice de Tabelas

Tabela 1.1 – Fases da estratégia de pesquisa. Fonte: (SANTOS & WERNER, 2012a).	15
Tabela 2.1 – Classificação de características em ED e LPS	27
Tabela 2.2 – Tipos de características na notação Odyssey-FEX. Fonte: (FERNANDES, 2009).....	28
Tabela 2.3 – Relacionamentos na notação Odyssey-FEX. Fonte: (FERNANDES, 2009)	29
Tabela 2.4 – Propriedades de um componente. Fonte: (SANTOS, 2010)	33
Tabela 3.1 – Termos e acrônimos utilizados para ECOSs na literatura. Fonte: (BARBOSA <i>et al.</i> , 2013).....	61
Tabela 3.2 – Tipos de papéis chave nos ECOSs. Fonte: (HANSSEN, 2012)	64
Tabela 3.3 – Taxonomia dos ECOSs. Fonte: (BOSCH, 2009)	65
Tabela 3.4 – Identificando fontes de teoria em ECOSs. Fonte: (HANSSEN & DYBA, 2012).....	78
Tabela 3.5 – Desafios para ECOSs. Fontes: (BOSCH, 2009) e (JANSEN <i>et al.</i> , 2009b)	81
Tabela 3.6 – <i>Roadmap</i> de pesquisa para ECOSs em ES no Brasil. Fonte: (SANTOS <i>et al.</i> , 2012c)	85
Tabela 4.1 – ECOSs <i>free and open source software</i> (FOSS) do LENS/Reuse da COPPE/UFRJ.....	94
Tabela 4.2 – Comparação entre estratégias de abertura das plataformas dos ecossistemas do ECOS LENS/Reuse, com base em (ANVAARI & JANSEN, 2010), onde: <i>P</i> = possibilidade, <i>L</i> = <i>status</i> da licença, <i>Po</i> = possível, <i>Pc</i> = possível para alguns componentes, <i>Np</i> = não permitido, <i>Pn</i> = permissão não é necessária, <i>Pa</i> = permissão é necessária em alguns casos, e <i>Ps</i> = permissão é sempre necessária. Fonte: (SANTOS & WERNER, 2011b).....	99
Tabela 4.3 – Tipos de interação entre atores e conhecimentos nos ECOSs. Fontes: (SEICHTER <i>et al.</i> , 2010) e (SANTOS <i>et al.</i> , 2013b)	116

Tabela 4.4 – Resumo dos trabalhos relacionados em relação aos requisitos da abordagem proposta, onde  = atende;  = atende parcialmente;  = não atende	142
Tabela 5.1 – Cronograma de atividades para esta proposta de tese, onde T = trimestre	150

Capítulo 1 – Introdução

1.1 Preâmbulo

Software é um elemento ou entidade que transitou da “obscuridade” para “indispensabilidade” nos últimos 50, ou melhor, 60 anos, considerando os dizeres de MESSERSCHIMITT & SZYPERSKI (2003). A indústria de software é distinta, tecnológica, organizacional e socialmente falando, logo, a sua crescente importância demanda profissionais de diversas áreas, a fim de lidar com questões técnicas e sociais (CUKIERMAN *et al.*, 2007). Usuários e desenvolvedores de software precisam de um vocabulário comum para discutir as questões inerentes ao software e seus impactos no mundo dos negócios. Isso corrobora com o fato de que a Tecnologia da Informação (TI) não é somente um recurso útil como a eletricidade ou telefone, mas representa uma fonte fundamental de transformação das organizações e da sociedade, sobretudo no contexto globalizado do século XXI (VENKATRAMAN, 1994).

A indústria de software se tornou crucial para a economia global e gera benefícios diretos e indiretos para profissionais de várias áreas do conhecimento. Contudo, do ponto de vista da sociedade, existem alguns obstáculos inerentes à compreensão do software como produto (MESSERSCHIMITT & SZYPERSKI, 2003). Entre os desafios, pode-se citar: natureza “misteriosa” e “intangível” da tecnologia de software; falta de percepção da estrutura interna de funcionamento da sua indústria no contexto do uso; e oportunidades e desafios proporcionados pela natureza ubíqua/embarcada dessas tecnologias. Com o crescimento dessa indústria, avanços na variedade e quantidade de dispositivos tecnológicos e de sistemas produzidos na década de 2000 viabilizaram a chamada “Era da Web” (SANTOS *et al.*, 2012d). Essa “*Weblização*” gerou potencial para guiar a “informatização da informação” nos mais diferentes sistemas sociais, de negócios e militares (BOEHM & SULLIVAN, 2000).

Para apoiar essa indústria, a Engenharia de Software (ES) tem avançado em pesquisa teórica e aplicada, e.g., criando métodos para desenvolver sistemas de software de grande porte, interconectados e sujeitos à rápida velocidade de implantação e de evolução (SANTOS *et al.*, 2013a). Segundo BOEHM (2006), a meta da ES está em construir produtos, serviços e processos que agregam valor à sociedade. Ou seja, a indústria de software existe porque produz valor, de maneira que a sua percepção deve

ser considerada em todos os aspectos (BOEHM & SULLIVAN, 2000). Todos os envolvidos contribuem para (e se utilizam de) tais tecnologias como elementos de apoio às suas atividades. Eles se esforçam nas decisões e ações para maximizar uma noção simples ou complexa de valor sobre os artefatos resultantes de suas atividades, consciente ou inconscientemente, com relação a metas compartilhadas ou para satisfazer objetivos individuais e/ou coletivos (SANTOS *et al.*, 2012d).

Paralelamente à questão do valor na ES, do ponto de vista das organizações fornecedoras de software, permanece uma questão: o desenvolvimento de software de larga escala é complicado, custoso, lento e imprevisível (BOSCH & BOSCH-SIJTSEMA, 2010). Motivada pelos aspectos que permeiam a indústria de software, a área de ES vem se preocupando com as questões sociais e econômicas entrelaçadas ao processo de desenvolvimento, o que foi apontado como desafio para os próximos anos (BOEHM, 2006). Entre as explicações para este fato, pode-se destacar:

O desenvolvimento de software requer pensar mais cuidadosamente as plataformas que vão apoiá-lo e as suas redes de artefatos e de envolvidos, i.e., relacionamentos de conectividade e de dependência entre os produtos e as organizações, envolvendo fornecedores, distribuidores, organizações terceirizadas, desenvolvedores de produtos e de serviços relacionados e uma gama de outras organizações e clientes que afetam (e são afetados por) esse cenário.

(JANSEN *et al.*, 2009a)

Inovações não mais se originam de uma única organização, mas da chamada “co-inovação” a partir de diferentes agentes da indústria de software, além dos processos denominados de “co-evolução”, possíveis somente a partir da reunião de organizações colaborativamente focadas em dar suporte a novos produtos, a satisfazer as necessidades dos clientes e a incorporar novas rodadas de inovação.

(ARNDT & DIBBERN, 2006)

Para BOSCH (2009), esse cenário demanda engenheiros de software com a habilidade de abstrair a complexidade do sistema como um todo. Neste caso, o sistema de software representa uma combinação de software, hardware e “peopleware”, constituída sobre um ambiente comum, i.e., uma plataforma. Pode-se identificar, então, uma tendência na ES: o nascimento, desenvolvimento, amadurecimento e eventual “morte” ou transformação de plataformas levam à “desconstrução” da visão tradicional da ES. Privilegia-se assim uma nova visão, que contempla a colaboração e a interoperabilidade entre os atores (CAMPBELL & AHMED, 2010). Na década de 2000,

essas estruturas foram alvos de pesquisa na ES, chamadas de Ecossistemas de Software ou ECOSs (MESSERSCHMITT & SZYPERSKI, 2003). Algumas das maiores organizações de desenvolvimento de software como Amazon, Microsoft, Nokia, SAP, Google e Apple estão liderando o desenvolvimento de ECOSs, o que vem contribuindo para que o tema alcance um *status* de tópico de pesquisa concebido pela movimentação real da indústria de software e de serviços relacionados (WERNER *et al.*, 2012).

Essas e outras transformações na perspectiva de pesquisa e prática da ES fez com que o advento dos ECOSs levasse os principais agentes da indústria a reconsiderarem as suas práticas operacionais (WERNER & SANTOS, 2012). Foi criado um fórum específico para o tema, *International Workshop on Software Ecosystems* (IWSECO). O IWSECO tem sido realizado desde 2009 em diferentes conferências¹, cujas publicações se concentram sobre os três pilares dessa nova “metáfora” para lidar com o desenvolvimento de software. Entre eles, estão questões relacionadas a negócios, envolvimento de terceiros com a formação de uma rede de produção e concepção de plataformas abertas (CAMPBELL & AHMED, 2010). No entanto, a missão da pesquisa em ECOS é mais ambiciosa, por lidar com várias facetas da ES, como reutilização, arquiteturas de sistemas, trabalho cooperativo apoiado por computador, redes sociais, qualidade e economia aplicada. Ou então, como preferem alguns, uma engenharia melhor percebida por um olhar sociotécnico² (CUKIERMAN *et al.*, 2007).

1.2 Motivação

Inspirado nos ecossistemas biológicos, sociais e de negócios, o tema ECOSs favorece o uso de analogias para aplicar novas visões e abstrações sobre os desafios da indústria de software, discutidos na Seção 1.1 (MOORE, 1993; IANSITI & LEVIEN, 2004a; JANSEN & CUSUMANO, 2012). Por exemplo, a influência da visão de negócio pode ser notada no fato de que as equipes de desenvolvimento têm trabalhado geograficamente dispersas, pois a indústria decidiu explorar processos e recursos

¹O IWSECO foi realizado em conjunto com a *International Conference on Software Reuse*, em 2009, *European Conference on Software Architecture*, em 2010 e *International Conference on Software Business*, em 2011 e 2012. Em 2013, será realizado novamente junto a *International Conference on Software Business*. O site para acesso da última edição é: <<http://iwseco12.wordpress.com/>>.

²Para CUKIERMAN *et al.* (2007), o olhar sociotécnico almeja apreender a ES sem fragmentá-la em fatores (ou aspectos) técnicos de um lado e fatores (ou aspectos) não técnicos de outro; sem fatorá-la em quaisquer outras dualidades (“fatores técnicos” versus “fatores humanos, organizacionais, éticos, políticos, sociais etc.”) que terminem por desfigurar o “pano sem costura” que imbrica, na ES, o técnico e o social em um mesmo e indivisível “tecido”.

distribuídos para reduzir custo e tempo. Isso permite a expansão da indústria para novos mercados, embora necessite de um olhar para questões sociais e culturais (PINTO *et al.*, 2009). O entrelaçamento resultante mostra três tendências que aceleram a complexidade no desenvolvimento no novo contexto da ES (BOSCH & BOSCH-SIJTSEMA, 2010):

- Ampla adoção de **Linhas de Produtos de Software** (LPSs) e os desafios de expor os ativos reutilizáveis³ e de abrir a arquitetura da plataforma;
- **Globalização do Desenvolvimento de Software** (GDSs) e o desafio de gerenciar a dependência socio-técnica⁴ em suas redes de produção;
- **Organização de ECOSs** em comunidades de desenvolvedores externos ao redor de um produto de sucesso⁵ e o desafio de entender as relações entre as partes sob a perspectiva da reutilização mais ampla.

Nessa direção, o desenvolvimento de software se voltou mais para a composição de partes existentes, construídas internamente pela organização, do tipo *open source*, ou ainda adquiridas no mercado, denominadas componentes (OVERHAGE & THOMAS, 2004). Seguindo a corrente de pensamento da Engenharia, o desenvolvimento industrial maduro é orientado à reutilização e usa artefatos existentes sempre que possível, a fim de manter repositórios de qualidade (ENSMENGER & ASPRAY, 2002). Chama-se de Desenvolvimento Baseado em Componentes (DBC) o conjunto de atividades para projetar e codificar componentes de software e apoiar a construção de sistemas de qualidade e com propriedades previsíveis (BASS *et al.*, 2000). No contexto dos ECOSs, o desenvolvimento de um novo software pode se ater às funcionalidades específicas, procurando atender as particularidades, necessidades do cliente, tecnologias existentes etc., que conferem vantagem competitiva ao sistema como um todo e que estão apoiadas sobre diferentes plataformas.

Além do desenvolvimento *com* reutilização discutido, deve-se considerar a construção propriamente dita de componentes: o desenvolvimento *para* reutilização (MOORE & BAILIN, 1991). Isso compreende a Engenharia de Domínio (ED), i.e., um processo de identificação e organização do conhecimento sobre uma classe de

³Ativo reutilizável consiste de um item, e.g., modelo, especificação, código fonte, documentação, casos de teste, manuais, procedimentos etc., que foi projetado para uso em múltiplos contextos e que agrega valor para a organização (SOFTEX, 2012).

⁴Uma rede socio-técnica compreende um grafo construído a partir da rede técnica, i.e., artefatos como nós e arestas como dependências entre eles, mas inclui também a identificação dos autores responsáveis por cada artefato (SANTOS *et al.*, 2012d).

⁵Produto de sucesso corresponde ao produto principal, ou estratégico, para algumas empresas, e.g., um software central e popular em um nicho de mercado (BOUCHARAS *et al.*, 2009).

problemas, o domínio do problema, para apoiar a sua descrição e solução (PRIETO-DÍAZ & ARANGO, 1991). O conceito de domínio pode ser entendido como uma classe de sistemas que apresentam funcionalidades similares. No contexto dos ECOSs, a estratégia tradicional de desenvolver um produto de software único tem sido substituída pelo desenvolvimento de múltiplos produtos, baseados em uma arquitetura e plataforma comuns (BOSCH, 2010). Além das questões técnicas, seja no desenvolvimento de software em ECOSs, ou na ES baseada em ED, alguns problemas podem comprometer o processo, e.g., desconhecimento (ou conhecimento errôneo) do domínio da aplicação, vocabulário conflitante e incerteza nas decisões (WERNER & BRAGA, 2005).

Similar ao que aconteceu nas décadas de 1990 e 2000 com a previsão de um mercado de componentes global e/ou corporativo (SANTOS, 2010), as organizações têm sofrido pressão do mercado para a abertura das plataformas e envolvimento de desenvolvedores externos. O software passa a não existir mais de maneira “fechada”, ficando distribuído e constituindo uma comunidade ao seu redor. Esta abertura gera algumas indagações (ANVAARI & JANSEN, 2010): “plataforma *open source* ou proprietária: qual cenário gera maior sucesso?”, “qual estratégia/plataforma tem recebido maior atenção da comunidade de desenvolvedores?” e “qual delas produz aplicações mais inovadoras?”. Questões críticas ao mercado de componentes continuam atuais, e.g., padronização de componentes, visualização de informação, diretrizes para customização e propriedade intelectual (WERNER *et al.*, 2009). Isso expõe, de forma mais ampla, os impactos das questões sociais e econômicas da ES.

Na tentativa de repensar o processo de desenvolvimento, a visão dos ECOSs reforça a perspectiva multidisciplinar, que envolve Comunicação, Sociologia, Economia, Administração e Direito. Além disso, as organizações fornecedoras de software não trabalham mais como unidades independentes, que entregam produtos distintos, mas dependem de outros fornecedores de componentes e de infraestruturas vitais, e.g., sistemas operacionais, bibliotecas, lojas de componentes e plataformas (JANSEN *et al.*, 2009b). Por sua natureza, a pesquisa em ECOSs impossibilita um olhar puramente de Engenharia pois, segundo HANSSEN & DYBA (2012), suas características e desafios fundamentais remontam antigas teorias, como a socio-técnica (TRIST, 1981) e a da ecologia organizacional (TRIST, 1977).

Diante disso, a motivação deste trabalho consiste em evoluir uma pesquisa realizada no escopo de uma dissertação de mestrado, relativa ao desenvolvimento de um ambiente para apoiar um mercado de componentes baseado em valor e com foco no

processo de Gerência de Reutilização (SANTOS, 2010). Como uma das conclusões reforça o fato de que não existe um mercado de componentes global e corporativo amadurecido (SANTOS *et al.*, 2010a), as dificuldades na proposição e percepção de valor entre os envolvidos em uma indústria baseada em reutilização permanecem atuais, sobretudo com os novos canais de distribuição com as *application stores*. Assim, esta proposta de tese busca explorar no tema ECOSs uma maneira de entender: (i) como os artefatos e os atores na indústria de software globalizada se relacionam com base em conceitos de reutilização, e (ii) como os elementos de negócio e sociais podem apoiar na compreensão dessas redes.

1.3 Problema

De acordo com recentes estudos secundários baseados em ES Experimental (i.e., mapeamentos sistemáticos da literatura), o tema ECOSs tem se tornado um tópico de pesquisa ativo e definido com os novos modelos abertos de negócio, que geram novos papéis e padrões para colaboração, inovação e proposição de valor (BARBOSA & ALVES, 2011; HANSSEN & DYBA, 2012; MANIKAS & HANSEN, 2013). Pelo interesse da indústria de software, uma comunidade de pesquisa se formou para investigar como a disciplina de ES é afetada e seus resultados vêm aparecendo no IWSECO, além de periódicos como a edição especial do *Journal of Systems and Software* (BOSCH, 2012). Em decorrência de ser um novo olhar sobre a ES, a terminologia e as definições dentro da comunidade variam amplamente, e mesmo o conceito de ECOSs ainda é vago e diverso. Um estudo sobre as bases teóricas dos ECOSs mostra que este cenário interfere diretamente nas contribuições de pesquisa orientadas pela experimentação (HANSSEN & DYBA, 2012). Apesar de natural em novos temas, isso instiga o desafio da convergência para um corpo de conhecimento, o que impacta na compreensão dos ECOSs (SANTOS *et al.*, 2012c).

MANIKAS & HANSEN (2013) conduziram uma revisão sistemática sobre ECOSs e mostraram que existe um alto número de publicações do tipo relatos e/ou casos de ECOSs, sobretudo *free and open source software*⁶ (FOSS), carecendo de

⁶Um software livre (*free software*) é um programa que pode ser usado, compartilhado, modificado e distribuído, inclusive comercialmente. O código fonte do programa é aberto e geralmente pode ser obtido gratuitamente. O aspecto mais importante do software livre é a liberdade. O código aberto (*open source*) usa o modelo de desenvolvimento do software livre e o torna mais atrativo para as organizações e para os negócios. O aspecto mais importante é que o código fonte é aberto e permite a cooperação, assim como no software livre. A principal diferença é que o código aberto não foca na liberdade e sim nos negócios.

modelos analíticos e de estudos de caso na indústria de software. BARBOSA & ALVES (2011) realizaram um mapeamento sistemático e resumiram os principais assuntos tratados no tema ECOSs: FOSS, modelagem, arquitetura de software, sistemas operacionais, co-inovação em software, evolução de software, negócios e LPSs. Alguns desafios listados em uma atualização do mapeamento sistemático podem ser destacados (BARBOSA *et al.*, 2013): ecossistemas *open source*, governança de ECOS, análise de ECOS, abertura de ECOS, qualidade e arquitetura de software. Ressalta-se que ambos os trabalhos confirmam que as pesquisas em ECOSs estão voltadas para o tratamento das questões sociais e econômicas no processo de ES, o que confirma os desafios apontados por BOEHM (2006).

Por outro lado, conhecimentos inspirados em ecossistemas biológicos e de negócios têm valor, desde que respaldados pelo paradigma experimental e pelas particularidades do “negócio do software”, e.g., redução da distância entre seu projeto e uso, intangibilidade e rapidez na inovação (HANSSEN & DYBA, 2012). Os negócios estão se transformando a partir de um novo pensamento do final da década de 1990: não faz mais sentido para uma organização construir ou se apropriar de todos os aspectos de seu modelo de negócio (GOLDMAN & GABRIEL, 2005). As organizações que querem gerar valor de diferentes formas estão interessadas em produtividade, o que inclui a capacidade de inovar, sobretudo na indústria de software global e dinâmica. Independente de quão espertos, criativos e inovadores sejam seus desenvolvedores e demais envolvidos internos, sempre existirão outros profissionais de nível semelhante ou superior fora de suas fronteiras, o que define a inovação aberta (CHESBROUGH, 2003; BURMANN & MAURO, 2011).

Toda organização deve ser orientada para levar mais produtos ao mercado, com ciclos mais reduzidos, o que pode apontar estratégias de terceirização ou de linhas de produtos mais flexíveis. Os canais de distribuição disponíveis podem ser evoluídos para criar novos modelos de negociação e de abertura das plataformas a cada interação, viabilizando análises de mercado. Portanto, a compreensão dos ECOSs é fundamental para o sucesso de uma organização e isso tem refletido nas recentes iniciativas de pesquisa. Por exemplo, a estratégia de visualização proposta por IYER (2012) discute um procedimento para determinar a estrutura da indústria, ao identificar as organizações e suas dependências em um diagrama de rede a fim de visualizar e interpretar um

Por serem parecidos, são chamados “FOSS”. Disponível em: <<http://idilix.net/pt-br/software-livre-open-source-freeware>>.

ECOS. Por sua vez, JANSEN & CUSUMANO (2012) discutem a governança de ECOSs, visando implantar ferramentas que apoiem indicações e estratégias para manter e melhorar a saúde do ecossistema em termos de produtividade, robustez e criação de nicho dentro da organização.

A contribuição desses esforços está centrada na análise de apenas um ECOS, por desconsiderar uma base de conhecimento comum para a sua definição e modelagem, o que impede a navegação entre outros ECOSs envolvidos a partir de seus elementos. Dessa forma, o tratamento de um problema anterior permanece: *a compreensão dos conceitos e de suas relações entre diferentes ECOSs diante da abertura de suas plataformas na indústria de software*. Ou seja, soluções para este problema são fundamentais para que se possa sedimentar as atividades de análise de ECOSs e gerar ações estratégicas e de inovação em um processo de desenvolvimento mais aberto.

1.4 Contexto da Proposta de Tese

Esta pesquisa estabelece uma estratégia para posicionar os ECOSs na indústria, estimulando a percepção das interferências que a criação de valor vem gerando no processo de desenvolvimento na ES. Estas interferências podem ser definidas como *senso de negócio* e *senso de comunidade* e são críticas para a implantação da reutilização sistemática na indústria, conforme vários estudos prévios (BOEHM & SULLIVAN, 2000; SZYPERSKI *et al.*, 2002; MESSERSCHIMITT & SZYPERSKI, 2003; OVERHAGE & THOMAS, 2004; FRAKES & KANG, 2005; BOEHM, 2006; SANTOS & WERNER, 2010a). Dispondo os sentidos de negócio e de comunidade em um quadrante (Figura 1.1), pode-se produzir uma visão da reutilização em direção aos ECOSs, cuja linha do tempo é discutida na Seção 3.2. Observa-se que a técnica de LPS representa uma tentativa de explorar o senso de negócio ao orientar a reutilização sob o ponto de vista do mercado. Por sua vez, a reutilização em ambientes FOSS busca explorar o senso de comunidade ao orientar o processo de ES para a inovação trazida pela participação da comunidade em torno de um empreendimento de software⁷.

Dessa forma, a reutilização pode se beneficiar da metáfora dos ECOSs como um nível de abstração mais alto para as suas técnicas, visando explorar um novo caminho

⁷Conforme SANTOS (2010), um empreendimento de software se refere a um artefato relacionado a software, considerando a sua concepção, desenvolvimento e manutenção/evolução, seja este artefato um componente, um projeto, um produto, um processo, a organização como um todo ou até mesmo um mercado estabelecido.

para a reutilização interorganizacional. Ou seja, é necessário definir, modelar e analisar as plataformas de ECOSs para entender as pressões do mercado sobre o seu processo de abertura. A ideia está na transição da ES baseada na reutilização tradicional para aquela baseada na reutilização com ECOSs, ao perceber os novos aspectos constituintes:

- no **senso de negócio**, a variável *estratégia* pode ser apoiada por mecanismos que estendem repositórios de reutilização para compor o repositório de ativos de ECOSs, que manipula atributos de negócio sobre produtos e serviços, além dos tradicionais componentes reutilizáveis (SANTOS & WERNER, 2010b; ALBERT *et al.*, 2012);
- no **senso de comunidade**, a variável *inovação* pode ser apoiada por mecanismos que estendem *sites* de redes sociais para compor o repositório de ativos de ECOSs, que manipula atributos de comunidade sobre atores externos à organização, além de seus relacionamentos (SCHNEIDER *et al.*, 2011; SANTOS *et al.*, 2012a).

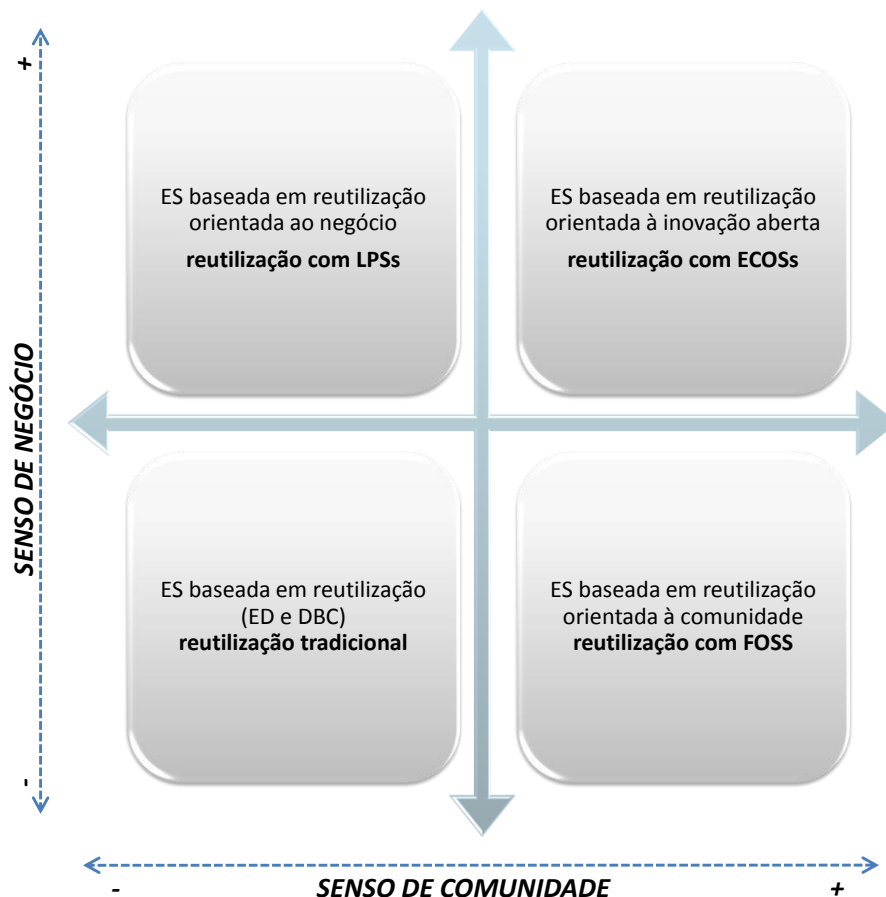


Figura 1.1 – Cenários da Reutilização de Software em direção aos ECOSs

Por exemplo, em um cenário de mercado, os impactos desses mecanismos podem ser sentidos quando uma organização deseja avaliar estratégias na indústria a partir da definição e modelagem do seu ECOS, consultando o canal de distribuição⁸ de seus produtos e serviços. Por sua vez, a comunidade interage com a plataforma utilizando repositórios locais e contribui para a base de dados históricos do ECOS, que pode ser cruzada com dados do mercado de outros ECOSs os quais o ECOS analisado se relaciona. Assim, o canal de distribuição funciona como um elemento mediador e catalisador do processo de abertura das plataformas, ao viabilizar a participação da comunidade no desenvolvimento de software com ECOSs, alvo da reutilização interorganizacional. No contexto da inovação aberta, esse canal representa uma fonte de informações para os intermediários, que aproximam os envolvidos na indústria de software, buscando conectar ofertas a demandas (SCHILLING, 2008).

A combinação entre a estratégia discutida e a ES tradicional permite propor o *ambiente de ECOSs baseado em reutilização*, cujo contexto geral é ilustrado pela Figura 1.2 e que adapta a ED para sua elaboração, conforme detalhado na Seção 4.3. O ambiente pode ser dividido em duas partes, conforme as atividades a serem realizadas (eixo das abscissas): (i) *ambiente de desenvolvimento para ECOSs* (DPE), que visa concretizar uma proposta de ECOSs baseada em reutilização ao definir o domínio dos ECOSs com uso de modelos conceituais e representar as redes dos seus componentes⁹; e (ii) *ambiente de desenvolvimento com ECOSs* (DCE), que visa implementar o domínio de ECOSs, estendendo o desenvolvimento de LPSs ao adicionar os ECOSs como um novo nível de abstração. No eixo das ordenadas, distingue-se entre o nível conceitual, voltado para a construção de uma base de conhecimento que permite a definição de ECOSs, e o nível concreto, voltado para a representação das redes formadas por eles.

Por fim, o eixo das cotas utiliza linhas para representar os processos de realimentação no ambiente. Na Figura 1.2, as linhas contínuas mostram o ciclo de engenharia. O ambiente de DPE possui um gerador de instâncias de ECOSs, i.e., um

⁸Canal de distribuição corresponde a uma entidade integradora on-line que possui um repositório de referência com mecanismos de documentação, armazenamento, publicação, busca e recuperação de ativos, focados em um mercado e na exploração e manutenção de dados históricos relacionados. Apresenta um conjunto de envolvidos que atuam como intermediários nas transações comerciais, podendo ser profissionais da área e/ou máquinas virtuais (automatizados) (SANTOS *et al.*, 2010a).

⁹Conforme apresentado na Seção 4.3.2, componentes de ECOSs consistem no conjunto de base de conhecimento, modelos conceituais instanciados, dados históricos, relatos de experiência, arquiteturas de ECOSs, atores, artefatos, relacionamentos, papéis, responsabilidades, plataformas de ECOSs, e produtos e serviços dos ECOSs.

módulo que consulta o modelo de domínio para selecionar conceitos e suas relações em um ECOS. Para isso, a ontologia consulta um glossário de termos e significados¹⁰ para apoiar o gerador neste processo. Uma base de dados históricos também é verificada para evitar que haja duplicação de instâncias de conceitos e de relações de outros ECOSs previamente registrados, e.g., uma organização cadastrada em outro ECOS, mas que tem papel de fornecedor no ECOS a ser definido. Isso corresponde à *análise de domínio de ECOSs*. Por sua vez, existe um extrator de elementos e conexões que analisa o modelo de domínio do ECOS instanciado para modelar as suas redes ao recuperar atores e artefatos (elementos), bem como papéis, responsabilidades e relacionamentos (conexões). Isso é feito através de consultas ao repositório de ativos de ECOSs, i.e., uma extensão de repositórios de reutilização relacionados a *sites* de redes sociais para explorar os sentidos de negócio e de comunidade e armazenar os componentes de ECOSs.

Com os elementos e conexões representados em redes, um extrator de subdomínios de ECOSs pode ser acionado para realizar configurações ou recortes sob o ponto de vista de um interessado, criando casos de ECOSs. Isso ocorre pela seleção de atores e artefatos do ECOS, bem como papéis, responsabilidades e relacionamentos de interesse, a nível organizacional, da visão interna ou da visão externa do ECOSs. O principal objetivo dessa extração está em explorar a variabilidade nos ECOSs, i.e. lidar com as diferenças entre os ECOSs e explorar as características e comportamentos comuns entre eles (e.g., atores e artefatos). Nesse contexto, para a representação das redes, existem algumas opções: (i) a rede de produção de software (RPS), cuja notação é proposta por BOUCHARAS *et al.* (2009) e originalmente apresentada por JANSEN (2007), com viés de negócio; e (ii) a rede sociotécnica, baseada na notação da rede socio-técnica discutida por MAGDALENO (2010) e VAHIA *et al.* (2011), com viés de comunidade. Isso corresponde ao *projeto de domínio de ECOSs*.

¹⁰Segundo BREITMAN (2005), embora importantes para difundir o conhecimento na comunidade de pesquisa e prática, glossários são mecanismos pouco formais e de baixa expressividade. Dessa forma, adota-se a estratégia de LOPES (2010) de separar o glossário da ontologia, de modo que conceitos e significados sejam organizados por um mecanismo (e.g., Wiki) e modelos conceituais expressem os seus relacionamentos, assim como suas propriedades, funções, atributos e regras de restrição. Ambos são dependentes entre si: o glossário não define as relações entre os conceitos da lista e o significado dos conceitos nos modelos ontológicos deve estar em sintonia com o glossário (MONTONI, 2003).

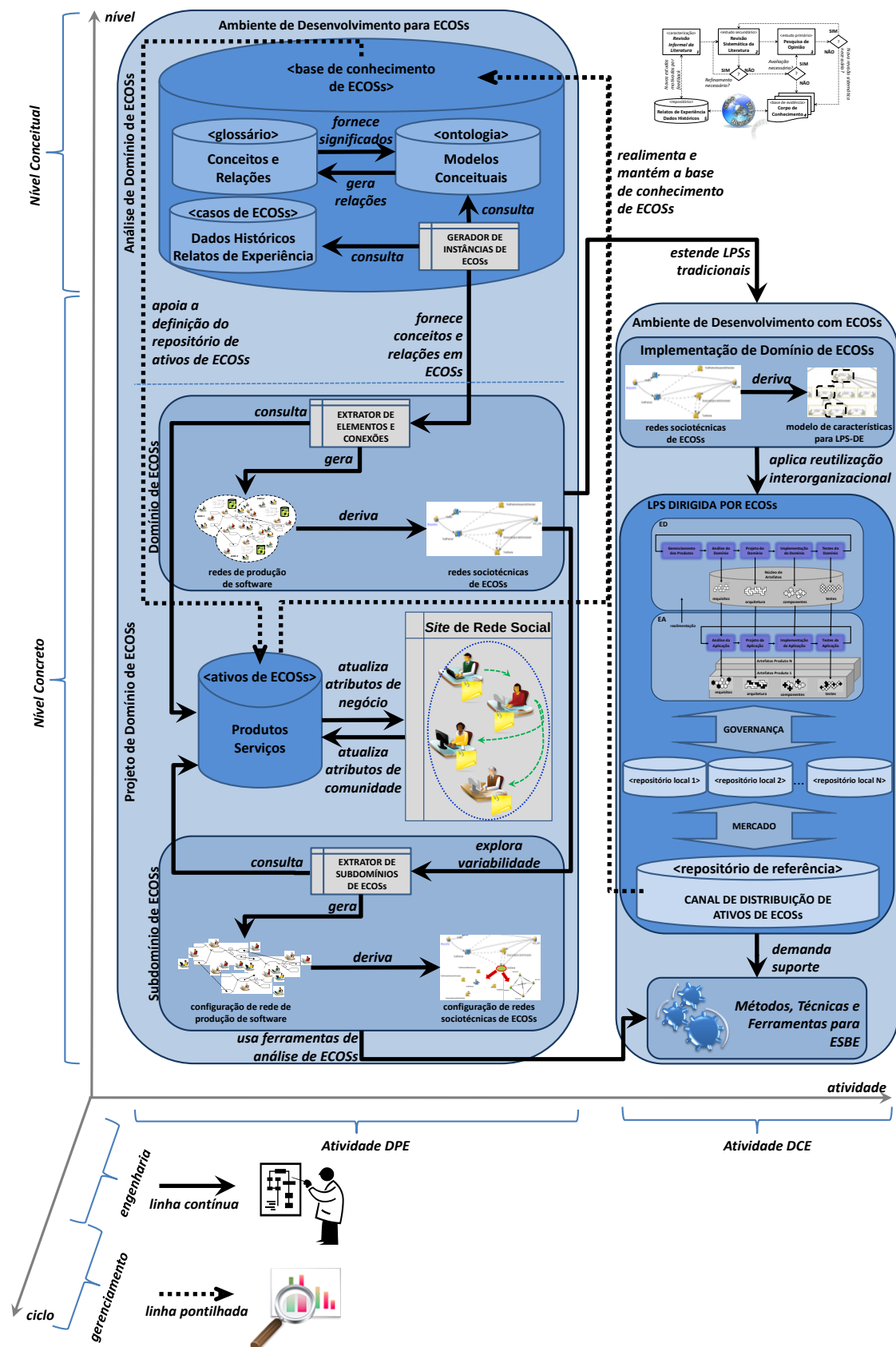


Figura 1.2 – Ambiente de ECOSs baseado em reutilização

Depois de selecionado um subdomínio, o interessado pode utilizar um método, técnica ou ferramenta de análise para explorar o caso do ECOS, a fim de calcular métricas ou gerar indicadores sobre suas três dimensões, i.e., *técnica*, *transacional* e *social* (CAMPBELL & AHMED, 2010; SANTOS & WERNER, 2011b; BARBOSA *et al.*, 2013). Alguns trabalhos desenvolvidos no contexto desta proposta de tese oferecerão este suporte: (i) um recomendador de tecnologias e de parceiros para uma organização que deseja compreender o seu ECOS e os ECOSs subjacentes (ALBERT, 2012), e (ii) um visualizador do ciclo social do ECOS, que explora produtividade, robustez e criação de nicho nas redes sociotécnicas de ECOSs ao calcular elementos de comunidade, e.g., interação, utilidade, reputação, promoção, contribuição e recomendação (SANTOS *et al.*, 2013b).

O ambiente de DCE utiliza os produtos da análise e do projeto de domínio de ECOSs com o intuito de apoiar a extensão de LPSs para gerir a abertura de suas arquiteturas. Por exemplo, implantar a governança de repositórios locais que contém artefatos dos núcleos de LPSs para incorporar os componentes de ECOSs e comunicar com o mercado visando a troca de informações, recursos e artefatos com o canal de distribuição. Isso corresponde a *implementação de domínio de ECOSs*. Por sua vez, as linhas pontilhadas na Figura 1.2 destacam o ciclo de gerenciamento. Do lado esquerdo, o repositório de ativos de ECOSs depende da base de conhecimento para manter coerência com o domínio de ECOSs. Do lado direito, a base de conhecimento deve ser realimentada a partir dos ambientes de DPE e de DCE, a fim de manter a validade dos modelos conceituais. O processo de realimentação propõe usar uma metodologia que orienta pesquisas mais aprofundadas sobre métodos de análise de ECOSs, considerando os problemas discutidos na Seção 1.3. O objetivo é organizar e manter um *framework* para engenharia e gerenciamentos (E&G) de ECOSs com base em suas três dimensões. A metodologia consiste em uma estratégia de pesquisa baseada em experimentação, estendida de (SPÍNOLA *et al.*, 2010) e adaptada de (SANTOS *et al.*, 2011; 2012b).

A Figura 1.3 esquematiza essa estratégia, cujas fases são descritas na

Tabela 1.1. A partir de uma revisão informal da literatura para compreender os conceitos e terminologias usados pela comunidade de pesquisa em ECOSs, as próximas fases são: (i) planejar e executar um conjunto de *revisões sistemáticas* específicas para verificar passos e atividades que devem compor o *framework* para E&G de ECOSs com base nas três dimensões, cujo enfoque está em analisar a literatura de ES para explorar

as contribuições de/para ECOSs; (ii) caso necessário, executar novamente estudos deste tipo, ou planejar e executar *pesquisas de opinião* (surveys) para validar os resultados com especialistas, a fim de sedimentá-los pela comunidade; (iii) atualizar o *framework* visando manter a *base de conhecimento* de ECOSs e analisar casos de interesse da comunidade de pesquisa no tema, e.g., Android, Blackberry, Force.com. Eclipse, Microsoft, Linux etc.; e (iv) essas análises devem nortear a organização de um *repositório de dados históricos e de relatos de experiências* para a gerência de casos de ECOSs, através do registro e monitoramento de seus elementos para extrair diagnósticos e realizar interferências.

Esse repositório se inspira no fato de que os ecossistemas aparecem, se desenvolvem, amadurecem e/ou desaparecem, como acontece com mercados, tecnologias, infraestruturas, organizações, processos, modelos, técnicas etc. Um indício disso está na demanda por relatórios de institutos de pesquisa e recomendação de Tecnologia de Informação (TI), como *Gartner* (GARTNER, 2013), *Forrester* (FORRESTER, 2013) e *Thoughtworks* (THOUGHWORKS, 2013), que dedicam seus esforços a acompanhar o mercado de software e serviços e agem como “conselheiros de TI” para as organizações (ALBERT, 2012). Estes conselhos apresentam gráficos gerados por processos de pesquisa, seja por refinamento de cenários de mercado ou pela condução de pesquisas de opinião com especialistas, com foco em objetivos estratégicos. Assim, a estratégia de pesquisa discutida sugere um caminho para explorar esforços colaborativos, distribuídos e especializados no tema ECOSs.

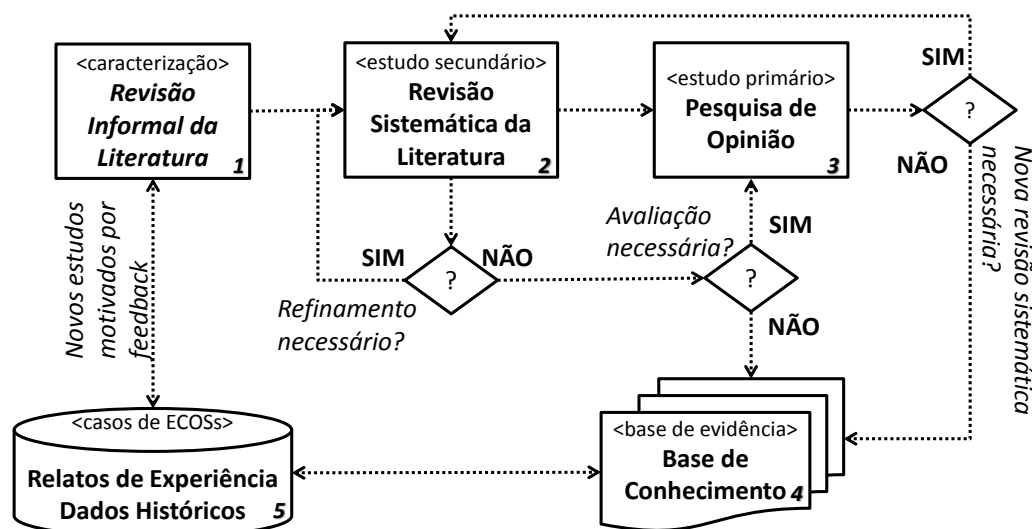


Figura 1.3 – Estratégia de pesquisa, estendida originalmente de (SPÍNOLA *et al.*, 2010) e adaptada de (SANTOS *et al.*, 2011; 2011d).

Tabela 1.1 – Fases da estratégia de pesquisa. Fonte: (SANTOS & WERNER, 2012a)

FASE	OBJETIVOS
Revisão Informal da Literatura	Visa identificar conceitos sobre ECOSs (i.e., termos, sinônimos e expressões) e permitir a definição de um protocolo de revisão sistemática, amplo e ao mesmo tempo preciso, para apoiar as suas dimensões e gerar um <i>framework</i> inicial.
Revisão Sistemática da Literatura	Visa elaborar e executar o protocolo de revisão sistemática para gerar estudos de verificação dos passos e atividades que devem compor um <i>framework</i> para E&G de ECOSs com base nas três dimensões. Baseado nos resultados extraídos da análise das fontes, os pesquisadores de cada estudo decidem se é necessário refiná-lo ou se o conjunto de conhecimento mapeado deve ser validado.
Pesquisa de Opinião	Visa planejar e executar estudos de avaliação do conhecimento adquirido na fase anterior, considerando especialistas em ES e em ECOSs específicos para o tópico de cada objeto de estudo analisado nas revisões sistemáticas.
Base de Conhecimento	Visa reunir todo o conhecimento obtido pelas três fases anteriores a fim de manter a base de conhecimento de ECOSs na forma do <i>framework</i> atualizado e calibrado por casos de interesse da comunidade de ECOSs. Entre seus componentes, estão: documentos (e.g., relatórios e artigos); dados abertos de repositórios de projetos FOSS; dados históricos de negócio de ECOSs coletados da Internet; redes sociotécnicas inferidas a partir de redes sociais e comerciais; dados arquiteturais coletados a partir de IDEs; e repositórios de reutilização. Esta fase requer um repositório para compartilhamento de informação relativa a E&G de ECOSs.
Repositório de Dados Históricos e de Relatos de Experiência	Visa apoiar o ciclo de vida de um ECOS e o desenvolvimento de sua plataforma por meio de um mecanismo integrado a um repositório de ECOSs para permitir busca, recuperação, manipulação, armazenamento, documentação e publicação de elementos, dados e relatórios de ECOSs.

A pesquisa relativa ao ambiente de ECOSs baseado em reutilização está inserida no contexto dos projetos da Equipe de Reutilização de Software da COPPE/UFRJ, mais especificamente o Projeto *Brechó* (BRECHÓ, 2013) e o Projeto *Odyssey* (ODYSSEY, 2013), por adaptar e aplicar conceitos de DBC e de ED/LPS. Devido à dimensão da pesquisa e ao entendimento da impossibilidade do trabalho em sua totalidade ser conduzido por um único indivíduo, os temas envolvidos estão sendo tratados por um grupo composto por um aluno de doutorado (autor desta proposta de tese) e alguns alunos de mestrado (atualmente um aluno) e graduação (atualmente uma aluna), alocados em trabalhos específicos. Sobre os temas envolvidos nessa pesquisa, tem-se:

- **ECOSs:** envolve a organização de uma base de conhecimento inicial, considerando as diferentes dimensões exploradas pela comunidade e os desafios identificados para os próximos anos, a fim de organizar um *framework* para E&G de ECOSs;
- **Reutilização de Software & ECOSs:** engloba as técnicas de ED, LPS e de DBC para explorar as atividades de análise, projeto e implementação de ECOSs, bem como entender o seu impacto no futuro da reutilização;

- **Ontologia em ECOSs:** refere-se ao estudo de modelos conceituais para representação de ECOSs para organizar uma base de conhecimento para a comunidade de pesquisa no tema e viabilizar a aplicação na indústria;
- **Redes Sociotécnicas em ECOSs:** foca em pesquisas relativas ao uso e análise de redes sociais, técnicas e socio-técnicas tradicionais como instrumentos para definir e modelar de elementos sociais em ECOSs; e
- **Governança e Mercados de ECOSs:** engloba estudos relacionados a modelos de governança de TI com a finalidade de derivar um modelo específico para aplicações, componentes e serviços em ECOSs, bem como definir e modelar elementos de negócios em ECOSs;
- **Visualização e Métricas para ECOSs:** refere-se a estudos sobre técnicas de visualização com a finalidade de selecionar ou derivar um conjunto adequado para analisar ECOSs, que seja utilizado para manipular métricas de diferentes níveis, i.e., técnicas (baseadas em projetos) e de negócio ou sociais (baseada em redes sociotécnicas);
- **Impactos da Multidão na Estratégia e Inovação em ECOSs:** refere-se a estudos sobre as multidões no contexto dos ECOSs e da abertura das plataformas por meio de estratégia e de inovação no processo de ES orientado ao mercado, centrado na arquitetura e no ciclo de vida da TI.

1.5 Objetivos e Visão Geral da Proposta

Considerando o problema apresentado na Seção 1.3 e o contexto da pesquisa discutido na Seção 1.4, o objetivo desta proposta de tese é apresentar uma abordagem baseada em reutilização para dar suporte a E&G de ECOSs, explorando, para isso, o conceito de domínio no ambiente de DPE. O foco está em criar uma abordagem integrada para definição, modelagem e análise de conceitos e suas relações na visão interna do ECOS, mas também nas redes que se constituem a partir da visão externa. Com esta pesquisa, espera-se gerar uma melhor compreensão e a percepção do(s) ECOS(s) na indústria de software global e dinâmica por uma base de conhecimento para interessados, tratando suas três dimensões: *técnica*, *transacional* e *social*. Nessa direção, esta proposta pretende avaliar as hipóteses a seguir:

H1: O uso de conceitos de Reutilização de Software como instrumento para definição e modelagem de ECOSs melhora a compreensão de seus elementos e conexões.

H2: O tratamento de elementos de negócio e sociais no processo de abertura dos ECOSs melhora a análise das redes que se formam a partir de um ECOS.

A abordagem proposta, denominada *Reuse-based Software Ecosystems Engineering and Management (ReuseSEEM)*, descreve o ambiente de DPE, i.e., uma infraestrutura para instanciação e análise de ECOSs. Ao mesmo tempo em que adapta e aplica conceitos de reutilização para a compreensão de ECOSs, a abordagem *ReuseSEEM* almeja contribuir para esta área ao inserir ECOSs como uma nova abstração para entender a reutilização interorganizacional. A abordagem propõe um processo para tratar o domínio de ECOS, dividido em quatro fases, das quais somente as três primeiras pertencem ao escopo desta proposta de tese (Figura 1.4):

- 1) *definição*: organização inicial e utilização de uma base de conhecimento para o domínio de ECOSs, utilizando técnicas para criar um modelo conceitual a partir do qual os elementos e suas conexões em um ECOS serão instanciados (*análise de domínio de ECOSs*);
- 2) *modelagem*: representação dos elementos e conexões no ECOS através de um grafo de rede sociotécnica, que utiliza a base de conhecimento e o repositório de ativos de ECOSs para consultar elementos do senso de negócio e de comunidade (*projeto de domínio de ECOSs*);
- 3) *análise*: configuração ou recorte de um caso de ECOS a partir do seu modelo conceitual, sob o ponto de vista de um interessado no nível organizacional, da visão interna e/ou externa do ECOSs, a fim de explorar as suas três dimensões (*extração de subdomínio*);
- 4) *manutenção*: ciclos de estudos dos casos de ECOSs para manter o *framework* para E&G de ECOSs ao construir o repositório de dados históricos e relatos de experiência e realimentar a base de conhecimento e o repositório de ativos de ECOSs (*estratégia de pesquisa*).

A fim de viabilizar a definição e modelagem de ECOSs, o primeiro passo consiste em construir o *framework* para E&G de ECOSs com base nas iniciativas da comunidade de pesquisa e prática, além de analisar casos de ECOSs utilizando este arcabouço inicial. Busca-se organizar uma base de conhecimento inicial para ECOSs, que permita criar uma proposta de ECOSs baseada em reutilização, i.e., adaptar e/ou aplicar conceitos relacionados a domínios e a componentes. Assim, diante das inúmeras

definições de ECOSs, torna-se possível a esta proposta de tese apresentar uma definição de ECOSs que tenta levar em consideração um olhar sociotécnico sobre a ES e mantida por uma infraestrutura de apoio.

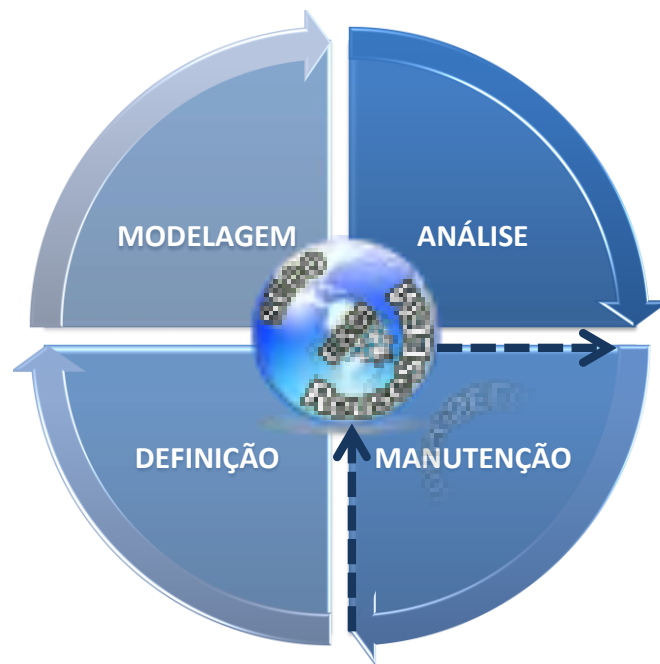


Figura 1.4 – Processo da abordagem *ReuseSEEM*

Após os primeiros passos, a base de conhecimento inicial deve ser reestruturada para permitir que ECOSs sejam instanciados. Para isso, a ela deve agregar um glossário de termos e significados de ECOSs e uma modelo que descreva os conceitos e suas relações. Além disso, um diagramador deve ser construído para representar os modelos conceituais gerados pela fase anterior na forma de RPSs, estendidas para as redes sociotécnicas. Por fim, as redes sociotécnicas devem permitir a seleção de elementos e suas conexões nos ECOSs a fim de gerar configurações ou recortes de casos de ECOSs. Dessa forma, busca-se explorar o potencial da graficabilidade, da interação e da navegabilidade indeterminada nas redes sociotécnicas, para ampliar a representação da ES. A solução posposta inclui a extensão da biblioteca de componentes Brechó (SANTOS *et al.*, 2010a; 2010b), de forma a permitir que os elementos de ECOSs sejam gerenciados ao longo do ciclo de vida. Para isso, serão implementados mecanismos estabelecidos pela abordagem, realizando-se as devidas adaptações tecnológicas, quando necessário.

A abordagem *ReuseSEEM* permite que as hipóteses *H1* e *H2* sejam avaliadas utilizando dois estudos experimentais (MAFRA & TRAVASSOS, 2006): (1) uma ***pesquisa de opinião***, para verificar o modelo de domínio de ECOSs, aplicando

entrevistas e questionários para especialistas no tema; e (2) um **experimento controlado**, para verificar a coesão e a estabilidade dos elementos e suas conexões nos ECOSs nas fases de definição, modelagem e análise, por meio de tarefas atribuídas aos participantes (pesquisadores e profissionais de ES), a partir de um repositório de ECOSs instanciado com exemplos. Após a construção e avaliação da abordagem, um estudo de caso em um contexto real poderá ser planejado após a conclusão da tese de doutorado.

1.6 Organização

Este exame de qualificação está organizado em outros quatro capítulos, além deste capítulo de introdução. O Capítulo 2 realiza um levantamento sobre conceitos e técnicas de Reutilização de Software que apoiam esta proposta de tese, entre eles, ED e DBC. A biblioteca de componentes e serviços Brechó também é discutida brevemente.

O Capítulo 3 apresenta uma linha do tempo da evolução da ES em direção aos ECOSs, do ponto de vista da reutilização, com o intuito de discutir a transição entre sistemas monolíticos, sistemas baseados em componentes, LPSs e ECOSs. A partir disso, os principais conceitos, definições, estruturas, características, fundamentos teóricos e desafios de ECOSs são discutidos.

O Capítulo 4 descreve a abordagem para E&G de ECOSs baseada em reutilização, que corresponde à solução desta proposta de tese. São apresentados os detalhes dos requisitos da abordagem, além do *framework* para E&G de ECOSs, inicialmente baseado nas iniciativas da comunidade de pesquisa no tema. Isso foi resultado de uma agregação de trabalhos publicados nas edições do IWSECO (2009-2012), refinado posteriormente por dois estudos secundários no tema. O *framework* foi utilizado em casos de ECOSs do Laboratório de Reutilização da COPPE/UFRJ (LENS/Reuse) a partir de suas três dimensões e os resultados permitiram a proposta da quarta dimensão de E&G, bem como de uma infraestrutura de apoio. Por fim, faz-se uma apresentação e comparação com os trabalhos relacionados.

O Capítulo 5 conclui este exame de qualificação, indicando as principais contribuições deste trabalho. As atividades em andamento e previstas, assim como os resultados preliminares e previstos, são expostos e mapeados a um cronograma para a execução desta proposta.

Capítulo 2 – Técnicas e Ambientes para Reutilização de Software

2.1 Introdução

De acordo com BIFFL *et al.* (2006), na Engenharia de Software (ES), a reutilização é ao mesmo tempo uma prática antiga e nova, pois os programadores reutilizam ideias, abstrações e processos desde os primórdios da Computação, ainda que de forma *ad hoc*. Os sistemas de software vêm agregando maior complexidade em sua construção frente à demanda pela qualidade de seus serviços, paralelamente às pressões do *time-to-market* e por um processo de desenvolvimento mais organizado e voltado efetivamente para a reutilização. Das *sub-rotinas* de 1960, dos *módulos* de 1970, dos *objetos* de 1980, dos *componentes* de 1990 e dos *aspectos* de 2000, a história da ES se desenrolou de forma ascendente para resolver a dualidade “crescente capacidade e aquecimento econômico” *versus* “crescente complexidade” (ALMEIDA *et al.*, 2007). Isso culminou com os *ecossistemas* de 2010 (SANTOS & WERNER, 2010a), pois os projetos de software têm se tornando mais dinâmicos e não controláveis, incluindo problemas clássicos com cronograma, custos e falhas em atender às proposições de valor dos envolvidos, e.g., conciliar os requisitos especificados e aqueles do sistema.

Em busca de uma sistemática, SAMETINGER (1997) afirma que a academia e a indústria investigam soluções para o processo de desenvolvimento baseado em reutilização desde a década de 1970, o que levou ao estabelecimento da área de Reutilização de Software nos anos 1980 (KRUEGER, 1992). Uma de suas técnicas corresponde ao Desenvolvimento Baseado em Componentes (DBC), cuja filosofia é “compre, não construa”. DBC consiste em uma abordagem para reutilização em que os componentes devem ser armazenados em repositórios voltados tanto para um cenário intra- e/ou interorganizacional como para um cenário competitivo de mercado (SANTOS & WERNER, 2009). Entretanto, WERNER & BRAGA (2005) reforçam que uma das maiores dificuldades está na criação de componentes que possam ser reutilizados em outras aplicações, o que requer uma sistemática voltada para um dado domínio, por meio da atividade de Análise de Domínio (AD). AD é uma tentativa de identificar objetos, operações e relacionamentos entre tudo o que os especialistas de um domínio julgam importante (NEIGHBORS, 1980). Pelo número de definições

apresentadas na literatura (BRAGA, 2000), constata-se a preocupação da pesquisa em ES por capturar, coletar, organizar e modelar o conhecimento dentro de um domínio de aplicação com base em dois pontos principais (PRIETO-DÍAZ & ARANGO, 1991):

- *a especificação para um domínio das informações reutilizáveis*: como identificar, obter e representar as informações dentro de um domínio;
- *a coesão e a estabilidade dos problemas de domínio*, onde:
 - a *coesão* se aplica na representação dos conhecimentos de um domínio para possibilitar a captura do conhecimento necessário para a solução de diversos problemas, sob a forma de um conjunto finito e pequeno de descrições reutilizáveis; e
 - a *estabilidade* torna possível a amortização do custo da captura e a representação de informação por meio de sua reutilização durante longos períodos em diversas situações e lugares.

Normalmente, a AD corresponde a uma das atividades que compõem outra técnica de reutilização mais ampla, a Engenharia de Domínio (ED), cuja meta é disponibilizar artefatos de software altamente reutilizáveis, de elementos de código fonte, ou binário, a estruturas de modelos, especificações, documentações etc. (KRUEGER, 1992). A ED tem uma ênfase mais sistematizada para AD, voltando-se para a construção de artefatos, criando um processo completo para a especificação de componentes reutilizáveis (ARANGO, 1988). A abstração é fundamental para viabilizar a reutilização, sendo especificada pela AD, da mesma forma que a efetiva realização da abstração em componentes reutilizáveis, que possam fazer parte do desenvolvimento de uma aplicação por meio das atividades de projeto e implementação de domínio (WERNER & BRAGA, 2005). ED passou a ser de grande interesse da indústria de software pela preocupação em realimentar o seu processo dinamicamente, o que derivou uma vertente chamada Linha de Produtos de Software (LPS). Segundo BOSCH (2000), um dos principais pontos fortes de uma LPS está na identificação das características comuns e variáveis entre os produtos, permitindo um aumento na flexibilidade e adaptação às necessidades dos clientes. De certo modo, este foco despertou o olhar da indústria sobre o tema Ecossistemas de Software (ECOSs), antes mesmo que a comunidade de ES pudesse empregar seus esforços (BOSCH, 2009).

Uma vez que as técnicas apresentadas podem ser utilizadas para a definição, modelagem e análise de domínios, a sua aplicabilidade também poderia ser explorada para o domínio de ECOSs, que carece de elementos para sua compreensão, o que

impacta o processo de tomada de decisão, e.g., “que tecnologia adotar?”, “que parceiros buscar?” e “a quais ECOSs pertencem?”. Assim, este capítulo apresenta um levantamento sobre conceitos de Reutilização de Software que podem apoiar esta proposta de tese. Entre as técnicas discutidas, estão ED e DBC, e entre os ambientes de apoio, a biblioteca de componentes e serviços Brechó. Para isto, o capítulo está organizado da seguinte forma: na Seção 2.2, uma visão geral sobre ED é exibida, contemplando os conceitos de LPS, ciclo de desenvolvimento, modelo de características, variabilidade e ontologia; na Seção 2.3, discute-se DBC e suas bases, e.g., classificação, armazenamento, evolução, arquitetura e mercados de componentes; a Seção 2.4 integra estas técnicas ao discutir critérios para apoiar DBC no contexto da ED; a Seção 2.5 mostra uma visão geral da biblioteca Brechó; finalmente, a Seção 2.6 conclui o capítulo com algumas considerações finais.

2.2 Engenharia de Domínio

ED corresponde a uma técnica do desenvolvimento *para* reutilização cuja abordagem visa definir escopo, especificar arquitetura e construir recursos para uma classe de sistemas, subsistemas ou aplicações (ISO/IEC, 1995; IEEE, 2004). Por outro lado, o desenvolvimento *com* reutilização compreende a Engenharia de Aplicação (EA) (MILER JÚNIOR, 2000), cujo foco está na reutilização de artefatos gerados pela ED. ED e EA funcionam juntas para promover a Reutilização de Software, ou seja, a ED provê um conjunto de artefatos que serão utilizados, ao passo que a EA constrói aplicações a partir da instanciação dos artefatos gerados na ED (OLIVEIRA, 2006). Além disso, a ED tem por base a AD, responsável por definir o seu objeto de estudo, i.e., o domínio do problema. Em certa comunidade, informações sobre o mundo real podem ser associadas em grupos de informações, ou domínio do problema, de modo que (SHAPER, 1974): (i) relacionamentos compreensíveis existentes entre os itens de informação sejam identificados com relação a alguma classe de problemas, (ii) os problemas tenham significado para os membros da comunidade, e (iii) haja o conhecimento necessário para a solução do problema.

Nesse sentido, duas perspectivas são atendidas: domínio como uma coleção de problemas reais e domínio como uma coleção de aplicações, existentes ou futuras, que compartilham características comuns (WERNER & BRAGA, 2005). Sob a primeira perspectiva, o resultado da AD deve ser uma teoria sobre os problemas do domínio,

visando apoiar a forma de pensá-lo, considerando predição, explicação ou derivação de fatos sobre o domínio, de difícil observação direta. Na segunda perspectiva, o resultado da AD corresponde a uma taxonomia de sistemas que demonstra as similaridades e diferenças entre os sistemas estudados, i.e., as variabilidades, de maneira a organizar os componentes e arquiteturas comuns a estes (ARANGO, 1994). Independente do caso, segundo BRAGA (2000), o desenvolvimento de software demanda cada vez mais (i) um alto grau de automação de suas atividades e do controle do processo; (ii) o uso de novas tecnologias, e.g., distribuição e componentização; (iii) a disponibilização de produtos de software que estejam em conformidade com os padrões adotados pelo mercado.

2.2.1. Linhas de Produtos de Software

Uma vertente da ED que vem sendo explorada pela indústria de software internacional é a LPS. Segundo NORTHROP (2004), uma LPS consiste em um conjunto de sistemas de software que compartilham um conjunto de características comuns e controladas, que satisfazem necessidades de um segmento de mercado em particular, e que são desenvolvidos a partir de um núcleo de artefatos (*core assets*), de forma predefinida. Esse núcleo inclui artefatos como modelo do domínio, arquitetura e componentes reutilizáveis que serão integrados para gerar os produtos, entre os quais a arquitetura é o elemento chave (GIMENES & TRAVASSOS, 2002). Uma LPS funciona como uma fábrica que instancia produtos com características similares, mas também apoia a sua variabilidade ao permitir a incorporação de características específicas que os distinguem, através da composição de componentes existentes (NORTHROP, 2004).

Como ilustra a Figura 2.1, uma LPS pode ser dividida em três atividades, cujo comprometimento deve ser estabelecido pela organização para que a LPS seja viável e executada com sucesso (TEIXEIRA, 2011): (1) *desenvolvimento do núcleo de artefatos*, base do paradigma de linha de produtos, inclui uma arquitetura de referência, componentes reutilizáveis, modelos de domínio, requisitos, cronogramas, orçamentos, planos e casos de teste, planejamento e descrição de processos etc.; (2) *desenvolvimento do produto*, utiliza os resultados produzidos pela atividade (1) e envolve subatividades como análise baseada no modelo de domínio, instanciação da arquitetura específica do produto em questão a partir da seleção incremental dos pontos de variabilidade de uma arquitetura genérica, e atividade de povoamento da arquitetura; e (3) *gerenciamento da linha de produtos*, visa supervisionar, coordenar e documentar as atividades (1) e (2).



Figura 2.1 – Atividades para o desenvolvimento da LPS. Fonte: (NORTHROP, 2004)

2.2.2. Ciclo de Desenvolvimento

ED deve ser vista como um conjunto de etapas, desde a identificação de funcionalidades e conceitos requeridos por uma família de sistemas através da AD, passando por sua representação em modelos genéricos, até o refinamento pelos estágios subsequentes de construção da infraestrutura de reutilização (WERNER & BRAGA, 2005). Quando combinada com a EA para viabilizar um mecanismo de realimentação dinâmica, origina um processo mais amplo, que evolui o ciclo de desenvolvimento das LPSs (Figura 2.2). A EA tem etapas similares às da ED, exceto pelo fato das atividades de modelagem das aplicações serem mais específicas, enfatizando as características de cada aplicação. Segundo OLIVEIRA (2006), durante a EA, realiza-se a seleção dos componentes necessários à aplicação em alto nível de abstração, por meio do modelo do domínio, descendo gradativamente até conseguir atingir os componentes do domínio implementados ou semidesenvolvidos, o que se denomina *recorte*. Esse recorte é influenciado pela variabilidade modelada no domínio, pois as características variáveis em artefatos do domínio determinam o tipo de aplicação que será instanciada. Isso implica em uma avaliação inicial do domínio, focando no desenvolvimento da aplicação almejada. A avaliação deve considerar as funcionalidades disponibilizadas no domínio e a afinidade da família de sistemas com o produto que será desenvolvido.

Como sumariza FERNANDES (2009) e TEIXEIRA (2011), as seguintes etapas constituem o processo de ED: (1) *AD*: o domínio e o seu escopo são identificados, determinando-se ainda as características comuns e variáveis de uma família de

aplicações; (2) *projeto do domínio*: os resultados da etapa (1) são utilizados para a construção de uma arquitetura adaptável, fruto da identificação e generalização de soluções para as características comuns, onde oportunidades de reutilização identificadas na AD são refinadas, de forma a especificar as restrições da arquitetura; (3) *implementação do domínio*: determinam-se mecanismos para a tradução dos requisitos e soluções resultantes das etapas (1) e (2) em modelos implementacionais, que incluem identificação, reengenharia e/ou construção, e manutenção de componentes reutilizáveis subjacentes; (4) *testes do domínio*: validam-se os componentes implementados na etapa (3); e (5) *gerenciamento dos produtos*: mantém-se o escopo da LPS, identificando os produtos que irão constitui-la como um todo.

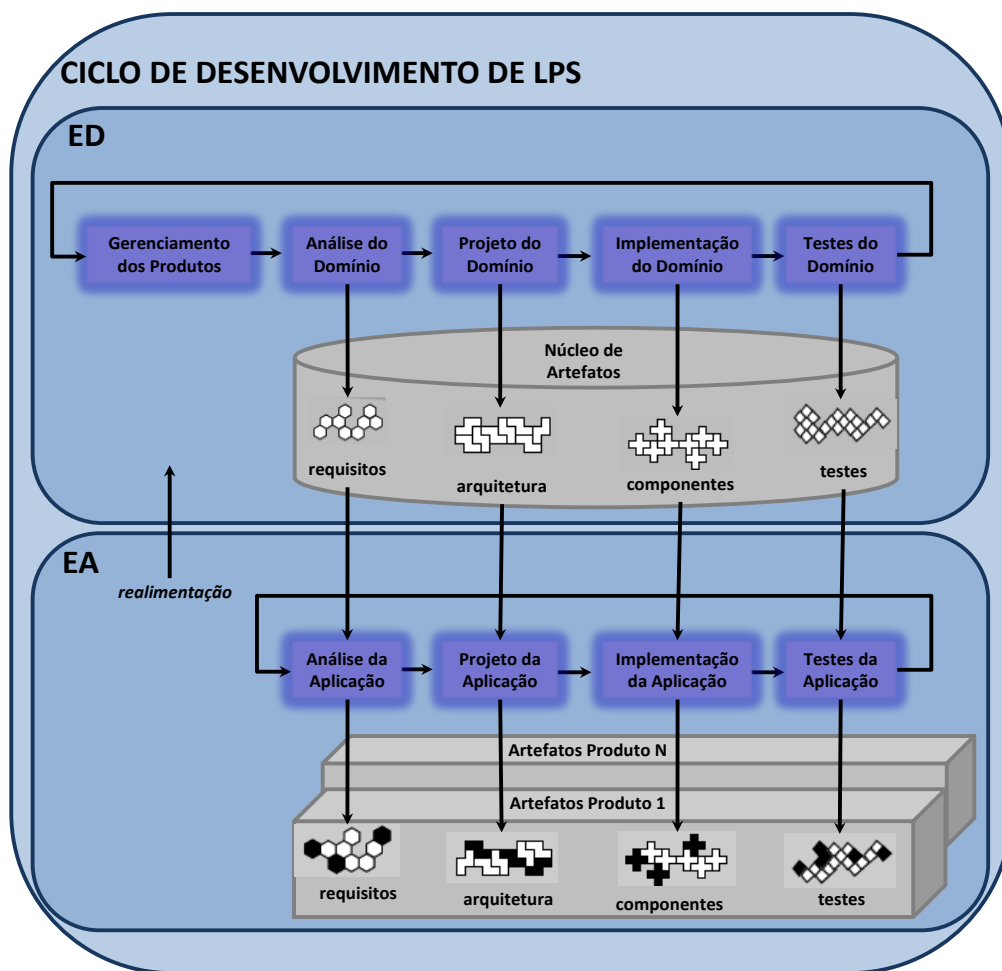


Figura 2.2 – Atividades do ciclo de desenvolvimento de uma LPS. Fonte: (FERNANDES, 2009)

2.2.3. Modelo de Características e Variabilidade

CZARNECKI & EISENECKER (2000) definem característica (*feature*) como um aspecto, uma qualidade, ou uma propriedade visível ao usuário, proeminente ou distinta, de um sistema ou vários sistemas de software. Normalmente, as características

são organizadas por meio de um diagrama que exibe uma estrutura hierárquica, como uma árvore ou grafo acíclico, dependendo da notação. A sua modelagem consiste em uma das atividades mais aceitas para representar produtos em uma LPS, por tratar a complexidade em expressar requisitos em forma de características (LEE & MUTHIG, 2006). Ou seja, a modelagem de características busca capturar, representar e gerenciar semelhanças, diferenças e suas relações em uma família de sistemas, a fim de facilitar a compreensão dos desenvolvedores e clientes sobre as capacidades gerais de um domínio, expressas através de características (KANG *et al.*, 1990). De acordo com BLOIS (2006), o modelo de características possui um alto nível de abstração, sendo considerado um elemento chave para relacionar os conceitos de mais alto nível aos demais artefatos de análise e projeto, além de um ponto de partida para o recorte necessário à instanciação de novos produtos. Sua função ainda está na padronização do entendimento do domínio entre os envolvidos, tais como usuários, especialistas de domínio e desenvolvedores (TEIXEIRA, 2011).

Nesse contexto, a variabilidade se torna um conceito chave para um processo de reutilização sistemática e de sucesso, pois consiste na forma de lidar com as diferenças entre os produtos e/ou aplicações e explorar as características comuns entre eles, como ressalta FERNANDES (2009). A alteração de decisões de projeto tomadas gera um alto custo no desenvolvimento, o que leva os engenheiros de software a postergarem-nas até as fases mais adiantadas, o que motiva o tratamento de variabilidade. Assim, permite-se documentar as possibilidades de configuração do software até o momento da tomada de decisão, o que aumenta o potencial de reutilização. Existem duas vertentes principais (OLIVEIRA, 2006): (i) *variabilidade no tempo*: a capacidade de um software dar suporte à evolução e à alteração de requisitos, nos seus vários contextos (Gerência de Configuração); e (ii) *variabilidade no espaço*, variabilidades entre os produtos ou aplicações individuais no domínio de uma LPS ou ED, em qualquer ponto no tempo. As características são classificadas quanto à *variabilidade* em pontos de variação, variantes ou invariantes, e quanto à *opcionalidade* em opcionais, mandatórias ou específicas de produto (Tabela 2.1) (OLIVEIRA, 2006; FERNANDES, 2009; TEIXEIRA, 2011).

Ao longo do ciclo de vida de uma LPS, por exemplo, as características podem variar em sua classificação (FERNANDES, 2009). Uma característica específica de produto pode vir a compor a arquitetura da LPS e ser incorporada por outros produtos, tornando-se uma característica variável, devido à evolução inerente da linha. Além disso, utiliza-se a *cardinalidade* para definir o número mínimo e máximo de

características que podem ser selecionadas de um conjunto de alternativas de um ponto de variação. Outro conceito é a especificação de *restrições* entre as características, i.e., relacionamentos de dependência e de mútua exclusividade. Por exemplo, no desenvolvimento de um novo produto em uma LPS, deve-se ter indicado a necessidade ou incompatibilidade da seleção conjunta de determinadas características. Isso deve ser verificado no processo de instanciação dos produtos como garantia da derivação de produtos consistentes em relação às especificações do domínio (OLIVEIRA, 2006).

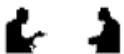
Tabela 2.1 – Classificação de características em ED e LPS

CATEGORIA	DESCRIÇÃO
<i>Pontos de Variação</i>	Pontos que refletem a parametrização no domínio de maneira abstrata e são configuráveis por meio das variantes, onde decisões são tomadas a respeito, por exemplo, de qual variante será utilizada.
<i>Variantes</i>	Alternativas de implementação disponíveis e necessariamente ligadas a um ponto de variação, visando configurabilidade.
<i>Invariantes</i>	Elementos fixos, i.e., não configuráveis no domínio. Não recebem nenhum tipo de nomenclatura especial.
<i>Opcionais</i>	Elementos que <i>podem</i> estar presentes nos produtos desenvolvidos por ED/LPS ou em aplicações instanciadas a partir de um domínio.
<i>Mandatórias</i>	Elementos que <i>devem</i> estar presentes em produtos desenvolvidos por ED/LPS ou em aplicações instanciadas a partir de um domínio.
<i>Específicas de Produto</i>	Elementos que estão presentes apenas em um produto específico e não fazem parte da arquitetura em ED/LPS.

Embora haja várias notações para representar a variabilidade no modelo de características (WERNER & BRAGA, 2005), será discutida a notação Odyssey-FEX (*Feature EXtended*), que estende os elementos do modelo de características do ambiente Odyssey (ODYSSEY, 2013), definidos originalmente por MILER (2000). A principal razão está no fato dela ter sido desenvolvida a partir das pesquisas realizadas pela Equipe de Reutilização da COPPE/UFRJ nos últimos 10 anos. Esta notação trata elementos que representam conceitos, funcionalidades e tecnologias, incluindo a variabilidade e os relacionamentos entre eles, com foco em ED. Isso se deve ao fato de que o objetivo desta proposta de tese não é contribuir com uma nova notação para modelagem de características, mas criar uma abordagem para tratar o domínio de ECOSs que, ao final, forneça elementos que possam ser incorporados por esta notação. Para a concepção da Odyssey-FEX, OLIVEIRA (2006) analisou um conjunto de notações mais utilizadas para o mesmo fim para resolver na Odyssey-FEX algumas deficiências existentes. A notação permite a disposição de características na forma de um grafo. Os tipos de características estão relacionados às etapas do desenvolvimento

de software e são mutuamente excludentes, não podendo uma característica pertencer simultaneamente a mais de uma categoria. Como mostra a Tabela 2.2, *domínio* e *entidade* são características de análise, ao passo que *ambiente operacional*, *tecnologia de domínio* e *técnica de implementação* são características de projeto.

Tabela 2.2 – Tipos de características na notação Odyssey-FEX. Fonte: (FERNANDES, 2009)

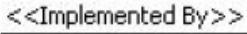
ÍCONE	CATEGORIA
	<i>Domínio</i> : característica ligada à essência do domínio (conceitual ou funcional). Representa uma funcionalidade e/ou conceito do modelo e corresponde a casos de uso e componentes estruturais concretos.
	<i>Entidade</i> : ator do modelo, i.e., entidade do mundo real que atua sobre o domínio (e.g., pode expor a necessidade de uma interface com usuário).
	<i>Ambiente Operacional</i> : representa atributos de um ambiente que uma aplicação do domínio pode usar e operar (e.g., sistemas operacionais).
	<i>Tecnologia de Domínio</i> : representa uma tecnologia utilizada para modelar ou implementar questões específicas de um domínio.
	<i>Técnica de Implementação</i> : representa uma tecnologia para implementar outras características, que pode ser compartilhadas entre domínios.

Conforme OLIVEIRA (2006), a notação Odyssey-FEX também apresenta dois tipos de regras de composição para expressar restrições existentes entre características, cuja semântica influencia o recorte para a instanciação dos produtos: (i) *inclusivas*, definem relações de dependência, indicando que duas ou mais características devem ser selecionadas juntas para um dado produto; e (ii) *exclusivas*, definem relações de mútua exclusividade, onde duas ou mais características não devem ser selecionadas juntas para um mesmo produto. As regras de composição têm a seguinte estrutura: *antecedente*, *palavra-chave* e *consequente*. A palavra-chave representa o tipo de regra, onde “requer” (*requires*) se refere às regras inclusivas e “exclui” (*excludes*), às regras exclusivas. Antecedente e consequente são expressões literais ou booleanas e representam uma característica do domínio (ou uma combinação delas).

Além disso, a notação aplica o conceito de cardinalidade e combina relacionamentos próprios com os da *Unified Modelling Language* (UML) para ter maior capacidade de representação e expressão (Tabela 2.3). Pelo fato dos relacionamentos não serem somente hierárquicos, pode-se gerar grafos para expandir o modelo em diversas direções. Esses conceitos são mostrados na Figura 2.3, como parte do exemplo discutido por FERNANDES (2009) sobre o domínio de guias turísticos móveis, cujo objetivo é auxiliar o turista durante a sua viagem. A característica *Visualizador*

Informações Cidade é uma característica invariante e mandatória, ao passo que a *Visualizador Mapas* é um ponto de variação opcional, configurado por duas características variantes do tipo funcional *Visualizador Imagem* e *Visualizador Mapa 3D*. A opcionalidade é indicada pela linha tracejada na representação da característica. Há relacionamentos da UML, como composição e agregação, e relacionamentos do tipo alternativo entre os pontos de variação e suas variantes com cardinalidade 0-1. O tipo da característica é representado tanto pelo ícone em marca d'água quanto pelo estereótipo acima do nome. Recentemente, algumas pesquisas estenderam o meta-modelo da notação Odyssey-FEX a fim de contemplar modelagem de características de sistemas sensíveis ao contexto (FERNANDES, 2009) e de características de processos de software (TEIXEIRA, 2011).

Tabela 2.3 – Relacionamentos na notação Odyssey-FEX. Fonte: (FERNANDES, 2009)

ÍCONE	CATEGORIA
 composição	Relacionamento em que uma característica é composta por outras, i.e., uma característica é parte fundamental de outra e a primeira não existe sem a segunda.
 agregação	Relacionamento em que uma característica representa o todo, e as demais, as partes, mas as características existem independentemente umas das outras.
 generalização	Relacionamento em que existe uma generalização ou especialização das características, denotando herança.
 associação	Relacionamento simples entre duas características, podendo ser nomeado para indicar um tipo específico de ligação.
 alternativo	Relacionamento entre um ponto de variação e as suas variantes, denotando a pertinência de uma variante a um determinado ponto de variação.
 implementado por	Relacionamento entre características de domínio e características tecnológicas, ou entre características tecnológicas de diferentes categorias.
 ligação de comunicação	Relacionamento entre características de entidade e características de domínio.

Por fim, vale ressaltar que tratar as diferenças entre produtos ou aplicações consiste em uma das questões essenciais analisadas quando artefatos reutilizáveis são construídos (WERNER & BRAGA, 2005). Apesar de alguns trabalhos focarem na fase de análise, de projeto ou de implementação de um domínio, a variabilidade modelada deve ser abordada em todas as etapas do desenvolvimento de uma família de aplicações, com uso de notação gráfica, textual ou mista (BLOIS, 2006).

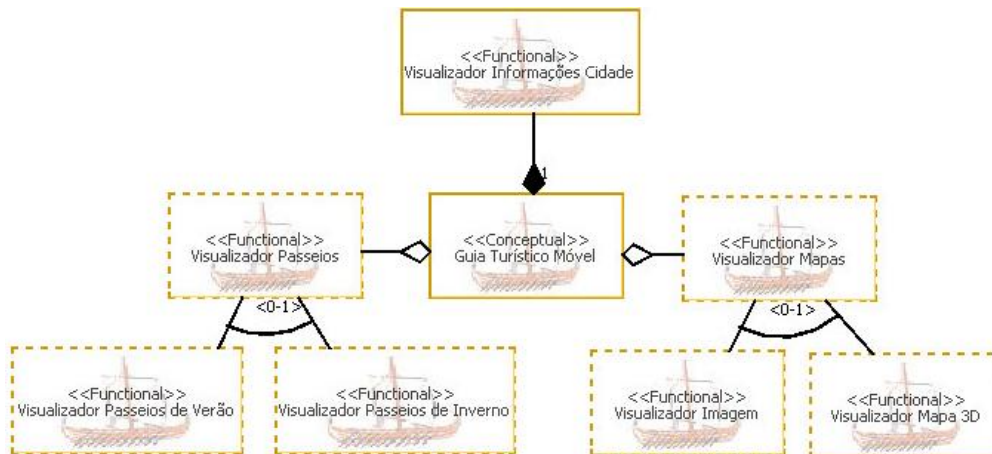


Figura 2.3 – Exemplo de modelo de características na notação Odyssey-FEX.
Fonte: (FERNANDES, 2009)

2.2.4. Engenharia de Domínio e Ontologias

Conforme discutido na Seção 2.1, a AD envolve a identificação, aquisição e análise do conhecimento do domínio que será reutilizado na especificação e construção de sistema, ou seja, visa produzir um modelo do domínio do problema. De acordo com PRIETO-DÍAZ & ARANGO (1991), este modelo serve como: (i) fonte de referência para tratar ambiguidades que surgem na análise de problemas ou mesmo durante a implementação de componentes reutilizáveis; (ii) repositório para aprendizagem e comunicação sobre o conhecimento compartilhado; e (iii) especificação para os desenvolvedores desses componentes. O conhecimento é a peça chave e pode estar embutido não apenas em documentos e repositórios, mas também em rotinas, processos, práticas e normas organizacionais (LOPES, 2010), de modo que a sua infraestrutura de armazenamento deva ser composta por uma base de conhecimento utilizando ontologias (O'LEARY, 1998). Uma ontologia consiste em uma especificação formal e explícita de uma conceitualização compartilhada (GRUBER, 1993).

Segundo FALBO *et al.* (2002), as ontologias podem ter um papel importante no contexto da ED, embora existam entraves quanto à sua ampla utilização na ES devido à carência de abordagens que as insiram no processo de desenvolvimento tradicional, e.g., para derivar modelos de objetos de ontologias a fim de derivar ativos reutilizáveis. No entanto, independente da variedade de formatos, uma ontologia inclui um glossário de termos e uma especificação de seus significados, i.e., definições e indicações de como os conceitos estão inter-relacionados (USCHOLD, 1998). Assim, constrói-se a estrutura de um domínio de interesse e limita-se a interpretação dos conceitos envolvidos, de

modo que a ontologia represente objetos e relacionamentos, bem como suas propriedades, atributos e funções e especificando o uso de regras de restrição (GRUBER, 1993). A AD pode se beneficiar nesse sentido, uma vez que modelos de características são mais intuitivos e simples de construir, mas o uso de ontologias para representar LPSs permite maior formalismo e alto poder de descrição semântica através de uma linguagem padrão (COSTA, 2013).

Entre outras motivações para o uso de ontologias, destacam-se (ABECKER *et al.*, 1998): (i) ajuda os pesquisadores e profissionais a compreenderem melhor uma determinada área de conhecimento, por demandar reflexão; (ii) ajuda-os também a atingir um consenso no seu entendimento sobre a área, como demanda o tema ECOSs (HANSSEN & DYBA, 2012); e (iii) ajuda os demais envolvidos na compreensão da área, evitando a constante dependência de especialistas. FERREIRA FILHO *et al.* (2012) citam ainda a possibilidade de recuperação de informação, de inferência e de rastreabilidade, dado que o conhecimento é formalmente representado e, dessa forma, passível de ser consultado por ferramentas inteligentes, o que viabiliza a extração de subdomínios por meio da criação e combinação de conceitos existentes (ABECKER *et al.*, 1998). Tais ferramentas podem apoiar várias aplicações de ontologias, classificadas em quatro categorias não mutuamente excludentes (JASPER & USCHOLD, 1999):

- *criação neutra*: uma ontologia é desenvolvida em uma única linguagem, que possa ser traduzida para diferentes formatos e usada para múltiplas aplicações;
- *ontologia como especificação*: uma ontologia de um dado domínio é criada e fornece um vocabulário para especificar requisitos para uma ou mais aplicações, tida como um modelo de domínio, o que permite a reutilização de conhecimento;
- *acesso comum à informação*: uma ontologia é construída para que múltiplas aplicações ou pessoas tenham acesso a fontes de informação heterogêneas, expressas por vocabulário diverso ou formato inacessível;
- *busca baseada em ontologia*: uma ontologia é utilizada para buscar informações em um repositório, a fim de melhorar a precisão e reduzir o tempo gasto nesta atividade.

Segundo COSTA (2013), uma vez que as ontologias são utilizadas para modelar um domínio específico, elas podem melhorar ainda a modelagem de características. Por

exemplo, as ontologias podem fornecer informações adicionais para o domínio de uma LPS, sobretudo quanto à especificação de restrições entre as características existentes. Em ontologias descritas em uma linguagem apropriada, as características podem ser expressas como classes desta ontologia, ao passo que a forma como elas se relacionam e suas restrições de uso podem ser expressas por propriedades ligadas às classes. Dessa forma, a combinação entre modelos de características e ontologias pode melhorar semanticamente a seleção de características desejáveis no desenvolvimento.

No contexto desta proposta de tese, cujo enfoque é interferir no problema de compreensão de ECOSs utilizando conceitos de reutilização, o domínio de ECOS pode se beneficiar dessa combinação. Ou seja, como ECOSs manipulam elementos sociais e de negócio no processo da ES e a LPS, por exemplo, trabalha sobre elementos técnicos, o uso de ontologias para modelar ECOSs pode representar um caminho tratar os entraves à reutilização em um nível de abstração mais alto. Assim, os impactos de um modelo conceitual de ECOS podem ser estendidos para modelos de características de LPS, contribuindo para os desafios da ES econômica e social (BOEHM, 2006).

2.3 Desenvolvimento Baseado em Componentes

O estabelecimento da área de reutilização nas últimas três décadas provocou na ES a expansão das práticas de desenvolvimento que privilegiam a separação entre interfaces e implementações, o que levou seus métodos e técnicas a se basearem no conceito de componentes (FRAKES & KANG, 2005). Entre os fatores que estimulam o DBC, destacam-se: (i) o avanço da Internet e da computação distribuída; (ii) a atualização de sistemas legados de plataforma alta (*mainframes*) para plataforma baixa, utilizando subsistemas interoperáveis; e (iii) o amadurecimento de padrões de infraestrutura para desenvolver aplicações baseadas em componentes (WERNER & BRAGA, 2000). Segundo BRAGA (2000), *componentes* podem ser definidos como artefatos autocontidos, claramente identificáveis, que descrevem ou realizam uma função específica e têm interfaces claras, em conformidade com um dado modelo de arquitetura de software, documentação apropriada e um grau de reutilização definido. GUERRA & COLOMBO (2009) resumizam as principais propriedades dos componentes a partir da proposta de SAMETINGER (1997), conforme a Tabela 2.4.

2.3.1. Classificação

De acordo com SOTEX (2007), componentes pode ser classificados em três grupos, considerando o grau de reutilização e a sua especificidade: (i) *componentes genéricos*, de uso comum em muitos sistemas, e.g., componentes de interface com os usuários; (ii) *componentes de serviços*, que fornecem serviços especializados, embora não sejam específicos a um domínio, e.g., componentes para tratamento de erros em comunicação de dados, criptografia, segurança, geração de gráficos etc.; e (iii) *componentes de domínio*, específicos, que implementam regras de negócios simples a complexas, e.g., regras do setor financeiro ou da construção civil. Sob o ponto de vista de SZYPERSKI *et al.* (2002), a Reutilização de Software com DBC pode ser dividida em quatro categorias, apesar de apenas as três primeiras serem mais comuns na ES:

- *reutilização de código fonte*: trechos de código são utilizados durante a implementação de um novo produto (“copiados e colados”);
- *reutilização de partes de software*: reutilização da arquitetura e da implementação de fragmentos de produtos de software em diferentes projetos, por meio de um processo de desenvolvimento mais elaborado, embora não sejam de fácil substituição (sem identificação clara);

Tabela 2.4 – Propriedades de um componente. Fonte: (SANTOS, 2010)

PROPRIEDADE	DESCRIÇÃO
<i>Autocontido</i>	Componentes podem ser reutilizáveis sem a necessidade de incluir ou depender de outros. Em caso de dependência, todo o conjunto deve ser abstraído como um componente reutilizável.
<i>Identificação</i>	Componentes devem ser facilmente identificados, i.e., devem estar armazenados em um lugar único, e não espalhados e misturados com outros artefatos de software ou documentação.
<i>Funcionalidade</i>	Componentes têm uma funcionalidade clara e específica que realizam e/ou descrevem. Podem realizar funções ou podem ser simplesmente descrições de funcionalidades.
<i>Interface</i>	Componentes devem possuir interfaces claras, que indicam como estes podem ser reutilizados e conectados a outros, devendo-se ocultar os detalhes desnecessários para a sua reutilização.
<i>Documentação</i>	A existência de documentação é indispensável para a reutilização. O tipo de componente e a sua complexidade indicarão a conveniência do tipo de documentação.
<i>Condição de reutilização</i>	Componentes devem ser mantidos para preservar a reutilização sistemática, cuja condição envolve informações variadas, e.g., quem é o proprietário, quem deve ser contatado em caso de problema, qual é a situação de qualidade, etc.

- *integração dinâmica de componentes de diversas fontes*: a reutilização ocorre na implementação do produto de software, quando o sistema está desenvolvido e novas funcionalidades são acrescentadas, do tipo *plug-in*;
- *componentização*: mais complexa, preconiza que a atualização, a extensão do sistema e a integração possam ocorrer dinamicamente, permitindo que DBC aconteça além das fronteiras das organizações (base para a potencial revolução que pode acontecer pelo uso de componentes).

2.3.2. Repositórios

Para que o desenvolvimento *com* reutilização aconteça, um elemento importante é o repositório de componentes, definido por SAMETINGER (1997) como uma base preparada para armazenar e recuperar componentes. Com o objetivo de garantir eficiência e eficácia nestas atividades, ressalta-se a importância do armazenamento de informações adicionais, dado que a chance dos engenheiros de software reutilizarem componentes previamente construídos se relaciona diretamente com sua disponibilidade e acesso em um repositório. Ao tratar a facilidade para localizar e recuperar componentes, distinguem-se três tipos de repositório (SAMETINGER, 1997): (i) *repositórios locais*, que armazenam componentes de propósito geral em um único repositório; (ii) *repositórios específicos de domínio*, que armazenam componentes específicos, com escopo bem definido, podendo fornecer componentes alternativos para as mesmas tarefas; e (iii) *repositórios de referência*, que auxiliam na busca por componentes em outros repositórios, como uma espécie de páginas amarelas.

O potencial da reutilização pode ser ampliado pelo acesso a diversos repositórios espalhados e conectados à Internet. Além do acesso a um repositório central, interno à organização, os engenheiros de software podem utilizar componentes provenientes de organizações parceiras e/ou de um contexto de *free and open source* (FOSS). Naturalmente, direitos de propriedade ganham foco neste último, a depender do tipo de repositório, i.e., privado, comercial, não lucrativo, governamental ou de domínio público (SZYPERSKI *et al.*, 2002). Adicionalmente, é interessante destacar alguns fatos que persistem até hoje sobre a evolução de repositórios (BANKER *et al.*, 1993; SANTOS, 2010): (i) o nível de reutilização não varia diretamente com o número de componentes disponíveis; (ii) a maior parte dos componentes é reutilizada dentro do mesmo projeto; e (iii) os engenheiros de software tendem a reutilizar componentes que

foram desenvolvidos por eles mesmos. Isso sugere que mecanismos de busca providos por repositórios são inadequados, ou não são complementarmente explorados.

SEACORD (1999) afirma que repositórios tradicionais, i.e., locais, de componentes genéricos e de larga escala, historicamente falharam, sobretudo em decorrência da sua concepção como sistemas centralizados e “inchados” (WERNER & BRAGA, 2005). Podem-se obter melhores resultados com repositórios específicos de domínio e/ou de referência, pelas oportunidades da Internet e, sobretudo, com o uso de ECOSs (i.e., reutilização interorganizacional). Ou seja, a possibilidade de interconectar repositórios heterogêneos e distribuídos, evoluídos a partir da comunidade subjacente, contribui para o crescimento de repositórios de referência (SANTOS & WERNER, 2010a). Ao mesmo tempo, deve-se considerar um processo de produção-gerência-consumo, que inclui alguns papéis básicos, conforme ilustra a Figura 2.4 (SAMETINGER, 1997; SZYPERSKI *et al.*, 2002; ALMEIDA *et al.*, 2007; WERNER *et al.*, 2007): (i) *produtores*, que focam na construção e publicação de componentes, i.e., desenvolvimento *para* reutilização; (ii) *consumidores*, que focam em recuperar e utilizar componentes, i.e., desenvolvimento *com* reutilização; e (iii) *gerentes de reutilização*, que focam em organizar e gerenciar a qualidade e a disponibilidade dos componentes.

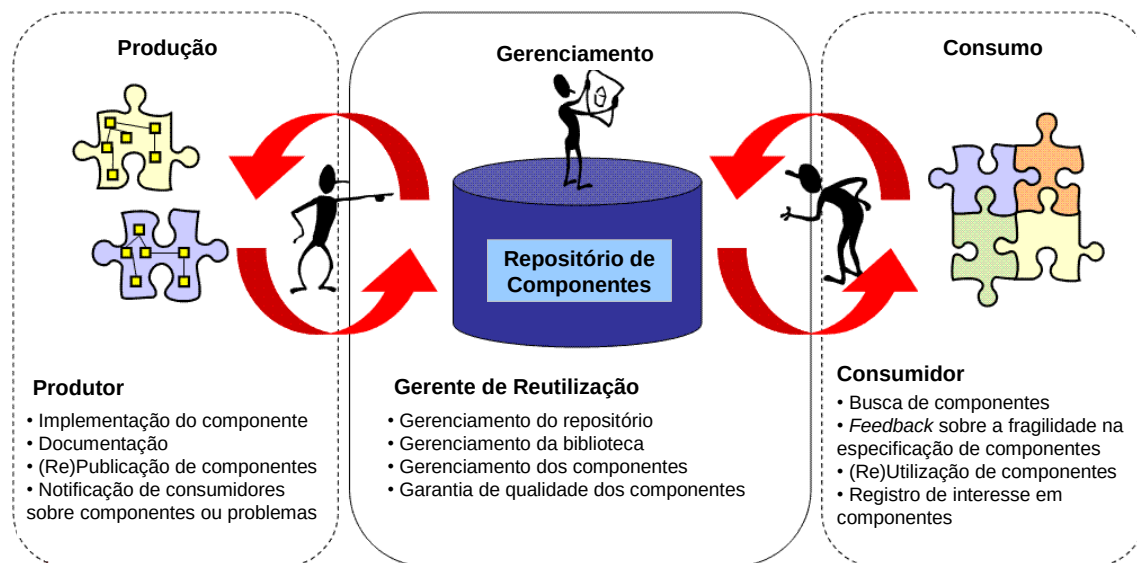


Figura 2.4 – Processo de produção-gerência-consumo em DBC. Fonte: (ALMEIDA *et al.*, 2007)

Conforme WERNER & BRAGA (2005), quando se trata de repositório, surge uma questão: “como se dão a busca e a seleção desses componentes armazenados?”. Isso tem relação direta com a qualidade dos mecanismos de classificação dos componentes, de modo que explorar DBC em larga escala, como em ECOSs, demanda mecanismos eficazes de busca, aquisição e compreensão de componentes. Além disso,

LUCRÉDIO *et al.* (2004) enfatizam que métodos, ferramentas, modelos de componentes e outras tecnologias têm pouca utilidade, caso não exista um mercado de componentes. No que tange à documentação, as categorias de componentes podem se basear em *vocabulário controlado* ou *não controlado* (BARROS, 1995). No primeiro caso, utilizam-se termos predefinidos para descrever um componente, e no segundo caso, qualquer termo pode ser utilizado abertamente. Como exemplos de estratégias de vocabulário não controlado, têm-se as buscas por texto livre ou palavras-chave, e, como exemplos de estratégias de vocabulário controlado, os pares de atributos, a classificação enumerada/hierárquica de termos ou a técnica de facetar (i.e., conjunto de dimensões e pontos de vista baseados em um vocabulário específico de domínio). Além disso, abordagens que exploram o potencial da *Web* e a natureza distribuída dos componentes, e.g., técnicas de aprendizado, introdução de filtros e personalizações, podem apoiar a classificação de componentes (BRAGA, 2000).

2.3.3. Arquiteturas, Estilos e Padrões

Os componentes podem ser organizados em uma arquitetura dividida em camadas, onde eles possuem diferentes níveis de abstração e as camadas inferiores provêm serviços para as camadas superiores (BLOIS, 2006). Além disso, o modelo de componentes deve especificar as dependências entre eles por meio de suas interfaces providas e requeridas. A abordagem de BROWN (2000) tem por base três camadas: (i) *componentes de negócio*, que implementam um conceito ou processo de negócio autônomo; (ii) *componentes utilitários*, que prestam serviços genéricos, necessários ao desenvolvimento das aplicações; (iii) *componentes de infraestrutura*, que estabelecem a comunicação com as plataformas de execução do software.

No contexto de ED, uma arquitetura deve se preocupar em propor uma solução estruturada para os requisitos de uma família de aplicações ou produtos de um dado domínio, denominada arquitetura de software específica para domínio (*Domain Specific Software Architecture* ou DSSA), cujos elementos constituintes são (XAVIER, 2001): (i) um modelo de domínio; (ii) os requisitos de referência, i.e., requisitos funcionais e não funcionais do domínio, sendo que estes estabelecem restrições sobre a arquitetura e a implementação; (iii) uma arquitetura de referência, i.e., arquitetura para uma família de aplicações construída na fase de projeto do domínio; e (iv) um processo de desenvolvimento que apoia a ED.

Especificamente sobre arquiteturas de referência, BLOIS (2006) conclui que grande parte das iniciativas de sua construção reforça o uso de linguagens de descrição arquitetural, estilos arquiteturais e padrões arquiteturais. Uma linguagem de descrição arquitetural (*Architecture Description Language* ou ADL) consiste em uma notação formal para representar arquiteturas de software em alto nível de abstração, inclusive aplicadas no contexto de arquiteturas baseadas em componentes (MENDES, 2002). Por sua vez, os padrões arquiteturais exibem um esquema de organização estrutural de sistemas, em que são definidos subsistemas, relacionamentos, regras e orientações para bom uso através de uma linguagem de padrões, i.e., nome do padrão, contexto, problema, solução, estrutura, usos conhecidos e padrões relacionados (BUSCHMANN *et al.*, 1996). Finalmente, um estilo arquitetural define um conjunto de regras de projeto que identificam tipos de componentes, conectores e restrições para sua composição, que podem ser utilizados para compor uma família de sistemas e subsistemas.

Componentes se comunicam via conectores, os quais possuem a especificação de um protocolo que define as suas propriedades, e.g., os tipos de interfaces que ele pode intermediar e a ordem em que os eventos acontecem (SHAW & GARLAN, 1996). Dessa forma, os estilos arquiteturais compreendem um vocabulário comum, e.g., tipos de componentes, tipos de conectores, estruturas de controle, comunicação de dados, interação entre dados e controle, regras e restrições. Nesta proposta de tese, a ideia está em adaptar e aplicar estes conceitos de arquitetura de referência, padrões e estilos para compreender definir e modelar ECOSs, analisando as características e comportamentos de seus elementos e respectivos relacionamentos.

2.3.4. Mercados

Há sinais de uma indústria de tecnologia de componentes desde 2000, observados em nichos de mercado bem definidos, incluindo componentes na forma de mercadoria (*commodity*), LPSs, infraestruturas (*frameworks* e ferramentas), intermediários (*brokers*) e consultores em DBC (BASS *et al.*, 2000; FRAKES & KANG, 2005; BOEHM, 2006; SANTOS & WERNER, 2010b). TRAAS & HILLEGERSBERG (2000) definem o mercado de componentes como um ambiente eletrônico baseado na Internet, representando um canal mais apropriado para comprar/adquirir e vender/publicar componentes proprietários e FOSS, promovendo sua ampla distribuição. Surgem da interação entre organizações especializadas em

desenvolver, manter e atualizar componentes, e seus clientes diretos, i.e., engenheiros de software (HILLEGERSBER *et al.*, 2001). Os clientes têm opções de adquirir componentes de diferentes fornecedores, desde que sigam o mesmo modelo, contrato e interface; caso contrário, torna-se difícil realizar comparações de mercado que levam em conta apenas os aspectos técnicos.

No que diz respeito às vantagens produtivas de natureza técnica e econômica (e.g., custos, escala, tempo, qualidade, facilidade operacional etc.), a visão dos ECOSs sugere que o mercado de componentes vai se desenvolver de alguma forma e em algum momento, considerando suas derivações, i.e., aplicações, sistemas e serviços (SANTOS & WERNER, 2010a). Os sistemas baseados em componentes prometem uma vantagem: a multiplicação de investimento e inovação (SZYPERSKI *et al.*, 2002). No entanto, conforme exibido na Seção 2.3.1, a componentização em larga escala ainda não é uma realidade devido à falta de um entendimento sólido sobre componentes, o que impede que DBC reúna um número significativo de adeptos quando se trata de um cenário competitivo, fora das fronteiras da organização (SANTOS & WERNER, 2012b). Investimento, distribuição e economia de escala para produto de software também se tornam fatores críticos na criação do mercado de componentes (WERNER *et al.*, 2009).

Os mercados obtêm sucesso onde a oferta e a demanda estão balanceadas, i.e., um novo produto pode criar um nicho se a sua chegada já é esperada ou se a sua existência gera e atende a uma demanda. Caso contrário, o custo de criação de sua demanda pode ser proibitivo (SZYPERSKI *et al.*, 2002). Por exemplo, uma nova técnica para tratamento de resíduos gerados pelas indústrias, baseada na magnetização de moléculas, que apresente desempenho menor ou igual às tradicionais técnicas baseadas em filtro de carvão ativado não encontra espaço na indústria, embora tenha valor como pesquisa, passível de patente. No caso de componentes, os consumidores potenciais iniciam uma busca a partir de necessidades que dizem respeito ao *que* é desejado e não *como* é realizado, o que pode levar ao não atendimento da proposição de valor. Além disso, a preocupação destes envolvidos passa a ser com a utilização dos componentes no desenvolvimento de software ou de outros componentes, se realmente têm sua qualidade garantida. Dessa forma, algumas questões relevantes devem ser tratadas pelo responsável pela integração, tais como seleção, interoperabilidade, controle de qualidade e manutenção (BRERETON & BUDGEN, 2000).

Conclui-se que as atividades de documentação, publicação, busca, recuperação, integração e manutenção de componentes são importantes para o sucesso de um

ambiente ou mercado, considerando um cenário intra- e interorganizacional. O suporte de ferramentas se torna um fator crítico para atividades de integração e de manutenção, especialmente a visualização, compreensão e avaliação dos riscos e dos impactos de decisões tomadas (LUQI & GUO, 1999). Por fim, SANTOS *et al.* (2010a) apontam que a definição de valor para componentes lidera o *ranking* de barreiras à componentização, o que requer identificar e modelar dados sobre a evolução de repositórios a fim de verificar se as necessidades dos envolvidos no mercado são satisfeitas. Outras questões devem ser tratadas, e.g., legais, econômicas, organizacionais e motivacionais, através da extração e visualização de informações (SANTOS, 2010).

2.4 Critérios para Apoiar o Desenvolvimento Baseado em Componentes no Contexto da Engenharia de Domínio

Após analisar diversos métodos de ED e DBC, bem como as suas limitações para DBC e modelagem de variabilidades, BLOIS (2006) propõe um novo conjunto de critérios para apoio de DBC no contexto de ED:

- *modelagem de variabilidades e opcionalidades em diferentes níveis de abstração*: especificada na fase de levantamento de requisitos, por meio de modelos de características, esta modelagem deve ser especificada também nos demais artefatos do domínio, mantendo a semântica;
- *manutenção da ligação e consistência entre os artefatos do domínio de diferentes níveis de abstração*: deve ser mantida a ligação existente entre artefatos gerados pela ED em diferentes níveis de abstração do domínio, relacionados a um mesmo conjunto de requisitos. Isso permite indicar que a consistência de um requisito do domínio e a sua respectiva variabilidade e opcionalidade são mantidas nos demais artefatos;
- *organização dos artefatos para composição de elementos arquiteturais*: métodos de ED e DBC devem sistematizar a organização de artefatos em elementos arquiteturais, os quais compõem uma arquitetura de componentes, com uso de critérios, métricas ou outras técnicas para apoio à criação destes elementos;
- *uso de princípios de DBC na modelagem do domínio*: indica o uso de uma abordagem de DBC por parte dos métodos de ED, pois DBC dá

suporte a várias características técnicas que promovem à reutilização (e.g., acoplamento e coesão);

- *necessidade de processos de ED com apoio ao DBC*: necessidade de um processo que apoie efetivamente todas as etapas da ED com base em princípios de DBC. Nenhuma especificação de artefatos deve ser feita com base em alguma tecnologia de componentes específica;
- *geração de código fonte na linguagem de programação empregada por uma tecnologia de DBC*: modelos de análise e projeto de domínio devem ser gerados sem vínculo tecnológico a fim de aumentar o potencial de reutilização, de modo que alguma sistematização apoie a geração de código fonte na tecnologia de componentes escolhida;
- *modelagem e reutilização de artefatos e modelos através de ferramental apropriado*: sugere a necessidade de uma infraestrutura de reutilização que possa dar suporte integrado ao ferramental de apoio aos critérios anteriores. Caso não exista, isso pode prejudicar a reutilização dos artefatos, limitando-se à uma etapa/atividade específica de um processo.

2.5 Biblioteca Brechó

Conforme enfatizado na Seção 2.3.4, quando se trata de mercado, a garantia da qualidade dos artefatos de software em geral é importante para o sucesso. Nessa direção, pesquisas e soluções foram desenvolvidas para incorporar e fornecer produtos de software e serviços relacionados de qualidade. Na indústria de software, especificamente em sistemas baseados em componentes, os repositórios correspondem às entidades que gerenciam e mantêm os ativos do mercado, i.e., componentes. Modelos de qualidade foram estendidos, visando atender a capacidade e a maturidade de processos na área de reutilização, em um cenário intra- e interorganizacional, tal como o Modelo de Referência MPS para Software (SOFTEX, 2012). No caso da Reutilização de Software, o desenvolvimento *com* e *para* reutilização são contemplados por dois processos específicos, o processo *Gerência de Reutilização* (GRU) e o processo *Desenvolvimento para Reutilização* (DRU), respectivamente. Pelo fato de que a solução posposta inclui a extensão da biblioteca Brechó, voltada para o GRU (WERNER & SANTOS, 2009; 2010), uma visão geral é apresentada a seguir.

A biblioteca Brechó faz parte do Projeto Brechó (BRECHÓ, 2013), desenvolvido pelo Grupo de Reutilização de Software da COPPE/UFRJ desde 2005, que visa pesquisar tópicos relacionados a repositórios e à indústria de componentes e serviços. Consiste em um sistema de informação Java *web* com uma base de dados de componentes e serviços, produtores e consumidores, e conta com mecanismos de armazenamento, documentação, publicação, busca e recuperação, cuja tela inicial é exibida na Figura 2.5. Considera ainda um conceito flexível de componente, que inclui todos os artefatos produzidos no desenvolvimento, e.g., processo, modelos, manuais, código fonte, binário, testes etc., o que permite a aquisição de diferentes conjuntos de artefatos empacotados ou disponibilizados como serviços (quando possível), atrelados a licenças personalizadas e configuráveis. A estrutura de documentação se baseia em categorias hierarquizadas e formulários dinâmicos e configuráveis associados a elas, que apoiam a construção da documentação de componentes na forma de um mosaico, dado que eles podem pertencer a várias categorias ao mesmo tempo. A biblioteca Brechó também gerencia um mapa de reutilização e o estabelecimento e inferência de dependências entre componentes.

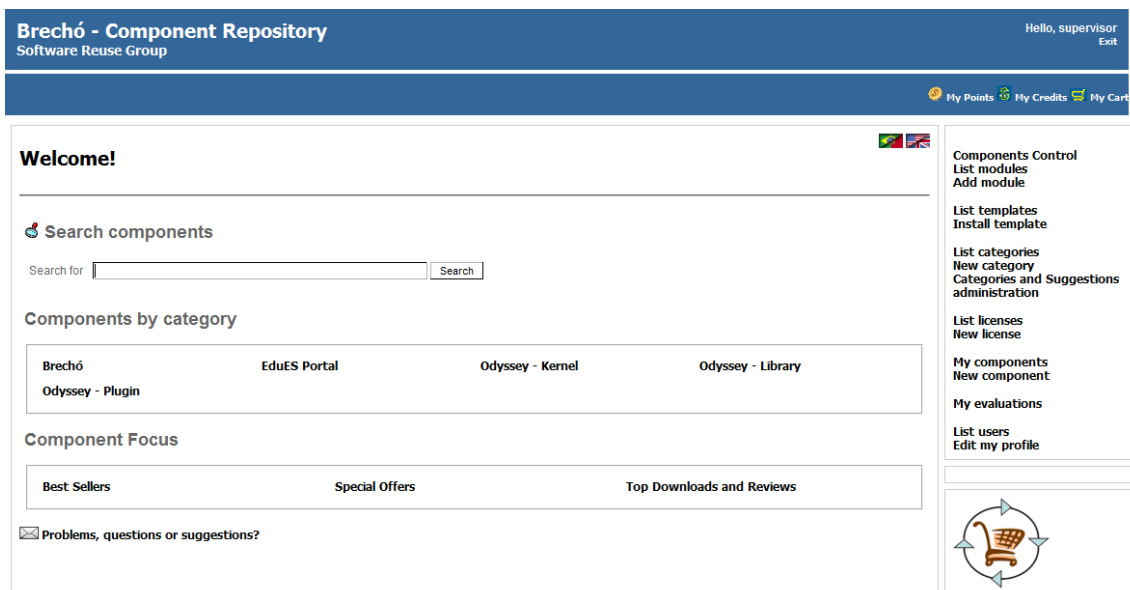


Figura 2.5 – Tela inicial da biblioteca Brechó. Fonte: (SANTOS, 2010)

A organização interna da biblioteca Brechó é dividida em níveis, conforme exibe a Figura 2.6. O primeiro nível, *Componente*, representa conceitualmente as entidades armazenadas na biblioteca. O nível seguinte, *Distribuição*, representa um corte funcional sobre as entidades, fornecendo conjuntos de funcionalidades que são desejadas por grupos específicos de usuários. O terceiro nível, *Release*, representa um

corte temporal sobre as distribuições, definindo as versões dos artefatos que implementam as entidades em um dado instante no tempo. A partir deste nível, as entidades passam a ter informações concretas sobre suas implementações, em diferentes níveis de abstração e empacotamentos distintos podem ser definidos para permitir a reutilização efetiva de parte destes níveis. O nível *Pacote* permite que seja feito um corte em níveis de abstração, possibilitando o agrupamento de artefatos de acordo com um público alvo. Por sua vez, o nível *Serviço* é um nível de abstração específico que viabiliza a reutilização de uma *release* através de serviços *Web*. Por fim, o nível *Licença* possibilita a definição de direitos e deveres sobre pacotes e serviços. Para cada pacote ou serviço, são estabelecidas licenças específicas, que garantem direitos e deveres de produtores e consumidores, devendo uma delas ser escolhida para a reutilização.

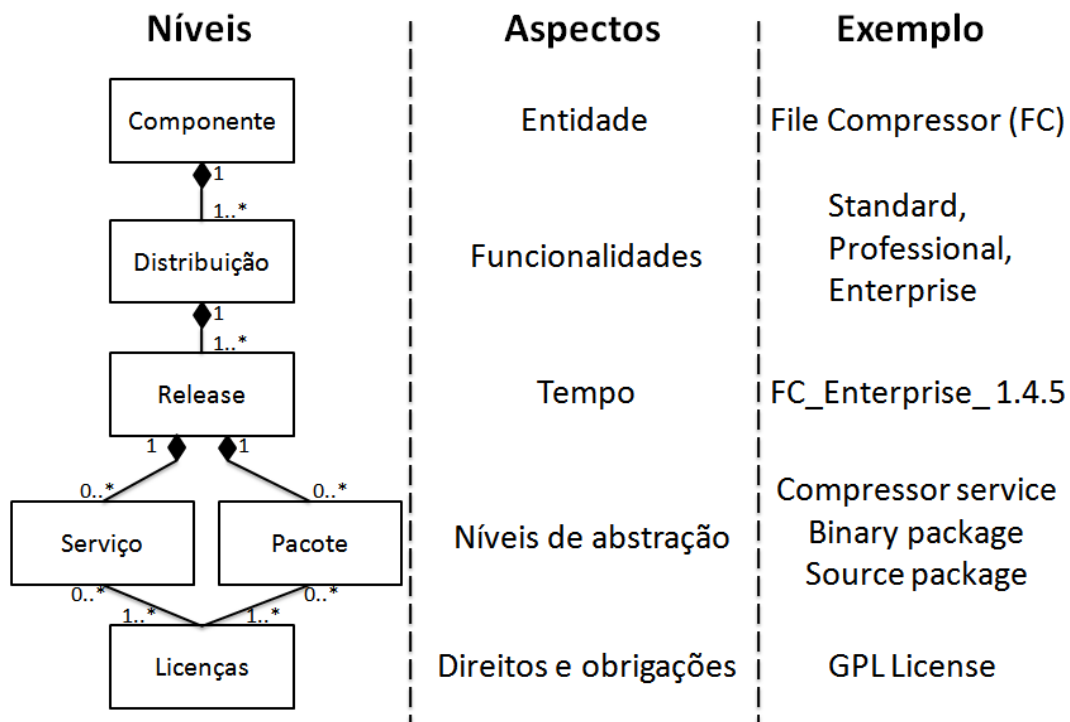


Figura 2.6 – Organização interna da biblioteca *Brechó*. Fonte: (MARINHO *et al.*, 2009b)

Além disso, a biblioteca *Brechó* apresenta as seguintes funcionalidades (WERNER *et al.*, 2007; WERNER *et al.*, 2009; SANTOS *et al.*, 2010a):

- mecanismos de dependência física (entre *licenças*) e de dependência lógica (entre *releases*) de componentes, tratando dependências diretas e indiretas (transitivas);
- mecanismos de controle de publicação de componentes e serviços pelo gerente da biblioteca;

- mecanismos para facilitar a publicação de componentes e de serviços, incluindo os conceitos de entidades *default* e de empacotamento automático de artefatos;
- mecanismos de controle de versão de componentes e de serviços;
- mecanismo de busca pela combinação das técnicas de palavras-chave, hierarquia de termos, filtragem dos componentes por elementos de documentação e correção sintática (sugestão);
- mecanismo de avaliação de componentes e de visualização de avaliações realizadas previamente;
- mecanismo de categorização, que permite ao produtor sugerir categorias e, ao administrador, reorganizar as categorias com apoio automatizado;
- desenvolvimento de uma API de serviços de acesso às funcionalidades da biblioteca para sua integração com outras ferramentas, a fim de facilitar o processo de publicação direta de componentes sem precisar acessar a biblioteca Brechó;
- mecanismo de tarifação, para coordenar diferentes formas de aquisição (ou uso) de artefatos (ou serviços) de componentes publicados;
- evolução para Brechó-VCM, com mecanismos para prover um canal de distribuição para um mercado de componentes baseado em valor, que agregam elementos de negócio (SANTOS, 2010):
 - *precificação*: incorporação do elemento *crédito* (moeda virtual) através de modelos econômicos para apoiar a atividade de publicação e recuperação de componentes partir da proposição e percepção de valor, seja com base em modelos matemáticos, ou analisando componentes similares e fornecendo estatísticas de mercado, e.g., valor médio de similares, número médio de consumidores e avaliação média daqueles adquiridos;
 - *marketing*: incorporação do elemento *ponto* (recompensa virtual) através de algoritmos de inteligência coletiva para premiar componentes mais vendidos e bem avaliados e para promover aqueles menos vendidos e mal avaliados, assim como recomendar componentes aos envolvidos no mercado e prover estatísticas de

mercado, e.g., categorias que mais vendem e que mais têm publicações de novos pacotes e serviços;

- *avaliação*: incorporação do conceito de *credibilidade* (grau de confiança) através de modelos de avaliação aplicados a componentes, para explorar a qualidade percebida no uso, e a produtores e consumidores, para explorar aspectos de reputação e experiência a fim de ponderar suas ações no repositório, e.g., o peso da similaridade construída na precificação de um artefato, quando o seu produtor avalia artefatos similares automaticamente recuperados do mercado e atesta aqueles que realmente o são;
- *negociação*: incorporação do conceito de *perfil* (reputação de mercado) através de um modelo de negociação baseado em ciclos de proposição de valor por meio de diálogos entre os envolvidos em uma transação, cujas ações contribuem para a construção de perfis de negociação para cada um deles, considerando seus diferentes papéis, i.e., produtor ou consumidor, compostos por dados sumarizados, e.g., histórico de resultados e avaliações de negociações conduzidas anteriormente;
- *visualização*: incorporação do conceito de *insight* (exposição de informação) através de um conjunto de dados históricos que permitem os mecanismos anteriores extrair, manipular e apresentar informações do negócio de componentes, a fim de que os envolvidos no mercado possam melhorar a proposição e percepção de valor (não apenas relativas a custo/lucro), e.g., informação “recomendação de componentes” derivada dos dados “históricos de compras”, “similaridade histórica”, “graus de satisfação” e “categorias de interesse”, estimulando a percepção das facetas de valor *benefícios*, *oportunidades* e *necessidades*.

2.6 Considerações Finais

Ao analisar o contexto mais amplo da construção de um produto de software, identificam-se duas abordagens distintas (SOFTEX, 2007): *sob encomenda*, em que se desenvolve o produto de software “do zero”, utilizando as ferramentas e bibliotecas

oferecidas pela linguagem de programação adotada; e *pacote*, em que se adquire um produto de software pronto, configurável para atender a uma parcela razoável dos requisitos dos usuários finais. Cada uma delas apresenta vantagens e desvantagens visíveis e é neste sentido que as técnicas de ED e de DBC visam interferir, a fim de alcançar um equilíbrio ao incorporar a reutilização no processo de desenvolvimento. Apesar de haver etapas comuns entre a ES tradicional e baseada em reutilização, a adoção dessas técnicas faz com que todo o sistema seja construído de maneira diferente, pois os envolvidos devem ter em mente que o sistema é desenvolvido pela utilização de componentes. Deve-se considerar disponibilidade, maximização do grau de reutilização e desenvolvimento de novos componentes, o que agrega ao projeto uma complexidade adicional, decorrente da preocupação econômica e técnica diante do tratamento da reusabilidade (CAI *et al.*, 2000). O repositório pode coletar e analisar dados referentes à avaliação contínua de procedimentos e ferramentas, apoiando na definição de valor para componentes. Neste caso, o repositório de referência se torna um elemento chave por prover um catálogo de componentes de diversos repositórios espalhados pela Internet, e deveria possuir propriedades funcionais e configuráveis para apoiar mecanismos de armazenamento, publicação, documentação, busca e recuperação. Além disso, este repositório deveria considerar cenários colaborativos e competitivos, incluindo a utilidade da coleção e da precisão da classificação de componentes, e a responsabilidade do repositório com relação a necessidades, percepções de valor e papéis dos envolvidos (SANTOS & WERNER, 2009).

Este capítulo apresentou um levantamento sobre conceitos de Reutilização de Software. Entre suas técnicas, foram discutidos ED e DBC, além da biblioteca Brechó. Vários conceitos foram apresentados com a finalidade de servir ao objetivo desta proposta de tese, cujo enfoque está em viabilizar a compreensão de ECOSs a partir da definição, modelagem e análise do seu domínio. Isso se deve às carências da literatura nessa direção, uma vez que ainda têm caráter exploratório no tema, i.e., os elementos que compõem os ECOSs ainda são discutidos de maneira divergente. Por exemplo, objetos e relacionamentos de um ECOSs têm papéis e responsabilidades que podem variar segundo a seleção de um caso, considerando um dado contexto onde ele está inserido, o que sugere uma possibilidade de utilização de ED para apoiar na sua compreensão. Alguns elementos de ECOSs são materializados em software, como os chamados ativos da plataforma do ECOSs, i.e., aplicações, componentes e serviços (FAVARO & PFLEEGER, 2011), e precisam ser geridos para favorecer a tomada de

decisão na indústria de software. Por exemplo, para organizações consumidoras, questões como “que tecnologias adotar?” e “que parceiros selecionar?” emergem em um mercado de software autorregulado pela comunidade ao seu redor e em modelos de negócio dinâmicos, paralelamente às questões técnicas. Nessa direção, os Capítulos 3 e 4 apresentam, respectivamente, uma visão geral do tema e a abordagem de solução para esta proposta de tese.

Capítulo 3 – A Evolução da Indústria Orientada aos Ecossistemas de Software

3.1 Introdução

Quando o software corresponde a um empreendimento de sucesso, i.e., (i) quando encontra as necessidades das pessoas que o utilizam, (ii) quando funciona perfeitamente por um longo período de tempo, e (iii) quando é fácil de manter, ou mesmo de usar, pode transformar e melhorar um dado cenário (MESSERSCHIMITT & SZYPERSKI, 2003). Por outro lado, quando o software falha, i.e., (i) quando os usuários estão insatisfeitos, (ii) quando a propensão a falhas é alta, e (iii) quando é difícil de modificar, ou mesmo de usar, pode prejudicar um dado cenário. Portanto, os engenheiros de software querem construir software de sucesso e evitar as consequências das falhas. Tradicionalmente na Engenharia, o sucesso está na disciplina de projeto e construção de produtos e serviços e, para o caso do software, essa busca tem se repetido desde a famosa “Crise do Software”¹¹ na década de 1960 (PRESSMAN, 2010).

Ainda hoje, a construção e a utilização de produtos de software permanecem entre as questões mais complexas e problemáticas do desenvolvimento da tecnologia moderna, cujo principal sintoma se relaciona ao crescente índice de falhas em grandes projetos, programas e negócios (BOEHM, 2006). O valor agregado deve ser analisado como base em um conjunto de dimensões que expõe empreendimentos de software a riscos imprevisíveis e uma taxa inaceitável (ITABORAHY, 2007). Mesmo cancelados, os projetos consomem recursos antes de serem assim classificados. Outros problemas também devem ser considerados, como: (i) atrasos no *time-to-market*; (ii) redução de qualidade de produtos de software; e (iii) perda de oportunidades. BOEHM & SULLIVAN (2000) afirmam que estes problemas se devem à falta de habilidades para gerenciar efetivamente as características de risco em sistemas de software, decorrentes da natureza da tomada de decisão em circunstâncias de incerteza, sobretudo durante o processo de desenvolvimento globalizado.

Dessa forma, conhecimento incompleto e mercado competitivo se mantêm como desafios que demandam estratégias para melhor investimento de recursos em busca da

¹¹Termo cunhado no final da década de 1960, devido às frustrações e apreensões que envolviam o desenvolvimento e a manutenção de produtos de software.

geração de valor (SANTOS, 2010). Isso propiciou um tópico que remonta à década de 1980, a *Economia Aplicada à Engenharia de Software* (ES), cuja meta está em gerar pesquisa e desenvolvimento com foco na interseção entre a economia aplicada à informação e a engenharia e projeto de software (BOEHM, 1981). Apesar dos avanços neste tópico, percebia-se que o conceito de valor em sentido amplo era desconsiderado, especialmente em se tratando de um produto/serviço como o software, i.e., o papel dual de ser simultaneamente um meio para entregar um produto e sê-lo *per si* (BOEHM, 2006). Quando as considerações de valor permanecem implícitas, o efeito global pode ser muito negativo (BIFFL *et al.*, 2006), incluindo conflitos entre as proposições dos envolvidos. Entre estes envolvidos, estão desde profissionais de *marketing* e analistas de negócio a desenvolvedores, arquitetos, especialistas em qualidade, em processo e em medição, gerentes de projeto e executivos. Quando as diferentes perspectivas de valor não são explicadas e reconciliadas, todos os envolvidos saem perdendo no final.

Ao final da década de 1990, surge o tópico *Engenharia de Software Baseada em Valor* (ESBV), que contempla a noção de valor¹² como ponto de partida para se referir ao desafio da ES de tratar as questões sociais e econômicas e os processos de engenharia de sistemas. A meta está em entender o valor agregado a qualquer artefato que esteja sendo gerado ou utilizado pelos envolvidos ao longo do processo de desenvolvimento e nos mercados subjacentes. No contexto da ESBV, o conceito de valor não se limita a termos econômicos ou monetários, mas consiste um conjunto de facetas quantificáveis e/ou qualificáveis que se transformam e transferem através das proposições e das percepções dos diferentes envolvidos em uma cadeia de valor¹³ (SANTOS, 2010). Isso se aplica aos benefícios que um empreendimento ou decisão deve produzir, à sua percepção para um indivíduo/grupo e à sua utilidade/importância.

Após três décadas de esforços em direção ao entendimento do conceito de valor na ES, confirmou-se a interferência de questões que estão além da alçada puramente técnica, como fatores-chave para o sucesso de projetos, processos e produtos de software (BOEHM, 2006). Pouco a pouco, esta percepção vem abrindo espaço para novas perspectivas na ES a fim de enxergar o software como um dos diversos elementos de um *sistema* (CUKIERMAN *et al.*, 2007; BOSCH, 2009; FRANÇA & SILVA, 2009;

¹²Entre as facetas de valor, estão custos, lucros, benefícios, riscos, tempo, oportunidades, necessidades, flexibilidades, comportamento e requisitos. Mais detalhes, ver (SANTOS, 2010).

¹³Cadeia de valor consiste em uma estrutura de circulação de valor (e de suas facetas) que integra os envolvidos a ações, atividades, papéis e decisões no que tange o empreendimento em questão, seja um componente, um projeto, um produto, um processo, a organização ou o mercado (SANTOS *et al.*, 2010a).

JANSEN *et al.*, 2009b; MEIRA & SILVA, 2009; SILVA & CÉSAR, 2009; SANTOS & WERNER, 2010a). A formação de redes socio-técnicas entre os diversos envolvidos no processo de desenvolvimento e os diferentes tipos de artefatos produzidos trouxe para a indústria de software a necessidade de pensar a abertura das fronteiras organizacionais. Dessa forma, atores externos, conhecidos ou não, são agregados e compõem uma nova maneira de estruturar modelos de negócio (SANTOS *et al.*, 2013a). Estas redes são denominadas Ecossistemas de Software (ECOSs), um conceito que tem se tornado importante para explicar a existência de novos elementos atrelados ao software (MESSERSCHMITT & SZYPERSKI, 2003).

Os ECOSs realçam uma nova questão da indústria de software: *inovação acontece em qualquer lugar, mas simplesmente existe um outro lugar além “daqui”, i.e., além da organização* (GOLDMAN & GABRIEL, 2005). Ou seja, por mais esperta, criativa e inovadora que uma organização de software possa ser, existem mais pessoas mais espertas, criativas e inovadores *fora* desta organização, do que dentro dela. Por exemplo, mesmo os clientes utilizando um software em seu dia a dia estão em ótimas posições para oferecer sugestões ou criar/evoluir este software. Este capítulo apresenta uma evolução da visão da ES a fim de contemplar os ECOSs, do ponto de vista da Reutilização de Software. Isso serve de base para discutir os ECOSs e seus desafios para os próximos anos. O capítulo está organizado da seguinte forma: na Seção 3.2, são estruturadas as “4 gerações” da Reutilização de Software, i.e., sistemas monolíticos, sistemas baseados em componentes, Linhas de Produtos de Software (LPSs) e ECOSs; na Seção 3.3, discute-se o estado da arte da pesquisa em ECOSs; a Seção 3.4 discorre sobre os principais desafios deste tópico de pesquisa agrupados por períodos de discussão na comunidade, incluindo o cenário brasileiro; e a Seção 3.5 conclui o capítulo com algumas considerações finais.

3.2 Da Engenharia de Software aos Ecossistemas: As “4 Gerações” da Reutilização

O termo *software* pode ser definido como conjunto de instruções que proveem características, funções e desempenho desejados, quando executados corretamente, além de estruturas de dados que permitem aos programas manipularem informações e o conhecimento que descreve a operação e uso dos programas em formato impresso e virtual (IEEE, 2004; MESSERSCHMITT & SZYPERSKI, 2003). No entanto, software é um elemento de um sistema lógico, mais do que físico, o que o diferencia

significativamente do hardware (PRESSMAN, 2010): (i) *software é desenvolvido ou “engenheirado”*, i.e., não é manufaturado, uma vez que envolve pessoas e requer boa gerência de projetos para tratar seus custos; (ii) *software não se desgasta*, i.e., não é suscetível a fatores ambientais como acontece com o hardware, mas se deteriora; e (iii) embora a indústria de software se mova em direção a DBC, *a maior parte dos sistemas continua sendo construída de maneira customizada*, i.e., apesar da indústria de hardware ter se apropriado do conceito de reutilização da Engenharia, isso não aconteceu na mesma proporção para software, pois as suas propriedades não favoreceram uma corrente de pensamento baseada puramente em interfaces e em bibliotecas de componentes.

Desse contexto, pode-se refletir a trajetória da ES, que historicamente remonta a conferência do comitê de ciência da Organização do Tratado do Atlântico Norte em 1968 (NAUR & RANDELL, 1969). De acordo com BROOKS (1995), apenas o artigo apresentado por Doug McIlroy tinha relação direta com a Engenharia, de modo que a complexidade econômica e social envolvida no desenvolvimento de software e a natureza única deste processo sempre apareceram na comunidade da Computação, mesmo que inconscientemente. A contribuição de McIlroy se referia à ideia de decompor sistemas de software em unidades reutilizáveis, chamadas “componentes”, e ao uso de biblioteca de programas utilitários para realizar tarefas conhecidas, como classificação de itens ou rotinas matemáticas (BARROCA *et al.*, 2005). No final da década de 1980, essa vertente da ES emergiu como uma área de pesquisa denominada Reutilização de Software, i.e., o uso de qualquer informação já disponível que um desenvolvedor pode necessitar no processo de criação de software (FREEMAN, 1987). Entretanto, a noção de reutilização é antiga e originária da busca por soluções consistentes para problemas, que pudessem ser aplicadas a novos problemas e cuja repetitividade as tornaria aceitas, generalizadas e padronizadas, como acontece na Matemática e na Física (WERNER, 1992).

Apesar dos maiores benefícios da reutilização corresponderem ao aumento da produtividade e à redução do esforço de desenvolvimento, seus maiores obstáculos são de natureza gerencial, organizacional, econômica e técnica, conforme BROOKS (1995) já ressaltava. Dessa forma, BARROCA *et al.* (2005) afirmam que o desenvolvimento de software baseado em reutilização deve ser visto como um ciclo contínuo de aprendizagem influenciada por atividades relacionadas à Engenharia de Domínio (ED). Isso despertou na indústria de software uma reação do ponto de vista prático, ao final a

década de 1990, culminando com a extensão do conceito de ED com ênfase no mercado e originando as LPSs (NORTHROP, 2004). Não obstante ter se transformado em uma das abordagens mais bem sucedidas para a reutilização intraorganizacional, segundo BOSCH (2009), a arquitetura e os componentes compartilhados em uma LPS, reunidos sob a denominação de *plataforma*, começaram a sofrer com as pressões de abertura para agregar novos envolvidos, externos à organização, como destaca WERNER (2009).

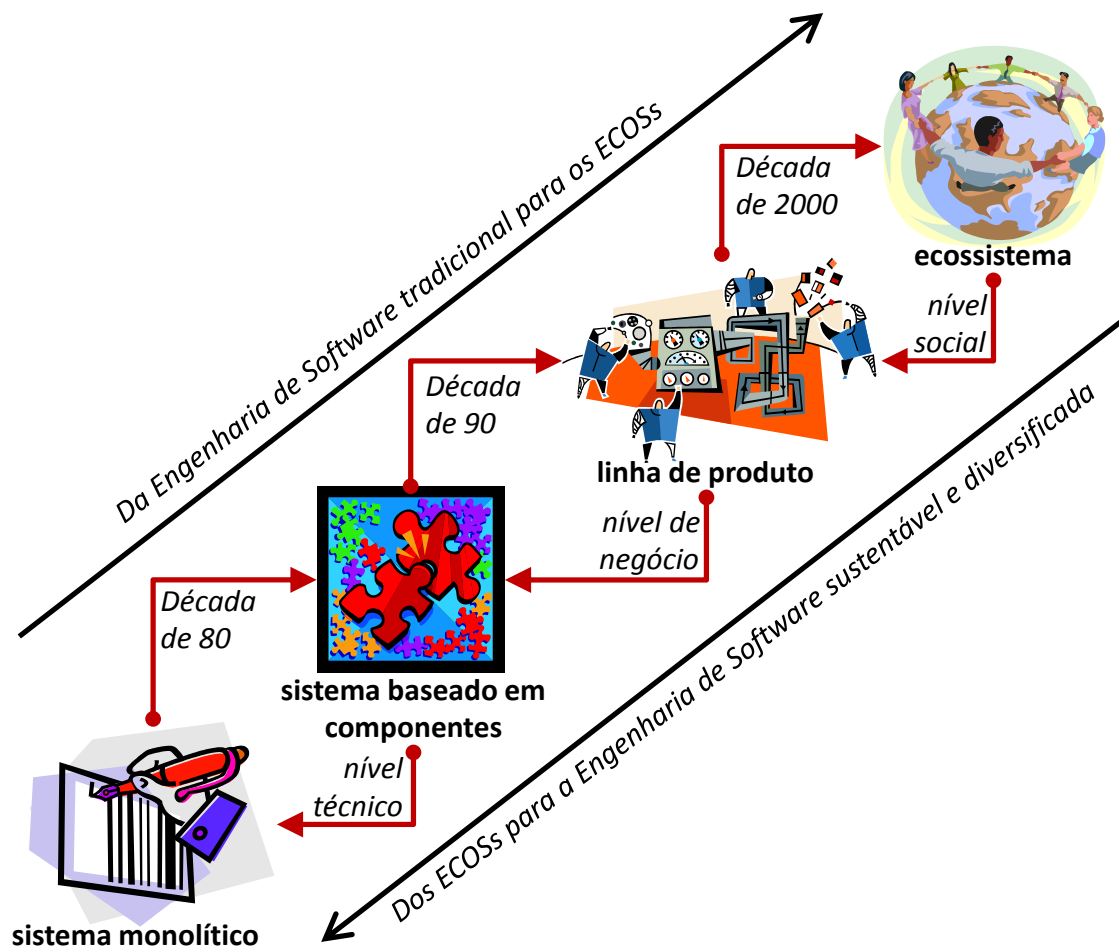


Figura 3.1 – Trajetória da Reutilização de Software em “4 gerações”

Como consequência das novas tendências de negócio de inovação aberta (CHESBROUGH, 2003), esse movimento propiciou a transição da visão tradicional da ES para os ECOSs, a fim de investigar as questões sociais e econômicas intrínsecas ao ciclo de vida do software baseado em reutilização, para a década de 2010 (SANTOS & WERNER, 2010b). Partindo desse discurso, pode-se definir uma trajetória da ES para os ECOSs sob a perspectiva da Reutilização de Software, ilustrada pelas “4 gerações” apresentadas na Figura 3.1. Observa-se que existem duas direções: (1) *Da ES tradicional para os ECOSs*, que considera a evolução histórica e o amadurecimento da

ES nas últimas três décadas, visando maximizar os benefícios preconizados e tratar os obstáculos inerentes; e (2) *Dos ECOSs para a ES sustentável e diversificada*, que considera os diferentes níveis (ou dimensões), igualmente importantes e cumulativos, sobre os quais a ES se constitui, visando a sua sustentabilidade¹⁴ e diversidade¹⁵. Nas subseções a seguir, discute-se cada uma das “4 gerações”.

3.2.1. Primeira Geração: Sistemas Monolíticos

A principal reação contra a abordagem “codifica-e-remenda” da década de 1960 envolveu processos nos quais a codificação era mais cuidadosamente organizada, precedida pela modelagem e esta pela engenharia de requisitos. Consequentemente, enfatizavam que programas sequenciais poderiam ser construídos com a programação estruturada, fosse esta baseada em métodos formais, ou em cálculo de programação. Entre meados das décadas de 1970 e 1980, princípios de modularidade foram explorados e fortalecidos, e.g., coesão e acoplamento, além de técnicas de encapsulamento e de tipos abstratos de dados (PRESSMAN, 2010), o que contribuiu para a construção dos *sistemas monolíticos*, i.e., sistemas normalmente compostos por conjuntos de rotinas que têm permissão para interagir livremente umas com as outras. Melhorias foram buscadas em direção à produtividade e à escalabilidade. Por exemplo, organizações que despendiam 60% do seu esforço na fase de teste descobriram que 70% destas atividades na verdade representavam retrabalho, que poderia ser menos custoso caso evitado ao estruturar melhor as fases anteriores do ciclo de vida (BOEHM, 2006).

Além disso, as abordagens com potencial de produtividade na década de 1980, tais como sistemas especialistas, linguagens de alto nível, orientação a objetos, estações de trabalho poderosas e programação visual, propiciaram a meta de racionalizar e evitar trabalho por meio da emergência da área de reutilização. Sistemas operacionais mais poderosos, sistemas de gerenciamento de banco de dados, construtores de interface gráfica, *middleware* distribuído e automação de escritório instigaram os envolvidos no processo de desenvolvimento sobre um avanço maior do que apenas a reutilização do tipo “copia-e-cola” (BOEHM, 2006). As linguagens de programação orientadas a objeto impulsionaram a prática de modularização na programação estruturada, cujo paradigma é norteado pela ideia de procedimentos e funções, visando um paradigma orientado à

¹⁴Sustentabilidade é a capacidade do ECOS sobreviver às diferentes perdas (DHUNGANA *et al.*, 2010).

¹⁵Diversidade é a capacidade de prover diversas oportunidades no ECOS (DHUNGANA *et al.*, 2010).

modelagem de classes de objetos, descritos por características e comportamentos baseados no mundo real. De certa forma, esse movimento reaproximou as bases da ES daquelas da Engenharia de Hardware da década de 1960, uma vez que o estilo de pensamento da área de reutilização¹⁶ recorria ao ideal de montagem das máquinas.

3.2.2. Segunda Geração: Sistemas Baseados em Componentes

Segundo BIFFL *et al.* (2006), com o avanço na área de reutilização no final dos anos 1980, técnicas como a ED e arquitetura de software viabilizaram a reutilização efetiva de componentes de aplicação através de *frameworks* e de linguagens de quarta geração específicas de domínio. Os engenheiros de software passaram a observar a experiência de outros setores da indústria na utilização de *sistemas baseados em componentes*, pois a indústria de software indicava que muitos requisitos não eram novos e já teriam sido desenvolvidos em um contexto um pouco diferente (BARROCA *et al.*, 2005). Métodos para orientação a objetos favoreceram a reutilização e a evolução de software por meio de estruturas e relacionamentos, e.g., classes, objetos, métodos e interfaces, para as aplicações de um domínio, de forma mais natural. BOEHM (2006) justifica esta tendência à reutilização pelo fato de que seus conceitos fornecem melhor apoio à modularização dos tipos abstratos de dados para módulos de alta coesão e de baixo acoplamento, sobretudo quando a produtividade da manutenção de software consumia de 50 a 75% dos esforços da maioria das organizações (BOEHM, 1981). Em paralelo, os ambientes de desenvolvimento de software estimularam um rápido crescimento do desenvolvimento orientado a objetos, com métodos para modelagem e implementação que convergiam pelo nascimento e estabelecimento da *Unified Modelling Language* (UML), na década de 1990, construindo o **foco técnico**.

Na mesma época, a comunidade de ES presenciava a expansão da Internet e da *World Wide Web* (WWW), o que contribuiu para fortalecer os métodos orientados a objetos e o caráter do software como elemento em um mercado competitivo (BOEHM,

¹⁶Estilo de pensamento corresponde a uma coerção determinada de pensamento, a totalidade da preparação e disponibilidade intelectual orientada a ver e atuar de uma forma e não de outra, de modo que a dependência de qualquer fato científico (ou artefato) seja evidente (FLECK, 1986). No caso da Reutilização de Software, os “desbravadores” da ES (engenheiros e matemáticos), seguindo o estilo de pensamento dominante que fundamentou a sua construção (i.e., busca por padrões e modelos universais, oriunda das outras engenharias) agiram vários anos como difusores de princípios e práticas de suas respectivas formações durante a sua construção inicial. Existe uma forte característica de incerteza nesse processo, pois, a priori, não se pode garantir que os elementos alistados agirão como esperado ou mesmo se permanecerão enredados, de modo que as peculiaridades da ES, que estão além do cunho puramente técnico, devem ser consideradas (SANTOS, 2010).

2006). O software passou a ser tratado como elemento de diferenciação no mundo dos negócios, provocando um movimento de processos de ES para contemplarem atividades concorrentes e, conseqüentemente, orientadas pela reutilização. Por exemplo, o caso da Hewlett Packard (HP) foi relatado por POULIN (1996) e mostrava que os investimentos em arquitetura de software e componentes reutilizáveis aumentou o tempo de desenvolvimento dos três primeiros produtos (1986-1987), mas reduziu o tempo, ano a ano, até 1992, quando finalizou o estudo de caso. De acordo com BOEHM (2006), duas direções de pesquisa ganhavam destaque: desenvolvimento *open source* e usabilidade (interface homem-máquina). Somados, todos estes fatores levaram a indústria de software a uma ampla aplicação de ED voltada para o negócio da organização.

3.2.3. Terceira Geração: Sistemas Baseados em Linhas de Produtos

Com as pesquisas da ES para tratar o desenvolvimento de infraestruturas *Web* e, no começo dos anos 2000, *Mobile*, novos fatores ganham destaque na indústria de software: software crítico, gerência de dependências, serviços *Web*, manutenção de sistemas legados, componentes de prateleira ou *comercial-off-the-shelf* (COTS) e *free and open source software* (FOSS), conforme SANTOS (2008). As metodologias ágeis ganham espaço com o Manifesto Ágil (BECK *et al.*, 2001), que enfatizam participação “co-localizada”¹⁷ do cliente, desenvolvimento de pequenos incrementos, modelagem simples, programação aos pares, refatoração e integração contínua. Mesmo que a aplicação dessas metodologias tenha funcionado mais em projetos de pequeno e médio porte, suas características ainda refletiam um momento de transição para a ES baseada em reutilização (BOEHM, 2006). Para as organizações, estava claro o esforço para atingir os benefícios de reutilização, sobretudo ao agregar **o foco no negócio**, o que levou ao conceito de LPS no final dos anos 1990 (NORTHROP, 2004). O objetivo da indústria era utilizar a ED com ênfase no mercado, inspirada pelo sucesso do conceito de linha de produção em organizações como a Toshiba (BOEHM, 2006).

Nas LPSs, os artefatos técnicos e gerenciais da base de ativos reutilizáveis da organização devem capitalizá-la, considerando as similaridades entre os produtos para apoiar a sua variabilidade (CLEMENTS & NORTHROP, 2001). No entanto, além do custo da construção dessa base, o desenvolvimento da infraestrutura de uma LPS também requer investimento, por envolver novas práticas, processos e ferramentas.

¹⁷ Presença constante do cliente no ambiente de trabalho da equipe de projeto de software.

NORTHROP (2004) destaca outros impedimentos para a adoção da abordagem de LPS: processos de desenvolvimento incompatíveis; carência de conhecimento e de talento necessários; resistência cultural; falta de apoio gerencial; práticas de aquisição incompatíveis; falta de práticas de gerência e de desenvolvimento disciplinadas; falta de visão de linha de produção; falta de adoção de um plano para LPS; falta de um “campeão”¹⁸ da LPS; e expectativas irreais. As organizações que arcaram com os custos de adoção de uma LPS conseguiram sucesso com a reutilização intraorganizacional (BOSCH, 2009), ao passo que as demais mantiveram a reutilização *ad hoc* ou baseada em componentes, pela complexidade. Independentemente do caso, MCGREGOR (2012) afirma que o processo de tomada de decisão na ES estratégica passa por uma transformação que impacta a adoção de LPS: a inovação “força” a abertura das fronteiras organizacionais, de modo que modelos de negócio, processos, produtos, serviços, recursos, informações e artefatos sejam alvos de uma comunidade em torno da plataforma da organização, podendo gerar vantagem competitiva.

3.2.4. Quarta Geração: Sistemas Baseados em Ecossistemas

Na década de 2000, várias organizações melhoraram a eficiência de setores de Pesquisa e Desenvolvimento (P&D), ampliaram os portfólios de produtos e passaram a oferecer níveis de configuração de seus produtos que poderiam ser alcançados por meio de componentes compartilhados com variabilidade de software relevante (BOSCH, 2010). Com a adoção de LPS dentro de uma organização e diante da evolução de seu escopo, as suas fronteiras começam a se expandir naturalmente, ou pelo menos serem reavaliadas. Mesmo para organizações de menor porte, que desejam explorar LPSs mas esbarram em seus custos subjacentes, estratégias de inovação aberta passaram a ser fundamentais para a sua sobrevivência (GOLDMAN & GABRIEL, 2005). Isso significa que a plataforma pode estar disponível para partes externas à organização e, uma vez que ela decide abrir a sua fronteira, pode ocorrer a transição de LPS para ECOS, onde o **foco social** é agregado frente à gerência da rede de envolvidos no processo de desenvolvimento, conhecidos ou não (BOSCH, 2009). Para obter sucesso, as organizações precisam descobrir maneiras de utilizar as inovações externas para torná-las parte de uma fábrica de inovação distribuída, pela combinação de licenças e

¹⁸Aquele que conhece perfeitamente o domínio de aplicação, que tem responsabilidade, autoridade e experiência suficientes dentro da organização e que tem motivação e entendimento das teorias de LPS (CLEMENTS & NORTHROP, 2001).

recompensas bem definidas, o que requer compreender e adaptar modelos de negócio, além de modelar e analisar as redes socio-técnicas (CAMPBELL & AHMED, 2010).

De acordo com BOSCH (2009), existem duas razões para uma organização se mover em direção à abordagem de ECOS: (i) ela pode perceber que a quantidade de funcionalidades que ela precisa desenvolver para satisfazer as necessidades dos clientes é muito maior do que ela pode construir, considerando o investimento em P&D e o tempo disponível para um retorno adequado; e (ii) a tendência à customização em massa direciona a demanda por investimento em P&D para aplicações de software de sucesso, de modo que estender o produto (e, por sua vez, a plataforma) com o apoio de atores externos à organização parece ser um mecanismo efetivo. Outros argumentos são apontados: aumentar o valor da oferta do núcleo da LPS para usuários existentes; gerar atratividade em novos usuários; aumentar a rigidez da plataforma, i.e., tornar mais difícil a mudança de plataforma; acelerar a produção através da inovação aberta, possível no ECOS, i.e., colaborar com parceiros para compartilhar custos da inovação do processo de ES, bem como incorporar a funcionalidade desenvolvida no ECOS, caso seja atestado o seu sucesso; e reduzir o custo total despendido pelo proprietário da plataforma do ECOS por “comoditizar” funcionalidade ao compartilhar a sua manutenção com parceiros no ECOS.

Vale destacar que a abordagem de ECOSs para aprimorar a ES baseada em reutilização ainda está em construção. O primeiro registro do uso deste termo na área remonta ao ano de 2003 (BARBOSA & ALVES, 2011). Apesar disso, as motivações de dez anos atrás continuam as mesmas de hoje (MESSERSCHMITT & SZYPERSKI, 2003; JANSEN *et al.*, 2013): o software é diferente, é ubíquo, faz o ambiente interativo, é importante, é sobre pessoas, pode ser melhor, faz parte de uma indústria em constante mudança radical, é uma criação social, é sofisticado e complexo, e pode ser controlado. A partir do conhecimento organizado ao longo da Seção 3.2, os ECOSs vêm sendo analisados em três níveis ou dimensões, sobrepostos e inter-relacionados, emergentes na ES ao longo dos últimos 40 anos: *técnica*, *transacional* ou *de negócio*, e *social* (CAMPBELL & AHMED, 2010; SANTOS & WERNER, 2011b; MANIKAS & HANSEN 2013; BARBOSA *et al.*, 2013). A Seção 3.3 e a Seção 3.4 apresentam uma visão geral da pesquisa em ECOSs, bem como os desafios para o estabelecimento da “quarta geração” da Reutilização de Software, ao passo que o Capítulo 4 explora as três dimensões de ECOSs e apresenta a quarta dimensão na abordagem proposta.

3.3 Visão Geral de Ecossistemas de Software

BOSCH (2009) explica que a noção de ecossistemas se originou da Ecologia. Um *ecossistema* é uma comunidade de organismos vivos (i.e., plantas, animais e microrganismos) em conjunto com componentes não vivos (e.g., ar, água e solo mineral), e suas relações entre si e com o ambiente, interagindo como um sistema (SMITH & SMITH, 2008). Estes componentes bióticos e abióticos estão relacionados por meio de ligações em ciclos de nutrientes ou matéria e em fluxos de energia em um ambiente que pode ter qualquer tamanho, embora normalmente abranja espaços específicos e limitados – apesar de alguns cientistas defenderem a tese de que o planeta Terra é um ecossistema como um todo (KREBS, 2008). Embora esta seja uma definição aceitável, ela se distancia da discussão apresentada acerca de ECOSs, de modo que a noção de ecossistemas humanos pode ser mais apropriada. BOSCH (2009) define um *ecossistema humano* como um conjunto de atores e suas conexões e atividades, bem como as transações realizadas a partir dessas conexões, considerando fatores físicos e não físicos. Pode-se distingui-lo entre ecossistemas comerciais e sociais. Em um *ecossistema comercial*, os atores são os negócios, fornecedores e clientes; os fatores são os bens e serviços; e as transações incluem não somente as financeiras, mas também compartilhamento de conhecimento e de informação, investigação, contatos pré e pós-vendas etc. (MCGREGOR, 2010). Por sua vez, os *ecossistemas sociais* consistem em um conjunto de usuários e suas conexões sociais, juntamente com as diferentes trocas de informação (e.g., conhecimento, soluções, ideias, referências, *feedback* etc.).

Na ES, BARBOSA & ALVES (2011) afirmam que a expressão *ecossistema de software* foi inspirada pelos conceitos de ecossistemas dito biológicos e de ecossistemas comerciais ou de negócios. Para BERK *et al.* (2010), um ECOS pode ser entendido como um subtipo de ecossistema de negócio (*business ecosystem*) que, de acordo com MOORE (1996), representa uma analogia do começo dos anos 1990 para descrever uma nova forma de observar certas redes de negócios pelas Escolas de Administração. Estes ecossistemas possuem basicamente três elementos (IANSITI & LEVIEN, 2004b): um **centralizador** (*hub*), uma **plataforma** (e.g., tecnologia ou mercado) e um conjunto de **agentes de nicho** (*niche players*). Neste contexto, o centralizador é o proprietário da plataforma e os agentes do nicho podem utilizá-la para gerar valor para ela e para si mesmos. Por exemplo, analisando o caso do sistema operacional Windows, a Microsoft é o centralizador, o Windows é a plataforma, e as outras organizações são os agentes

deste nicho, que utilizam esta plataforma para construir suas aplicações. Quanto mais as organizações seguem este exemplo e “abrem” suas estruturas de negócios para outras, mais ECOSs começam a se formar. Um problema chave está no gerenciamento da quantidade de organizações e relações na forma de um ECOS (SANTOS *et al.*, 2013a), formado em torno de **desenvolvedores** (i.e., programadores), **parceiros** (i.e., fornecedores de ferramentas e de partes do software) e **clientes** (incluindo os usuários). Isso pode dificultar *insights* relativos aos papéis que cada um possui no ECOS, bem como os impactos de suas ações e decisões no desempenho do ECOS como um todo. BERK *et al.* (2010) colocam ainda que a literatura de ecossistemas de negócio provê uma base sólida, mas não voltada especificamente para o mercado de software, o que impede a sua aplicação direta aos ECOSs.

A essência do conceito de ECOSs encoraja um conjunto de desenvolvedores externos a utilizarem uma plataforma de software pertencente a um grupo de organizações, e a contribuírem para o desenvolvimento de produtos (CAMPBELL *et al.*, 2010). Estabelece-se uma comunidade que acelera o compartilhamento de conteúdo, conhecimento, problemas, experiência e habilidades. Ao obterem sucesso com os sistemas e LPSs reunidos sobre a plataforma, as organizações experimentam pressões internas e externas para tornarem públicas as suas interfaces, impactando os principais envolvidos, como fornecedores, organizações responsáveis pelo desenvolvimento de padrões, clientes e parceiros (MCGREGOR, 2010). Isso se relaciona com o fato de que os ECOSs emergem como uma abordagem promissora para melhorar a reutilização intra e interorganizacional, no cenário da ES globalizada, conforme indicado por BOSCH & BOSCH-SIJTSEMA (2010). Busca-se promover a reutilizabilidade ao envolver desenvolvedores e demais atores externos ao processo, entender o software como um elemento de um sistema mais amplo e compreender como ele se conecta aos envolvidos no ECOS (PETTERSSON *et al.*, 2010).

BOSCH (2010) alega que as organizações querem se reposicionar para agirem como atores em rede e reduzirem a sua força de trabalho. Os agentes do nicho passam a ter um papel importante, por pertencerem direta ou indiretamente ao desenvolvimento de uma plataforma. Assim, abordagens do estilo *app-store* ganham destaque a fim de aumentar a riqueza das funcionalidades fornecidas aos clientes. Por outro lado, CATALDO & HERBSLEB (2010) discutem que o potencial dos ECOSs é diretamente proporcional à quantidade de barreiras técnicas e organizacionais, e que os benefícios preconizados advêm (e dependem) de dois princípios básicos: transparência e projeto

modular de sistema. Por *transparência*, entende-se a disponibilidade de qualquer tipo de informação relacionada ao processo de desenvolvimento, de documentos de projeto a código fonte, tarefas de desenvolvimento, defeitos e interações entre os envolvidos no ECOS; e *projeto modular de sistema* consiste na aplicação do princípio tradicional da Engenharia a partir do qual o sistema é decomposto em partes gerenciáveis, minimizando o acoplamento técnico entre os elementos constituintes.

Motivada por este cenário, a comunidade de ES começou a investir esforços na pesquisa em ECOSs, o que culminou com a criação de um fórum específico (IWSECO) e de novas conferências, como: (1) *International Conference on Software Business* (ICSOB), cuja terceira edição será realizada em 2013, juntamente com o quinto IWSECO; (2) *ACM/IFIP International Conference on Management of Emergent Digital EcoSystems* (MEDES), cuja quinta edição será realizada em 2013; e (3) *IEEE International Conference on Digital Ecosystems and Technologies* (DEST), cuja sétima edição será realizada em 2013. O registro da primeira publicação aparece em 2003, como um livro escrito por pesquisadores envolvidos com DBC e negócios em software, ainda sem uma ideia formada acerca de ECOSs, mas focada nas preocupações sociais e econômicas da ES (MESSERSCHMITT & SZYPERSKI, 2003). Apesar disso, o tópico começou a receber atenção da comunidade de ES a partir da emergência das primeiras conferências que trouxeram o conceito de ecossistemas para a Computação, como o DEST (2007), sob o rótulo de *ecossistemas digitais*¹⁹, conforme mostra a Figura 3.2.

A pesquisa em ECOSs vem crescendo, de maneira divergente, o que é natural em novos temas, e possui o desafio de construir uma base teórica comum, a fim de que os pesquisadores possam direcionar seus esforços e fundamentar experimentalmente os conhecimentos produzidos (HANSSEN & DYBA, 2012). *A priori*, esta necessidade teve como resultados dois estudos secundários: um mapeamento sistemático (BARBOSA & ALVES, 2011; BARBOSA *et al.*, 2013) e uma revisão sistemática (MANIKAS & HANSEN, 2013). Além disso, duas revisões *ad hoc* da literatura foram publicadas: um *framework* de pesquisa em ECOSs baseado em dimensões (SANTOS &

¹⁹Ecossistema digital (*digital ecosystem*) é um paradigma emergente para inovação tecnológica e econômica. Consiste de uma infraestrutura digital auto-organizável com o intuito de criar um ambiente digital para as organizações (ou agentes) conectadas em rede, provendo apoio à cooperação, compartilhamento de conhecimento e desenvolvimento de tecnologias adaptativas e abertas, além de ambientes ricos de conhecimento de domínio evolucionário. Ecossistemas digitais capturam a essência da comunidade ecológica na natureza, onde organismos biológicos formam um ecossistema complexo dinâmico e inter-relacionado, como uma analogia para organismos econômicos (e.g., entidades de negócio) ou digitais (e.g., aplicações de software). Espécies digitais, como as biológicas, “criam e conservam recursos que os humanos acham valiosos” (DEST, 2007).

WERNER, 2011b; 2011d; 2012a) e uma introdução à construção de teorias para ECOSs a partir do estado da arte (HANSSEN & DYBA, 2012). Considerando estes e outros estudos em ECOSs, as próximas subseções apresentam uma visão geral dos ECOSs: definições, classificação, características, benefícios e dificuldades, sinais de teorização, impactos na ES tradicional e grupos e áreas de pesquisa.

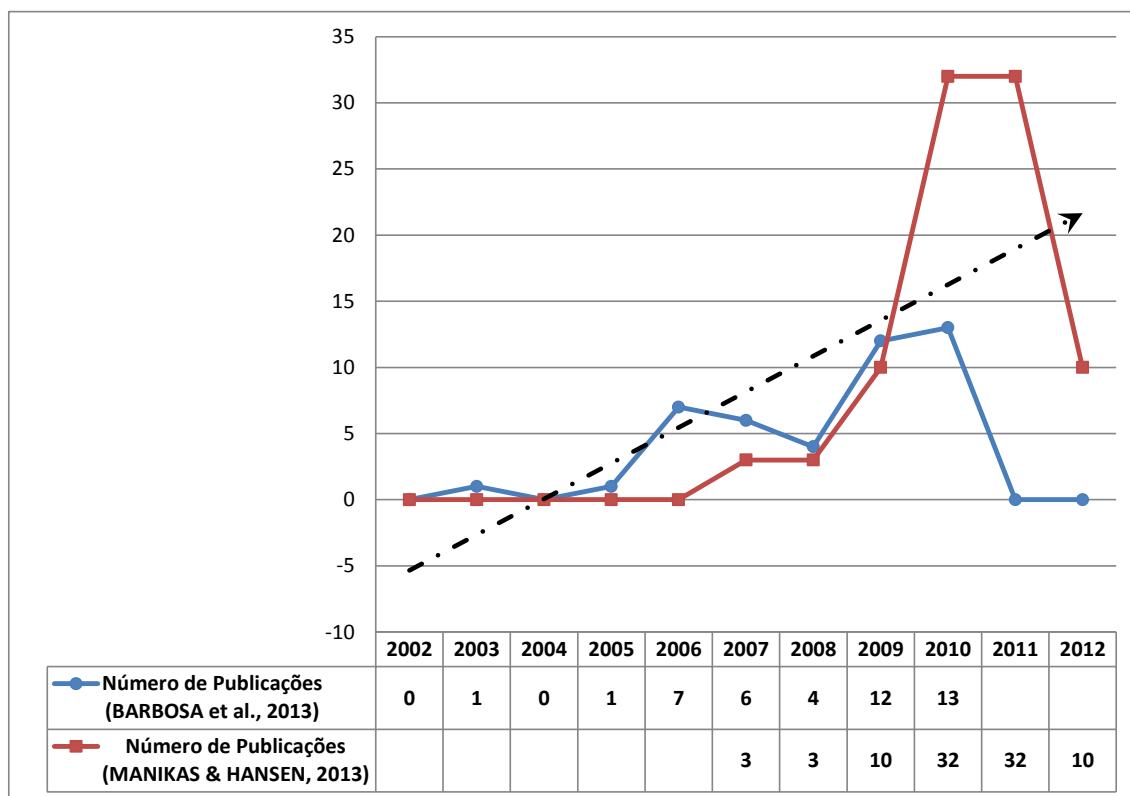


Figura 3.2 – Distribuição das publicações sobre ECOSs ao longo dos últimos anos, extraído de dois estudos secundários. Fontes: (BARBOSA *et al.*, 2013) e (MANIKAS & HANSEN, 2013)

3.3.1. Terminologia e Definição

Os ECOSs têm se tornado um campo de pesquisa ativo e definido na área de ES pelo aparecimento de novos modelos de negócio abertos que levam à redefinição dos tradicionais papéis e padrões para colaboração, inovação e proposição de valor (HANSSEN & DYBA, 2012). Isso contribui para a criação de comunidades interconectadas em complexas redes de atores e de organizações, interessadas em uma tecnologia de software central (e.g., LPS ou plataforma), que oferece oportunidades para geração de valor por diversos atores (HANSSEN, 2012). Em alguns casos, uma organização dominante governa o desenvolvimento e, em outros, o controle é difuso, como em comunidades FOSS. Em qualquer caso, são estas redes transversais às tecnologias e às organizações que se denominam ECOSs. HANSSEN & DYBA (2012)

afirmam que a abordagem de ECOSs representa um salto radical em como a ES está sendo feita, de maneira que o desenvolvimento de software está se tornando um processo aberto em um complexo ambiente distribuído. BARBOSA *et al.* (2013) resumiram a maior parte dos termos e acrônimos utilizados para se referir aos ECOSs quando existe ligação com a ES (Tabela 3.1), uma vez que diferentes grupos de pesquisa e da indústria têm investigado o tema de modo independente.

Tabela 3.1 – Termos e acrônimos utilizados para ECOSs na literatura.
Fonte: (BARBOSA *et al.*, 2013)

TERMINOLOGIA
(em ordem decrescente de importância)
<i>Software Ecosystems</i>
<i>Digital (Business) Ecosystem / D(B)E</i>
<i>SECO (Software Ecosystem)</i>
<i>Mobile Learning Ecosystems (MLES) / Mobile Ecosystem</i>
<i>FOSS Ecosystem / Open Ecosystem</i>

A partir das pesquisas realizadas por SANTOS & WERNER (2010a; 2011b), BARBOSA & ALVES (2011), HANSEN & DYBA (2012) e MANIKAS & HANSEN (2013), verifica-se a existência de diversas definições para ECOSs na literatura. As seguintes definições foram identificadas como as mais expressivas do ponto de vista da área de ES, embora a de JANSEN *et al.* (2009a) e a de BOSCH (2009) sejam as mais citadas pelos estudos publicados desde 2003 (a definição elaborada para esta proposta de tese é discutida na Seção 4.3.3):

MESSERSCHIMITT & SZYPERSKI (2003)

Um ECOS se refere tradicionalmente a uma coleção de produtos de software que têm algum determinado grau de relacionamento simbiótico – primeira definição técnica.

JANSEN *et al.* (2009a)

Um ECOS consiste em um conjunto de atores funcionando como uma unidade que interage com um mercado distribuído entre software e serviços, juntamente com as relações entre estas entidades. Estas relações são frequentemente apoiadas por uma plataforma tecnológica ou por um mercado comum e realizadas pela troca de informação, recursos e artefatos – definição mais abrangente e genérica.

BOSCH (2009)

Um ECOS consiste em um conjunto de soluções de software que possibilitam, apoiam e automatizam as atividades e transações realizadas por atores em um ecossistema social ou de negócios, e por organizações que provêm estas soluções. Assim, o foco está nos aspectos organizacionais dos ECOSs e nas relações entre as organizações enquanto entidades de software e serviços – definição técnica, mas com foco social e econômico.

KITTLAUS & CLOUGH (2009)

Um ECOS compreende uma rede informal de unidades legalmente independentes, que possuem uma influência positiva sobre o sucesso de um produto de software – são mais limitados na definição proposta.

BOSCH & BOSCH-SIJTSEMA (2010)

Um ECOS consiste em uma plataforma de software, um conjunto de desenvolvedores internos e externos e uma comunidade de especialistas no domínio a serviço de uma comunidade de usuários que compõem elementos de solução relevantes para satisfazer as suas necessidades – refinamento da definição anterior de BOSCH (2009).

LUNGU et al. (2010)

Um ECOS é uma coleção de projetos de software que são desenvolvidos e evoluem juntos em um mesmo ambiente – como a primeira definição, também é mais técnica.

HANSSEN & DYBA (2012)

ECOS consiste em uma comunidade de atores e de organizações em rede, que apoia as relações entre eles sobre um interesse comum no desenvolvimento e no uso de uma tecnologia de software central – refinamento das definições anteriores, com foco mais próximo da ES.

MANIKAS & HANSEN (2013)

Um ECOS é uma interação de um conjunto de atores sobre uma plataforma tecnológica comum, que resulta em um número de soluções ou serviços de software. Cada ator é motivado por um conjunto de interesses ou modelos de negócio e está conectado aos demais atores e ao ECOS como um todo por meio de relacionamentos simbióticos. Por sua vez, a plataforma tecnológica está estruturada para permitir o envolvimento e a contribuição dos diferentes atores – reunião das definições anteriores.

Alguns exemplos de ECOSs são o ECOS MySQL/PHP, o ECOS Eclipse, o ECOS Microsoft e o ECOS iPhone (JANSEN *et al.*, 2009a; SANTOS *et al.*, 2013a). Estes exemplos podem ser usados para estabelecer as características típicas dos ECOSs. Os ECOSs podem estar contidos em outros ECOSs, como o ECOS Microsoft CRM (*Customer Relationship Management*), contido no ECOS Microsoft. Pode-se dizer que ECOS iPhone com sua AppStore é um ECOS fechado e controlado, ao passo que o ECOS MySQL/PHP e o ECOS Eclipse são abertos, desde que as organizações tenham acesso ao código fonte e às bases de conhecimento relacionadas, permitindo maior flexibilidade no desenvolvimento. Entretanto, um ecossistema formado em uma plataforma proprietária não precisa ser fechado (ANVAARI & JANSEN, 2010), o que mostra que esses exemplos ilustram de maneira simplista o conceito de ECOS e negligenciam a ampla gama de configurações (ou variações).

3.3.2. Estruturas de Classificação

Apesar das numerosas definições apresentadas (além de pelo menos mais nove) e de pelo menos 40 publicações sem citar nenhuma delas (MANIKAS & HANSEN, 2013), **dois conceitos fundamentais** são intrínsecos aos ECOSs: (1) um interesse comum em uma *tecnologia de software central*, i.e., sistemas, serviços ou plataforma de software; e (ii) uma *rede de organizações*, i.e., algum tipo de relacionamento, seja simbiótico, ou de evolução conjunta, comercial ou técnica. Considerando os diferentes relacionamentos que a organização pode ter com tecnologia de software central, **três papéis chave** emergem nos ECOSs, conforme mostra a Tabela 3.2. Outros papéis são organizações de padronização, revendedores e operadores de sistemas (JANSEN *et al.*, 2009a). Cada um dos atores pode se beneficiar de suas participações no ECOS, e o espectro de relacionamentos simbióticos depende de seus papéis e atividades.

Cinco tipos básicos de relacionamentos podem ser identificados (MANIKAS & HANSEN, 2013): dois atores podem (i) ter benefícios mútuos (*mutualismo*); (ii) estar em competição direta (*competição/antagonismo*); (iii) não ser afetados (*neutralismo*); ou (iv) um não ser afetado enquanto o outro é beneficiado (*amensalismo*) ou prejudicado (*parasitismo*) pelo relacionamento. Dessa forma, há **três elementos principais** em ECOSs (SANTOS & WERNER, 2010a): (i) *software*, como plataforma tecnológica comum, tecnologia de software central, soluções de software, plataforma de software ou LPS; (ii) *transações*, como um senso que inclui modelos de lucro ou

recompensa, mas também possíveis benefícios além dos financeiros, e.g., algo que o ator obteria ao se envolver em um projeto FOSS; e (iii) *relacionamentos*, como conexões entre atores com base nos elementos (i) e (ii).

Tabela 3.2 – Tipos de papéis chave nos ECOSs. Fonte: (HANSSEN, 2012)

PAPEL	DESCRIÇÃO
Organização Chave ou Proprietário da Plataforma (<i>Keystone</i>)	Uma organização, ou um pequeno grupo, que de alguma forma conduz o desenvolvimento da tecnologia de software central.
Usuários Finais (<i>End-users</i>)	Papel chave para a tecnologia de software central, pois representa quem precisa dela para realizar seu negócio, seja de qual tipo for.
Organizações Externas (<i>Third Party Organizations</i>)	Utilizam a tecnologia de software central como base para produzir soluções ou serviços relacionados (incluem desenvolvedores externos).

Algumas estruturas de classificação foram desenvolvidas por pesquisadores do tema. BOSCH (2009) alega que os ECOSs existem em múltiplas áreas, mesmo que tenham recebido atenção especial no contexto da *Web 2.0*, quando diferentes organizações lutaram para dominar o nicho dos sistemas operacionais para *Desktop*. Na década de 1990, o apoio dos mais influentes e numerosos desenvolvedores terceirizados era reconhecido como crucial, como IBM e Microsoft. Assim, uma *taxonomia dos ECOS* em dois espaços dimensionais pode ser construída (Tabela 3.3). A primeira dimensão representa o nível de abstração ou **categoria** em que o ECOS existe e se divide em três níveis: (i) *sistema operacional* – a primeira categoria onde os ECOSs foram explicitamente identificados e gerenciados, com forte ligação com os dispositivos de hardware, base para diversas aplicações *stand-alone* e para o desenvolvimento de ferramentas computacionais; (ii) *aplicação de software* – categoria para aplicações específicas de domínio, que obtêm sucesso no mercado sem o apoio inicial de um ecossistema, mas evolui para uma comunidade de clientes, desenvolvedores terceirizados e financiadores; e (iii) *programação de usuário*, categoria de menor importância, representa o “santo graau” das plataformas de software ao visar um ambiente de configuração e composição de sistemas para usuários finais. Por sua vez, a segunda dimensão analisa a evolução da indústria de software em termos da **plataforma** dominante: *Desktop*, *Web* e *Mobile*. Para BOSCH (2009), esta taxonomia é um ponto de partida, pois reconhece os argumentos sobre a existência de *mainframes* e

minicomputadores anteriores à plataforma *Desktop*, assim como plataformas posteriores a *Mobile* na era da Computação Ubíqua²⁰.

Tabela 3.3 – Taxonomia dos ECOSs. Fonte: (BOSCH, 2009)

<i>plataforma</i> <i>categoria</i>	<i>DESKTOP</i>	<i>WEB</i>	<i>MOBILE</i>
SISTEMA OPERACIONAL	<i>MS Windows, Linux, iOS</i>	<i>Google AppEngine, Yahoo Developer, Coghead, Bungee Labs</i>	<i>Nokia S60, Palm, Android, iPhone</i>
APLICAÇÃO DE SOFTWARE	<i>MS Office</i>	<i>SalesForce, eBay, Amazon, Ning</i>	Nenhum
PROGRAMAÇÃO DE USUÁRIO	<i>MS Excel, VHDL, Mathematica</i>	<i>Google's mashup editor, MS PopFly, Yahoo! Pipes</i>	Nenhum

A estrutura de classificação proposta por BOSCH (2009) se mostra mais técnica e inspirou novas iniciativas, considerando questões sociais e econômicas, tais como (JANSEN *et al.*, 2009b): (i) as organizações de software precisam ter uma percepção do ECOS em que atuam; (ii) devem perceber as estratégias de sobrevivência que existem entre os seus participantes; e (iii) necessitam de uma visão geral de possíveis caminhos para que possam abrir suas plataformas sem perder de vista a propriedade intelectual. A fim de compreender estes desafios, JANSEN *et al.* (2009a) modelaram os ECOSs em três níveis de escopo (Figura 3.3), cujos objetos de estudo possuem as propriedades de saúde, interação, desempenho, entradas, saídas, competição, compartilhamento de valor e métodos de coordenação:

- O primeiro nível de escopo, chamado **organizacional** (Figura 3.3.a), tem como objetos de estudo os *atores* e seus *relacionamentos* da organização inserida em algum ECOS, visando analisar desempenho e evolução como fatores dependentes dos empreendedores do ECOS. Neste nível, a abertura da organização é a questão alvo, apreciando aspectos de compartilhamento de conhecimento, i.e., P&D, gerenciamento do produto, vendas, *marketing* e suporte, além de pesquisa de mercado e de tecnologia com seus parceiros;
- No segundo nível de escopo, i.e., **visão interna de um ECOS** (Figura 3.3.b), os objetos de estudo são as *redes de produção de software*

²⁰Consiste em um novo paradigma do século XXI, que permite o acoplamento do mundo físico ao mundo da informação e fornece uma abundância de serviços e aplicações ubíquos visando que usuários, máquinas, dados, aplicações e objetos do espaço físico interajam uns com os outros de forma transparente, em *background* (SBCUP, 2012).

formadas dentro de um ECOS, bem como os relacionamentos entre elas. O alvo está no conjunto dos participantes, i.e., fornecedores, clientes, distribuidores e desenvolvedores externos ou terceiros, e nas características internas relacionadas à saúde e à estabilidade do ECOS, i.e., tamanho, tipo, papéis, conectividade etc.; e

- O terceiro nível de escopo, i.e., **visão externa de um ECOS** (Figura 3.3.c), possui como objeto de estudo os *ECOSs per si*, incluindo os relacionamentos entre eles. É importante entender que os ECOSs devem ter suas fronteiras bem definidas: (i) um *mercado*, e.g., *Computer Aided Design* (CAD) ou *Enterprise Resource Planning* (ERP); (ii) uma *tecnologia*, e.g., linguagem de programação *Ruby* e *Internet Protocol version 6* (IPv6); (iii) uma *infraestrutura*, e.g., plataforma Eclipse ou *framework Ruby on Rails*; ou (iv) uma *organização*, e.g., Google e SAP.

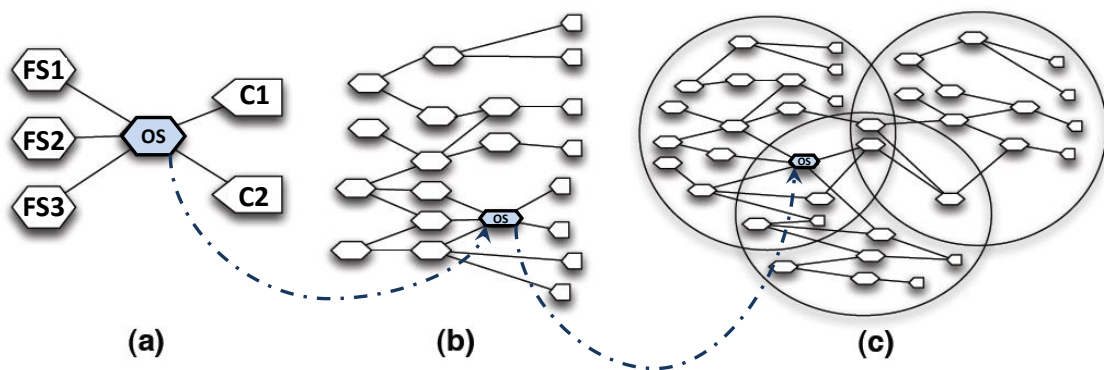


Figura 3.3 – Níveis de escopo dos ECOSs, onde OS = organização de software; FS = fornecedor de software e C = cliente. Fonte: (JANSEN *et al.*, 2009a)

Apesar dessa classificação, ressalta-se que as fronteiras dos ECOSs podem ser sobrepostas durante sua análise, ou seja, um ator pode estar em diferentes ECOSs com diferentes papéis, considerando restrições geográficas, especificações de componentes da plataforma, disponibilidade de licenças para uso, e sua idade e história. Para ilustrar as *perspectivas do ECOS*, um exemplo fornecido por JANSEN *et al.* (2009b) é estendido (Figura 3.4). A *Organização de Software* é provida com componentes do *Desenvolvedor Externo A*, cuja metodologia é baseada em código aberto, e do *Fornecedor de Software X*. A partir destes insumos, a organização entrega um produto (P.1) e um serviço (S.1) para o *Cliente 1* e o *Cliente 2*, respectivamente. É comum que exista uma competição de mercado pela participação de outras organizações que entregam produtos e serviços para outros clientes, e que são providas por outros

fornecedores, como a *Organização de Software I* (JANSEN, 2007). Assim, (i) o *nível da organização de software* (OS), ou *independent software vendor* (ISV), envolve todos os produtos e serviços relacionados à organização, bem como ela mesma, (ii) o *nível da rede de produção de software* (RPS), ou *software supply network* (SSN), considera todos os clientes e fornecedores, sejam produtores ou consumidores de componentes e serviços de software, que têm contato direto com a organização, e (iii) o *nível do ECOS*, ou *software ecosystem* (SECO), reúne todas as organizações de software relacionadas.

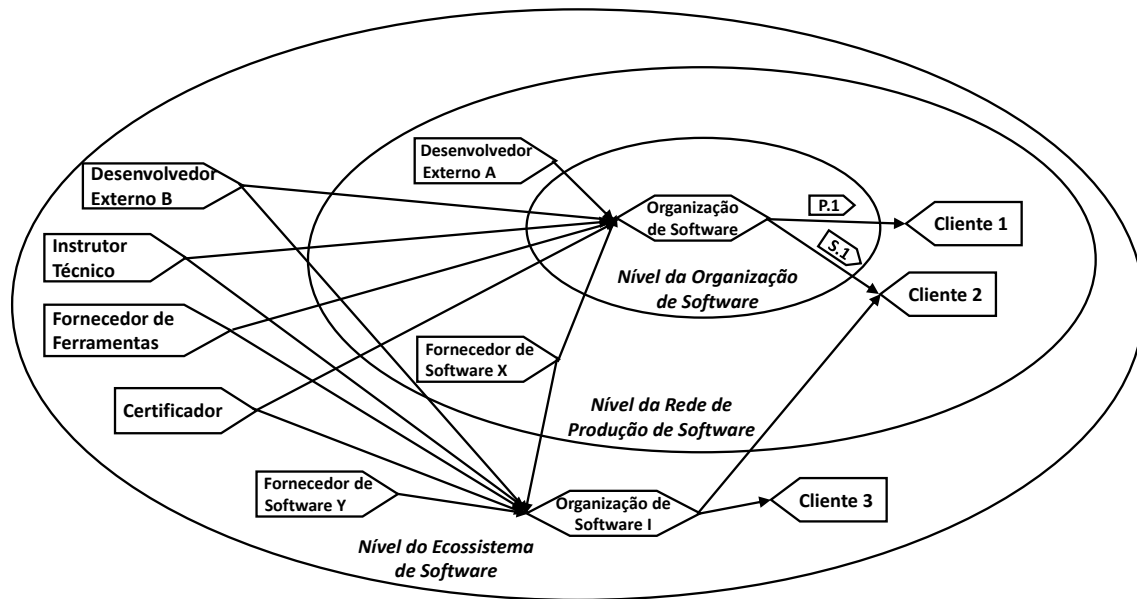


Figura 3.4 – Perspectivas dos ECOSs. Fonte: (JANSEN *et al.*, 2009b)

Por sua vez, segundo CAMPBELL & AHMED (2010), o conceito de ECOS tem raiz nas teorias do desenvolvimento de plataformas similares e das redes sociais e pode ser um caminho para a transição, evolução e inovação da ES. Para isso, apresentam uma visão dos ECOSs em três dimensões a fim de entender as funções assumidas por cada um de seus pilares, conforme mostra a Figura 3.5:

- **arquitetura:** envolve a plataforma (i.e., tecnologia ou infraestrutura) em que o ECOS vai estar inserido, assim como questões da arquitetura de software, LPS e processos de ES;
- **negócio:** envolve o conhecimento sobre o mercado, decisões que os atores devem tomar sobre modelos de negócio, definição do portfólio de produtos do ECOS, estratégias de licenças e de vendas;
- **social:** define a forma como a rede de atores irá se relacionar dentro do ECOS para atingir seus objetivos e fomentar o crescimento do ECOS através de uma proposição de valor onde todos possam obter ganhos.

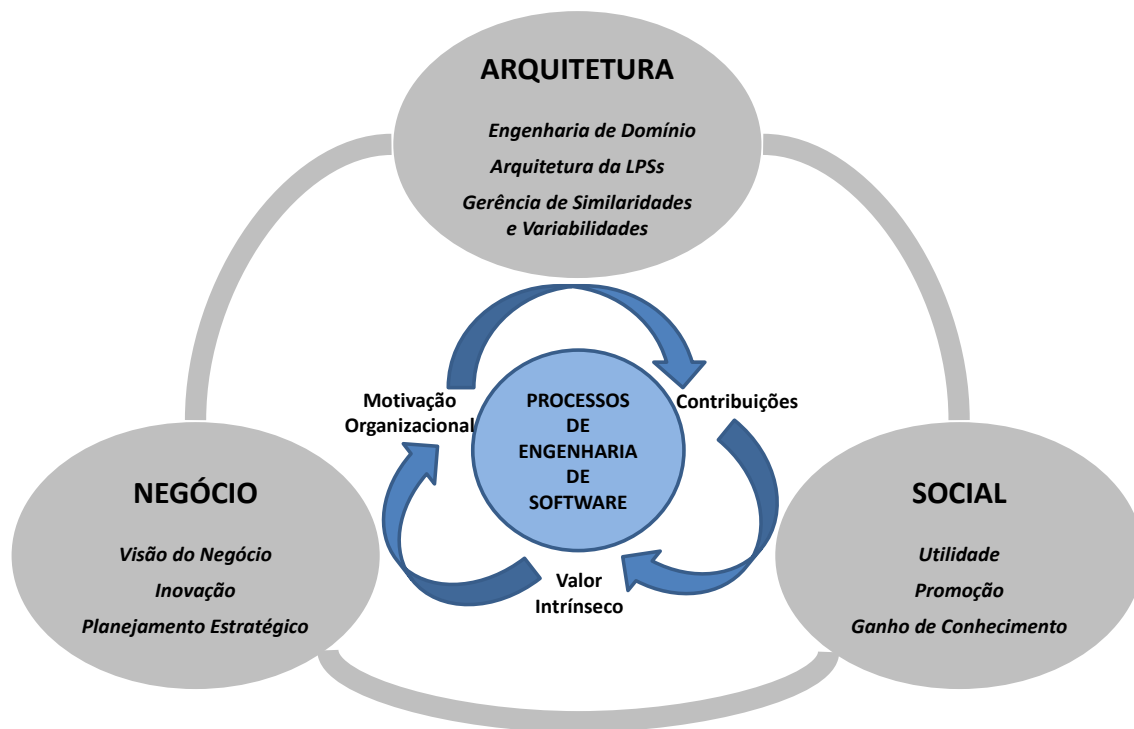


Figura 3.5 – Visão do ECOSs em três dimensões. Fonte: (CAMPBELL & AHMED, 2010)

Uma vez que a dimensão de *negócio* envolve uma transação, o ciclo comercial dos ECOSs pode ser analisado considerando quatro fases, conforme JANSEN *et al.* (2009b): (1) o estabelecimento de um relacionamento de mercado com uma organização chave e focada; (2) o surgimento de uma rede preliminar; (3) a diminuição do poder da organização chave e o estímulo das comunidades; e (4) a manutenção de uma comunidade de criação do ECOS, onde não existem organizações chave e o poder é distribuído. Assim, as dimensões de *arquitetura* e de *negócio* são normalmente alvos de controle pela organização chave, mas a dimensão *social* é dinâmica (BARBOSA, 2012). Ou seja, esta dimensão é a “pedra angular” para o sucesso do ECOS ao impactar as fases do seu ciclo de vida, i.e., nascimento, desenvolvimento, amadurecimento e morte/transformação. Partindo dessa premissa, SANTOS *et al.* (2013a) apresentam uma generalização das três dimensões para tratar transações não comerciais:

- **técnica:** dimensão sobre o entendimento da abertura das plataformas e do envolvimento de atores externos. É dirigida por três fatores:
 - *engenharia da plataforma:* definição, modelagem e construção do sistema de software;
 - *arquitetura:* estrutura dos componentes do sistema de software, relacionamentos e princípios/diretrizes para sua evolução;

- *gerência de requisitos*: funcionalidades específicas/comuns.
- **transacional**: dimensão sobre o entendimento da colaboração ou competitividade, das expectativas dos envolvidos e do impacto na produtividade do seu ambiente. É dirigida por três fatores:
 - *visão do ambiente*: realizar planos / estar aberto a ações situadas;
 - *inovação*: usar a criatividade / atuar conforme um segmento;
 - *planejamento estratégico*: seguir objetivos / ousar oportunidades.
- **social**: dimensão sobre o entendimento da abertura da organização como um todo, das comunidades criadas ao redor da plataforma e do desenvolvimento colaborativo. É dirigida por três fatores:
 - *utilidade*: compensações e recompensas esperadas e percebidas pelos membros da rede, financeiras ou não;
 - *promoção*: reconhecimento implícito ou explícito de capacidades e habilidades do colaborador (i.e., pessoa que contribui);
 - *ganho de conhecimento*: armazenamento e manutenção de experiências e oportunidades para os interessados se envolverem com novas tecnologias e ferramentas no ambiente colaborativo.

Caso a transação não seja comercial, o efeito das plataformas na formação da comunidade com base em *sites* de redes sociais *on-line* (e.g., *Facebook*) pode ser analisado em quatro fases, por meio do ciclo social do ECOS (Figura 3.6), estendido por SANTOS *et al.* (2012a) a partir de (RUSS, 2007) e (JANSEN *et al.*, 2009a):

- **Iniciação**: criação de uma página para a plataforma em um *site* de rede social, a nível organizacional, visando estabelecer um relacionamento de mercado com os demais atores. Isso agrega valor ao relacionamento com fornecedores, clientes, distribuidores e organizações externas e amplifica a capacidade de *marketing*, suporte, pesquisa de mercado e de tecnologia com seus parceiros. Distribuidores podem criar suas páginas e agregar valor à sua rede de relacionamento, incluindo seu relacionamento direto com a organização chave. O fator mais importante para atrair usuários da rede social já existente é o valor do conteúdo dessa página;
- **Propagação**: o contágio social é iniciado pela adesão de novos atores e artefatos, originando o ECOS. Surge uma rede preliminar de atores com interesses em comum (i.e., qualquer pessoa que possua um perfil) ou

páginas de artefatos produzidos por fornecedores ou organizações externas. São criados conteúdos e comentários e são formados grupos ou comunidades, com a consequente diminuição do poder da organização chave. Cresce o estímulo da participação, atração de novos membros e formação de comunidades, quando a massa crítica é alcançada;

- **Amplificação:** estabelecimento de uma estrutura auto organizável e manutenção de uma comunidade engajada e calcada na rede de atores e de artefatos de um ECOS. Nenhuma organização é dominante pelo fato do poder se tornar distribuído, com a vantagem de estar no *site* de rede social e utilizar funcionalidades e recursos de comunicação, colaboração, recomendação e de *marketing*, que ajudam na divulgação e interação;
- **Terminação:** normalmente, um serviço de rede social *on-line* termina devido à saturação ou substituição por um novo serviço, ou ainda porque surgem mercados e tendências que provocam a “evaporação” dos integrantes desta rede e, logo, da comunidade. Pode ser decorrente de um término ou quebra de sustentabilidade/diversidade de um ECOS.

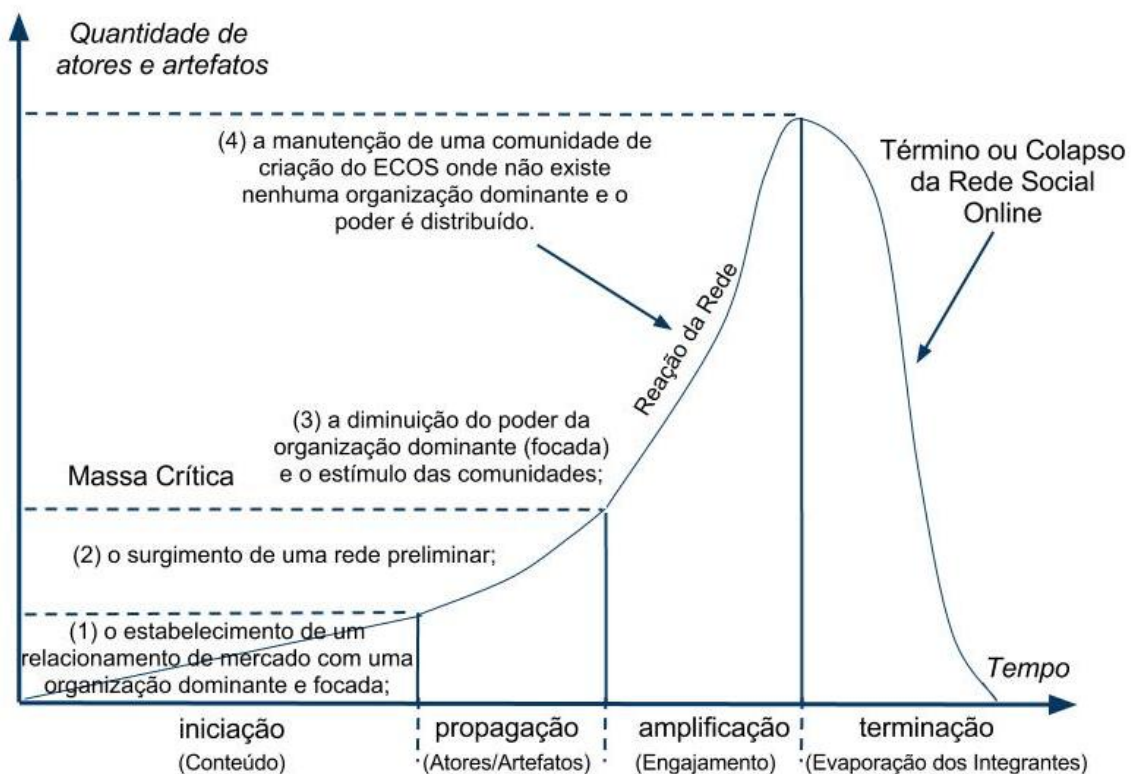


Figura 3.6 – Ciclo de vida social dos ECOSs. Fonte: (SANTOS *et al.*, 2012a)

Nesse contexto, as redes sociais em ECOSs agem de forma semelhante à propaganda boca a boca, onde os consumidores disseminam a sua percepção sobre produtos e serviços, com um poder multiplicador muito maior (SEICHTER *et al.*, 2010). O principal atrativo está em atingir a 800 milhões de pessoas dentro de uma única plataforma (SANTOS *et al.*, 2013b). Fica claro que as organizações que não acompanham a rapidez da inovação de seus mercados propiciada pelas redes em ECOSs podem não conseguir atravessar as barreiras cada vez mais altas (CHESBROUGH, 2003). Todavia, o surgimento de produtos ou serviços não acontece espontaneamente, mas por meio do trabalho estruturado de institutos de P&D, da criação de novos nichos de mercado e da comercialização de novos produtos. Neste contexto, a abertura da plataforma pode estimular a inovação e representa um mecanismo estratégico para a criação da novidade (produto ou serviço), pois atende a dois requisitos: (i) a novidade gera ganho social, por permitir que mais bens e serviços sejam entregues à sociedade; (2) gera retorno para o inovador (financeiro e outros), por ser consumida por ela.

Dessa forma, SANTOS *et al.* (2012a) expandiram o ciclo de vida social para classificar as organizações sob o ponto de vista da adoção de um modelo de inovação nos ECOSs, quando um novo ator pode ser incorporado à fase de *propagação*: o **intermediário** (*technology broker*). Os intermediários têm papel de facilitar o encontro entre oferta e demanda; agregar os atores certos a fim de fomentar a inovação; e apoiar e coordenar o trabalho colaborativo entre os diferentes níveis (SCHILLING, 2008). No caso de ECOSs, isso é mostrado pela Figura 3.7 e pela Figura 3.8, onde a primeira ilustra o processo da *inovação fechada* em um sistema tradicional, i.e., cada organização concebe suas inovações internamente, e a segunda ilustra o processo de *inovação aberta*, onde existem intermediários para agir como mediadores ou avaliadores dos diferentes ECOSs e gerar relatórios ou realizar ações.

Complementando os ciclos de vida comercial e social a partir da visão de MOORE (1993), o ciclo de vida transacional do ECOS pode ser definido (ANDRADE, 2012). Em particular, esta definição considera que as relações sociais entre atores de um ECOS podem ser tanto *cooperativas* como *competitivas*, em uma classificação baseada em analogias com os ecossistemas biológicos (DHUNGANA *et al.*, 2010). Estes conceitos podem ser observados nos ECOSs para entender como os atores interagem e garantem a própria sobrevivência e a do ecossistema como um todo:

- **Nascimento:** é o começo do ecossistema, quando as espécies iniciam a interação. O foco é identificar as necessidades do cliente e definir o que

agrega valor. Potenciais participantes irão aderir ao ECOS e contribuir com produtos ou serviços. Como no ECOS, o nascimento de um ecossistema biológico é um processo confuso e a forma como espécies se unem não é clara. Na natureza, componentes não vivos atraem espécies, conduzindo a uma comunidade mais estruturada. No contexto de software, capital, infraestrutura tecnológica inovadora e potencial de lucro são exemplos de fatores de atração para as organizações;

- **Expansão:** envolve a luta pela conquista de novos clientes nos ECOSs e o crescimento para novos mercados. Disputas entre ecossistemas rivais são comuns durante essa fase. Participantes começam a formar estruturas mais complexas e precisam atuar em redes sociais para aumentar a cooperação com parceiros, a fim de minimizar os efeitos da competição;
- **Liderança:** os participantes do ecossistema usufruem de estabilidade obtida pelo sucesso individual e coletivo. Várias oportunidades de colaboração, tanto interna como externa, são identificadas no ECOS, mas também podem existir disputas entre os participantes para obter mais poder e, consequente, liderança dentro do ecossistema;
- **Autorrenovação:** o ecossistema pode se transformar a partir de efeitos externos, tais como mudanças climáticas, forte competição de rivais ou a própria morte natural de seus participantes. Para garantir um ciclo de renovação nos ECOSs, é necessário que os participantes aumentem a sua capacidade de adaptação às mudanças e à inovação (sustentabilidade e diversidade). Caso contrário, existe o risco de morte do ecossistema.

Por exemplo, um dos ECOS comerciais mais ativos é o ECOS Apple (ANDRADE, 2012), composto por três ecossistemas específicos: ECOS iPhone, ECOS iPad e ECOS Mac. Dado que a Apple não fabrica nenhum destes dispositivos de hardware, ela age em mutualismo com os fabricantes, pois tanto a Apple como os fornecedores (parceiros) podem obter benefícios mútuos pelo crescimento do ECOS. Relação similar ocorre com as organizações ou desenvolvedores externos, que programam as aplicações disponíveis na AppStore. Os concorrentes são responsáveis por retirar valor da organização chave do ECOS, conhecidos como **competidores** ou **dominadores** (*dominators*). No caso do ECOS Apple, o sistema operacional iOS, que é um ECOS, compete diretamente com o ECOS Android, interno ao ECOS Google. Apesar da Google não dominar diretamente o ECOS de dispositivos móveis,

organizações parceiras como Samsung, Motorola Mobility e HTC fabricam dispositivos que utilizam Android. Dessa forma, a competição da Apple/iOS ocorre com várias organizações que utilizam sistemas operacionais rivais. Por outro lado, se a análise for feita a partir do ECOS Google, os papéis são redefinidos, o que caracteriza a presença da variabilidades na definição dos ECOSs. Além disso, observa-se que o ECOS Mac já passou pela fase de autorrenovação, pois o computador Mac está no mercado há quase 30 anos e teve que se reinventar diversas vezes. Se comparado ao sucesso de vendas de 55 milhões equipamentos do ECOS iPad em um ano após o seu lançamento (nascimento), foram necessários dois anos no ECOS iPhone e 22 anos no ECOS Mac para atingir a mesma meta, o que reflete a redução do ciclo de criação de produtos e o intervalo para atingir a liderança (CHESBROUGH, 2003).

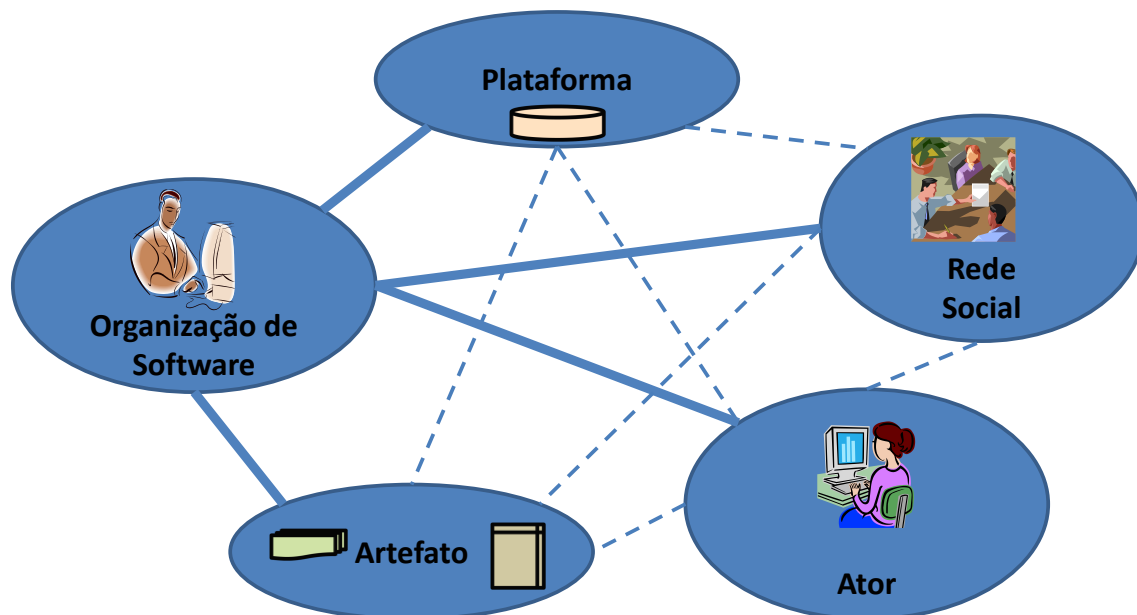


Figura 3.7 – Fontes de inovação fechada em um sistema tradicional, onde as linhas pontilhadas são relações que não estão estabelecidas claramente, dado que a organização de software centraliza as relações. Fonte: (SANTOS *et al.*, 2013b)

No contexto dos ECOSs FOSS, as comunidades de software livre desenvolvem sistemas visando objetivos altruístas²¹ para compartilhar conhecimento e apoiar o desenvolvimento de soluções que possam beneficiar a sociedade, e.g., ECOSs Linux, PHP e Eclipse. Nestes ecossistemas, observa-se uma evolução de forma orgânica, com as decisões tomadas pelos membros mais ativos de cada comunidade. O ciclo de vida destes ECOSs tende a ser bastante longo, se comparado aos comerciais, e.g., o ECOS Linux, ativo e importante desde 1991, com licenças comerciais ou não. Entretanto,

²¹Quando um ser/entidade sacrifica seu próprio bem-estar em benefício do outro (SANTOS *et al.*, 2013a).

organizações como IBM, Oracle e Google têm se associado a projetos FOSS para obter benefícios comerciais na forma de mutualismo. Desta parceria entre organizações FOSS e comerciais, um fenômeno recente criou o conceito de software livre comercial. Várias organizações utilizam partes de código livre, *frameworks*, bibliotecas e componentes para desenvolver sistemas comerciais que serão integrados a outras soluções. Essas organizações têm objetivos estratégicos, pois querem mudar as regras do mercado e reduzir a influência de organizações chave como a Microsoft.

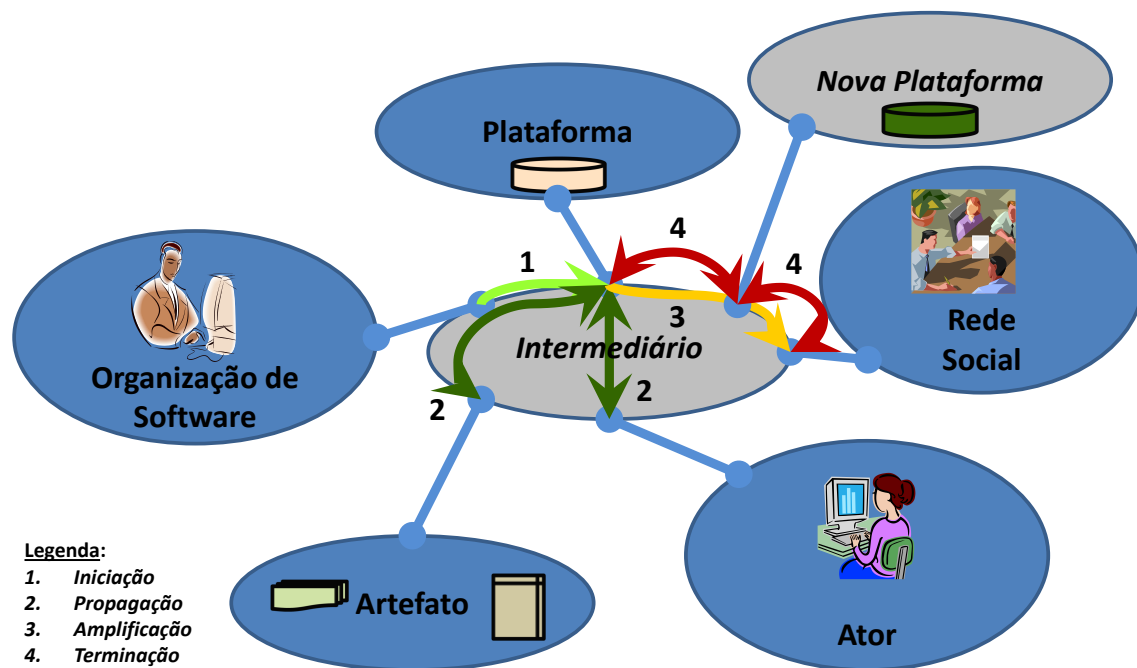


Figura 3.8 – Fontes de inovação aberta em um ECOSs, incorporando aspectos do ciclo de vida social (SANTOS *et al.*, 2012a). Fonte: (SANTOS *et al.*, 2013b)

Por fim, MANIKAS & HANSEN (2013) classificam os *ECOS quanto à missão* em proprietários e/ou FOSS, frente às suas diferenças significativas. No *ECOS proprietário estrito*, o código fonte e os demais artefatos produzidos são protegidos por serem produtos que geram retorno, e os novos atores provavelmente devem se certificar de alguma forma para participar do ECOS. No *ECOS FOSS tradicional*, os atores não necessariamente participam do ecossistema visando obter rendimentos por suas atividades, portanto, sendo mais fácil colaborar, pois tipicamente não requer qualquer atestado de novos atores. No entanto, essa classificação representa um modo simplista usado apenas para delinear as diferenças entre esses tipos de ECOS, dado que a maior parte dos ECOSs seria categorizada como *ECOSs híbridos*, i.e., que combinam elementos desses dois tipos. Em sua revisão sistemática, MANIKAS & HANSEN (2013) observam que as publicações centradas em ECOSs FOSS tratam mais as

questões de natureza técnica e social, ao passo que as publicações que envolvem ECOSs proprietários incluem questões estratégicas e de negócio. Isso é natural, pois projetos FOSS permitem utilizar facilmente técnicas de mineração e de processamento sobre código, *logs* de *commits* etc., embora não haja um modelo de negócio para participação de atores no ECOS como um todo. ECOSs proprietários conseguem prontamente informação sobre estratégias do ECOS e seu posicionamento no mercado, mas o acesso ao código fonte e *commits* de desenvolvedores é mais difícil.

3.3.3. Características

De acordo com BARBOSA & ALVES (2011), existe um número significativo de ECOSs organizados com base em modelos FOSS, o que destaca a importância da colaboração entre os atores como elemento fortalecedor de uma plataforma. Conforme apresentado, os ECOSs também podem ser caracterizados por meio de analogias com ecossistemas biológicos visando a sua compreensão, e.g., utilizando conceitos de mutualismo, comensalismo²², amensalismo, simbiose²³ etc.. Os ECOSs podem ainda estar centrados em uma plataforma representada pelas LPSs (ligação explícita com a Reutilização de Software), ou mesmo estar voltado para um mercado de software e serviços por um ecossistema de negócio, sendo caracterizado por explorar processos de inovação, seja aberta ou fechada. Esses cenários são considerados os catalisadores para os estudos de ECOSs.

Além disso, os ECOSs (BARBOSA *et al.*, 2013): (i) focam na arquitetura da plataforma ao cuidar de questões de estabilidade de interface, gerência de evolução, extensibilidade, segurança e confiabilidade (ALSPAUGH *et al.*, 2009; BOSCH, 2010; CATALDO & HERBSLEB, 2010; HANSEN, 2012); (ii) podem ser utilizados para apoiar processos de negociação de requisitos e alinhar as necessidades reais com soluções, componentes e portfólios (FRICKER, 2009); (iii) podem dar suporte ao desenvolvimento colaborativo para governo eletrônico na forma de ecossistemas digitais (STEFANUTO *et al.*, 2011; LIMA *et al.*, 2013); (iv) são modelados com base em conceitos de redes de produção de software (BOUCHARAS *et al.*, 2010); e (v) impactam as organizações de pequeno e médio porte.

²²Relação em que apenas uma das espécies se beneficia, sem, no entanto, prejudicar ou beneficiar a outra espécie envolvida (SMITH & SMITH, 2008).

²³Relação entre indivíduos de espécies diferentes, em que ambos são beneficiados e em que ambos precisam um do outro para sobreviver (SMITH & SMITH, 2008).

3.3.4. Benefícios e Dificuldades

Um benefício potencial de se tornar um membro de um ECOS é a oportunidade de explorar a inovação aberta, a fim de que os atores colaborem para alcançar benefícios locais e globais, sobretudo em ECOSs FOSS (CHESBROUGH, 2003). O esforço dos atores externos em desenvolver para uma plataforma de ECOS pode transformar a inovação em algo positivo para estes atores (e para seus clientes), ao mesmo tempo em que contribui para a organização chave, ao estender ou melhorar a plataforma e aumentar o número de envolvidos (HANSSEN, 2012). Assim, a proximidade entre a organização chave e os atores externos se torna fundamental para o sucesso de um empreendimento de software. Ou seja, todos os envolvidos passam a usufruir de benefícios diferentes daqueles da ES tradicional, com um baixo nível de colaboração e/ou distribuição, com papéis segregados e com processos fechados. Um ECOS funcionando corretamente se realiza como uma configuração complexa que inclui colaboração além das fronteiras organizacionais fechadas, e não por um esforço planejado e deliberado. As organizações resultam de uma congruência de fatores, tais como desenvolvimento de tecnologia, globalização, novos padrões de colaboração e a participação de clientes cada vez mais acostumados com a modelagem e construção das tecnologias que eles utilizam (HANSSEN & DYBA, 2012).

Outros benefícios identificados com ECOSs são (BARBOSA *et al.*, 2013): (i) acelerar o sucesso do software, seja como componente ou serviço, ou em direção a se tornar uma plataforma de ECOS, por meio de “co-evolução”, que atrai novos atores (CAMPBELL & AHMED, 2010); (ii) reduzir custos inerentes ao desenvolvimento e distribuição de software, além de buscar alternativas na modelagem e manutenção de aplicações distribuídas (BOSCH, 2010); (iii) apoiar análises arquiteturais com o objetivo de escolher que plataforma participar e utilizar (ANVAARI & JANSEN, 2010); (iv) apoiar a cooperação e o compartilhamento de conhecimento, bem como a comunicação de requisitos entre múltiplos envolvidos, e.g., proprietários da plataformas, distribuidores, intermediários etc. (ANGEREN *et al.*, 2011; POPP, 2011); e (v) favorecer tarefas de identificação do negócio, projeto arquitetural de produtos, identificação de riscos e gerência de dependências de software (LUNGU *et al.*, 2010; MENS & GOEMINNE, 2011; MOLDER *et al.*, 2011).

Entre as dificuldades enfrentadas pelos ECOSs, estão (BARBOSA *et al.*, 2013): (i) estabelecer e modelar relacionamentos entre os atores dos ECOSs através de uma

representação adequada para os envolvidos e conhecimentos (BOUCHARAS *et al.*, 2010); (ii) tratar questões arquiteturais e de negócio, e.g., apoio a estratégia de negócio, seleção de arquiteturas adequadas para o processo de desenvolvimento com FOSS e a medição da abertura e flexibilidade da plataforma (BOSCH, 2010; HYRYNSALMI *et al.*, 2012); (iii) lidar com a heterogeneidade de licenças de software e sua evolução no ECOS, dado que as organizações querem mantê-las para reduzir riscos de dependência (POPP, 2012; SALMINEN & MIKKONEN, 2012); (iv) como a organização chave deve se engajar no ECOS a fim de se diferenciar dos competidores através da dimensão social, rompendo obstáculos técnicos e sócio-organizacionais relativos à comunicação de requisitos em projetos distribuídos geograficamente (SANTOS *et al.*, 2013b); e (v) ferramentas para acelerar a interação social, tomada de decisão e desenvolvimento inter-organizacional envolvendo ECOS proprietários e FOSS (POPP, 2012).

3.3.5. Sinais de Teorização

A partir da análise de 72 publicações sobre ECOSs, HANSSEN & DYBA (2012) identificaram teorias que podem motivar um estudo ou desenvolver uma base para esse recente tópico de pesquisa na ES. Por teorias, foram encontradas referências explícitas à *teoria socio-técnica*, tópico bem estabelecido, mas também a conceitos mais amplos como *ecologia organizacional* e *inovação aberta*. Isso permitiu a construção de uma lista de grandes temas para as teorias e conceitos identificados (Tabela 3.4). De acordo com MANIKAS & HANSEN (2013) e BARBOSA *et al.* (2013), a maior parte das publicações correspondem a relatos, os quais documentam o conhecimento e a experiência obtidos, regras básicas, listas de verificação ou observações interessantes, mas não de maneira sistemática para descrever um modelo. Isso significa que não são trabalhos genéricos o suficiente para serem aplicados em diferentes domínios, ou são abstratos demais para prover uma contribuição concreta para o tema.

Destaca-se um conjunto de pesquisas referentes a ferramentas e notações voltadas para ECOS FOSS, a procedimentos e técnicas para resolução de tarefas específicas e a de modelos qualitativos ou descritivos. Juntamente com os relatos, isso compreende a mais de 80% das publicações em ECOSs analisadas por MANIKAS & HANSEN (2013). Recentemente, podem-se incluir dois estudos secundários e dois *frameworks* de pesquisa (Seção 3.3). Apesar disso, pela natureza do tópico, as pesquisas

ainda possuem caráter exploratório, sobretudo por estarem vinculadas diretamente com a indústria de software e por terem sido tratadas primeiro por esta (BOSCH, 2009).

Tabela 3.4 – Identificando fontes de teoria em ECOSs. Fonte: (HANSSEN & DYBA, 2012)

CATEGORIAS	TEORIAS E CONCEITOS	FOCO TEÓRICO
funções e características de ECOSs	<i>abertura/transparência</i>	Entender o ECOS como sistema aberto.
	<i>inovação</i>	Explorar inovação em ECOSs.
	<i>controle e o papel da organização chave</i>	Explicar como os ECOSs são controlados, às vezes, de modo central.
	<i>desempenho e questões econômicas</i>	Explicar o desempenho dos ECOSs, como mecanismo de recompensa.
	<i>teoria da atividade</i>	Explicar a interação de consciência e atividade humana no ambiente.
	<i>teoria de custo de transação</i>	Explicar a coordenação de atividades de criação de valor distribuídas
estrutura e forma de ECOSs	<i>papéis e relacionamentos</i>	Explicar os papéis e os relacionamentos identificados em ECOSs.
	<i>teoria de sistemas (regras de projeto)</i>	Entender um sistema através das interdependências entre componentes.
	<i>teoria socio-técnica</i>	Entender ECOSs como a sobreposição entre sistemas sociais e técnicos.
	<i>teoria de intermediação</i>	Explicar a necessidade e função de intermediários em prover serviços.
	<i>evolução dos ECOS</i>	Descrever como os ECOSs evoluem.
descrição e modelagem de ECOSs	<i>visualização de redes</i>	Descrever e visualizar ECOSs com redes.
	<i>inspiração da Ecologia</i>	Compreender o ECOS como uma variação de ecossistemas biológicos.
	<i>modelagem e taxonomia</i>	Descrever os ECOSs por meio de modelos e componentes.

3.3.6. Impactos na Engenharia de Software

Como MANIKAS & HANSEN (2013) afirmam, ECOSs apresentam problemas em aberto para a área de ES, seja enquanto um produto ou vários sistemas de software. Parte da literatura analisada em sua revisão sistemática está voltada para a utilização de práticas de ES diretamente, e outra parte trabalha com a adaptação dessas práticas para o contexto de ECOS, ambas com foco na plataforma. Questões de arquitetura de software são de suma importância, conforme discutem SANTOS & WERNER (2011b), de modo que deva ser estendida pela natureza dos ECOSs para tratar gerenciamento, regras de negócio e restrições, integração e existência de múltiplas funcionalidades (CATALDO & HERBSLEB, 2010). Segurança e confiabilidade se tornam instrumentos para equilibrar modularidade e flexibilidade na arquitetura da plataforma, visando interoperabilidade e manutenção de interfaces (BOSCH, 2010). Inconsistências causadas por evoluções devem ser verificadas e validadas nas dependências de componentes, o que ressalta a importância de mecanismos de coordenação que afetam

abordagens centradas em processo (BOSCH & BOSCH-SIJTSEMA, 2010). A constante evolução demanda processos adaptáveis, para que o desenvolvimento seja centrado em integração, implantação independente e grupos de liberações de versões da plataforma.

Por sua vez, a análise e o projeto arquitetural passam a incluir princípios, tais como identificar metas de negócio e descrever requisitos significativos, táticas e avaliação arquitetural. O processo de elicitação de requisitos se torna um desafio relevante de ES em ECOSs, pois os novos atores se juntam aos tradicionais envolvidos no processo de desenvolvimento, alguns deles numerosos e distantes do gerenciamento central do ECOS (FRICKER, 2009). Cadeias de valor e cadeias de resultados²⁴ passam a apoiar a engenharia de requisitos (YU & DENG, 2011), além da modelagem, análise e visualização de redes sociais, técnicas e socio-técnicas para identificar redes de influência e de interoperabilidade (SANTOS & WERNER, 2012a; SANTOS *et al.*, 2013b). BOSCH (2009) acrescenta mais duas questões: tratamento da agilidade na ES por meio de ECOSs e composição de produtos e serviços em ECOSs.

Por fim, MANIKAS & HANSEN (2013) listam os 43 ECOSs existentes que têm sido explorados pelas publicações analisadas em sua revisão sistemática, dos quais 30 são discutidos cada um em uma publicação, 12 estão presentes em mais de uma publicação e um não foi nomeado, embora esteja em três publicações. O ECOS Eclipse/Eclipse Foundation é o mais estudado, com sete publicações, seguido pelos ECOSs: GNOME e Open Design Alliance (quatro); Software Público Brasileiro e Linux/Linux Kernel (três); Android, GX Software, Evince, FOSS, FreeBSD, iPhone/iPad App Store e SAP (dois); Apache Web Server, Artop, Braserio, CAS Software AG, CSoft, CubicEyes, Debian, Google Chrome, Google, Gurux, Firefox, HIS GmbH, HISinOne, Mac App Store, Microsoft, Nokia Siemens Networks, Nautilus, Pharo, Ruby, S. Chand Edutech, SOOPS BV, Squeak, Symbian, TFN 200, UniImprove, Unity, US Department of Defense, WattDepot, WinMob e World of Warcraft (um). Somente dois ECOS são fechados (GX Software e SAP), o que pode refletir o desafio de conseguir dados desse tipo de ECOSs, ao contrário dos ECOS FOSS, que apresentam repositórios públicos disponíveis, conforme apresentado na Seção 3.3.2 e mostrado em estudos como (SANTANA & WERNER, 2013).

²⁴Cadeias de resultados consistem em modelar os atores pelo ciclo composto por três elementos: a partir de uma suposição, uma iniciativa deve ser definida e visa a uma contribuição, que conduz a um resultado. É utilizada no planejamento estratégico de mercados (BOEHM & HUANG, 2003).

3.3.7. Grupos e Tópicos de Pesquisa

BARBOSA & ALVES (2011) listam as instituições acadêmicas mais ativas com base nos resultados do seu mapeamento sistemático: *Utrecht University*, na Holanda; *University College London*, no Reino Unido; *Boston University*, nos EUA, *Babson College*, nos EUA, e *University of Lugano*, na Suíça. Apesar de não haver muitas publicações da *London School of Economics* e da *Imperial College London* (Reino Unido), estas instituições têm grupos de pesquisa emergentes, como é o caso de países como Finlândia, Suécia, Noruega, Dinamarca e Canadá. No Brasil, três grupos estão trabalhando diretamente com ECOSs: Universidade Federal do Rio de Janeiro (COPPE/UFRJ), Universidade Federal de Pernambuco (CIn/UFPE) e, mais recentemente, Universidade Federal da Bahia (DCC/UFBA). Isso propiciou a apresentação de alguns tutoriais sobre o tema, como no *VII Simpósio Brasileiro de Sistemas de Informação* (SANTOS & WERNER, 2011a), *XV Conferência Iberoamericana de Software Engineering* (WERNER & SANTOS, 2012) e *III Congresso Brasileiro de Software: Teoria e Prática* (WERNER et al., 2012).

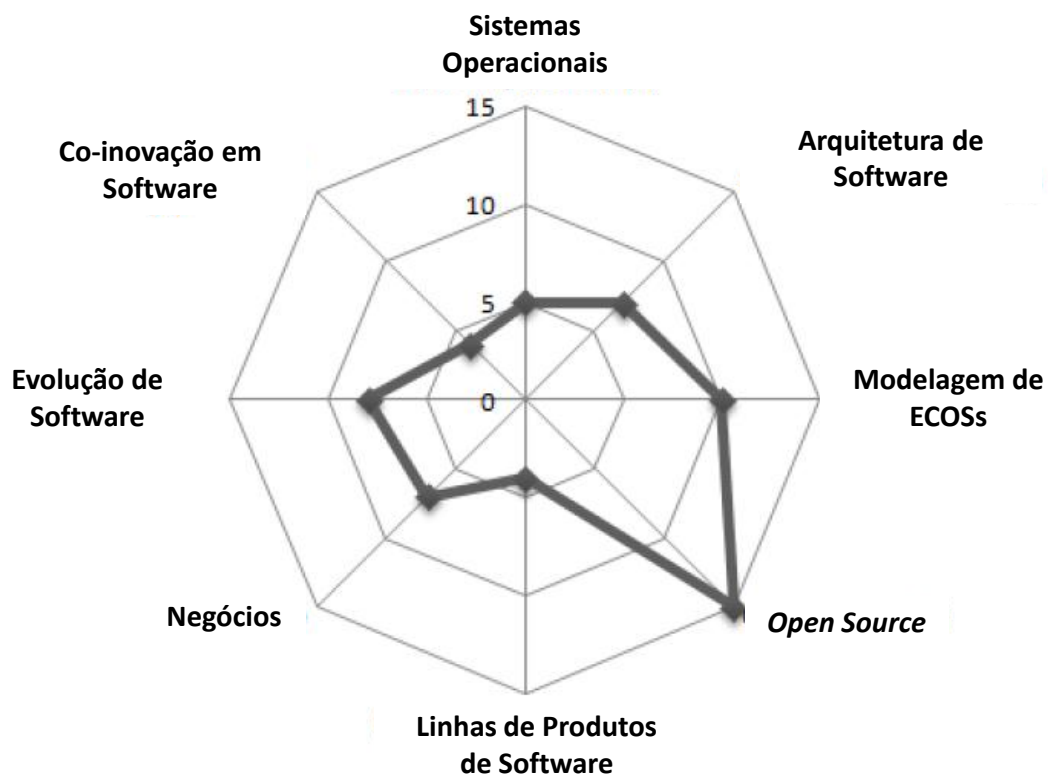


Figura 3.9 – Tópicos de pesquisa mais investigados em ECOS. Fonte: (BARBOSA & ALVES, 2011)

O tema conta com um livro publicado em 2003, *Software Ecosystem: Understanding an Indispensable Technology and Industry* (MESSERSCHIMITT &

SZYPERSKI, 2003), e passa a contar com um novo livro em 2013, *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry* (JANSEN *et al.*, 2013). Por fim, BARBOSA & ALVES (2011) resumizam os tópicos de pesquisa mais investigados em ECOSs (Figura 3.9). Estes tópicos contemplam as três dimensões dos ECOSs pela análise feita por SANTOS *et al.* (2013a): *técnica* (arquitetura de software, sistemas operacionais, evolução de software e LPSs), *transacional* (modelagem, co-inovação e negócios) e *social* (FOSS).

3.4 Desafios em Ecosystemas de Software

Alguns desafios e fatores de sucesso foram apontados por BOSCH (2009) e JANSEN *et al.* (2009b), sumarizados na Tabela 3.5: (i) caracterizar e modelar ECOSs, considerando diferentes perspectivas, e.g., no Brasil, a organização e evolução destes ecossistemas podem depender diretamente da atuação do Estado, ou seja, de questões políticas; (ii) estabelecer os relacionamentos entre as redes de produção de software; (iii) gerenciar a qualidade em ECOSs; (iv) planejar portfólios e linhas de produtos em ECOSs; (v) prover métodos, técnicas e ferramentas para desenvolver arquiteturas de sistemas que atendam à extensibilidade, à portabilidade e à variabilidade; e (vi) tratar implicações gerais em ES, incluindo mecanismos de comunicação, agilidade na concepção e desenvolvimento de soluções e composição de produtos e serviços. Nas subseções a seguir, são listados os desafios por categoria, i.e., desafios discutidos no período 2009-2010, no período 2011-2012, no cenário nacional e uma conclusão.

Tabela 3.5 – Desafios para ECOSs. Fontes: (BOSCH, 2009) e (JANSEN *et al.*, 2009b)

NÍVEIS	DESAFIOS	DESCRIÇÃO
Organizacional (ISV)	#1: <i>Portfólio e planejamento da linha de produção</i>	Gerenciar configurações e reutilização para funcionalidades da plataforma.
	#2: <i>Gestão do conhecimento</i>	Aplicar mineração para extrair e visualizar <i>feedback</i> dos envolvidos.
	#3: <i>Arquitetura extensível, portátil e com variabilidade</i>	Explorar interfaces, requisitos e comunicação sobre plataformas.
	#4: <i>Integração de sistema de software na organização</i>	Desenvolver <i>frameworks</i> para gerência do produto de software e seus ciclos.
Rede de Produção (SSN)	#1: <i>Estabelecimento das relações</i>	Identificar papéis e relacionamentos.
	#2: <i>Tempo de liberações</i>	Gerenciar versões e seus impactos.
	#3: <i>Gestão de qualidade</i>	Gerar diretrizes para certificação.
Ecossistema (ECOS)	#1: <i>Caracterização e modelagem de ECOS</i>	Identificar tamanho, vizinhança, padrões e papéis no ECOS.
	#2: <i>Políticas e estratégias entre ECOSs</i>	Determinar instrumentos para permitir orquestração em cada ECOS.
	#3: <i>Determinação da estratégia para vencer e lucrar na rede</i>	Integrar modelos de negócio à ES.

3.4.1. Desafios 2009-2010

Devido à imaturidade do tópico de pesquisa, ainda em estabelecimento e definição, foi realizado um *brainstorming* no encerramento do IWSECO 2010 em Copenhague, Dinamarca, para estabelecer que casos de ECOSs na indústria são interessantes para análise e quais são as questões e desafios de pesquisas. Este *brainstorming* reuniu em torno de 30 participantes, entre ouvintes e apresentadores de trabalhos técnicos, e teve como mediador o pesquisador e consultor Jan Bosch, que conduziu a discussão a partir dos vários aspectos contemplados pelos artigos publicados. Devido ao caráter prático dos ECOSs, alguns casos foram eleitos como alvos de estudos de caso, entre eles, *Drupal*, *Eclipse*, *iPhone / iPod*, *Android*, *DoD Systems* e *Microsoft CRM*. Do ponto de vista acadêmico, algumas questões foram discutidas para nortear os trabalhos da comunidade de pesquisa, entre elas:

1. Como construir e evoluir a plataforma de um ECOS, uma vez que ele é conduzido pela comunidade subjacente?
2. Por que os ECOSs “morrem” ou “desaparecem”?
3. Como transformar atores em organizações chave ou competidoras no contexto dos ECOSs?
4. Como decidir qual ECOS iniciar, desenvolver e monitorar?
5. Como definir os ECOSs sob a perspectiva dos envolvidos?
6. Como definir o escopo e as características de um ECOS?
7. Como ampliar a classificação dos três tipos de plataformas de ECOSs (sistema operacional, aplicação de software e programação de usuário)?
8. Quais os recursos financeiros diretos e indiretos abarcados nas transações realizadas dentro dos ECOSs?
9. Qual o papel de padrões e dos dados históricos nos ECOSs?
10. Quais as conexões existentes entre os ECOSs e ED/DBC/mercados de componentes livres?

3.4.2. Desafios 2011-2012

De maneira similar ao que aconteceu no IWSECO 2010, em 2012, um novo *brainstorming* foi realizado no IWSECO em Boston, EUA. Diferentemente do primeiro *brainstorming*, quando o *workshop* foi realizado no contexto da *European Conference on Software Architecture* e, anteriormente, na *International Conference on Software*

Reuse, ambas de foco mais técnico, desta vez ocorreu no contexto da *International Conference on Software Business*, após dois anos co-localizado com esta conferência, de foco em negócios e no mercado. A seguir, são listados os desafios discutidos a partir da questão “*Como os ECOSs evoluem?*”:

1. Organização e ecossistemas – estrutura e usuários:
 - O que mantém um ECOS integrado? Informações, indivíduos, software, organizações, coordenação e colaboração?
 - Quais os benefícios potenciais e efetivamente percebidos, a fim de entender o que o faz único em comparação com plataformas tradicionais?
 - Como tornar explícitos os fatores de sucesso do ECOS?
2. Interação entre a evolução do ECOS e a evolução da indústria:
 - Como desenvolver e expandir um ECOS do zero? Relacionar ECOSs e modelos de negócio a fim de melhorar o seu entendimento?
 - Quais as ligações entre comportamento e estrutura em ECOSs?
 - O que significa retorno de investimento em ECOSs? Como as mudanças econômicas e as mudanças tecnológicas modificam a estrutura do ECOS?
3. Padrões para auto regulação em ECOSs:
 - Como pequenas organizações constroem e se beneficiam de possíveis ECOSs?
 - Como gerenciar o decaimento do ECOS e dos competidores sob o ponto de vista da organização chave?
 - Qual é o ponto de equilíbrio entre colaboração e competição em ECOSs? Ecossistemas para desenvolvimento de dados de forma colaborativa e aberta e ecossistemas FOSS? Que propulsores tornam a plataforma mais atrativa para candidatos a complementadores?
 - Quais as ameaças aos ECOSs? Quais os custos envolvidos na quebra de um ECOSs existente?
 - Como os usuários finais participam dos ECOSs?
4. Inovação aberta, requisitos abertos e criação de valor:
 - Que tipos fundamentais de ECOSs podem ser distinguidos?

- Que mudanças as organizações podem fazer na API da plataforma e por quê? Quais os efeitos em pequenos agentes?
 - Quais as questões relacionadas à qualidade de dados para os pesquisadores em ECOSs?
 - Quais as relações entre ECOSs e ecossistemas de ECOSs?
 - Quais os relacionamentos entre projeto de plataforma e sucesso do ECOS? Por exemplo, como tratar abertura *versus* fechamento, modularidade *versus* integralidade, entre outros níveis?
5. Conclusão: como ajudar a produção de software e organizações FOSS a alavancarem ECOSs e alcançarem benefícios econômicos, ao passo que como as lições destas podem ser aplicadas a organizações como Microsoft e Apple?

3.4.3. Desafios sob o Ponto de Vista Nacional

Uma vez que até meados de 2012 o foco das pesquisas em ECOSs estava em grandes plataformas, seus conceitos não foram explorados pelas plataformas brasileiras, embora uma plataforma governamental integrada possa representar um estudo de caso significativo, pois o Brasil é um dos países com maior destaque nos avanços de governo eletrônico (ALVES *et al.*, 2009; FRANZOSI *et al.*, 2009; ALVES & PÊSSOA, 2010; ALVES *et al.*, 2011; STEFANUTO *et al.*, 2011; LIMA *et al.*, 2013). Além de pesquisas com a aplicação da Reutilização de Software, considerada estratégica para o futuro da ES brasileira (SOFTEX, 2007), a utilização de metodologias ágeis e modelos de qualidade de software específicos como o Modelo MPS (SOFTEX, 2012) pode criar oportunidades para gerar um grande ECOS a partir de um conjunto de micro, pequenas e médias organizações (SANTOS & WERNER, 2011b). Na Tabela 3.6, são sumarizados desafios de um *roadmap* de pesquisa para ECOSs, propostos por SANTOS *et al.* (2012c) sob o ponto de vista nacional.

3.5 Considerações Finais

Pelo papel relevante que a TI vem desempenhando na construção de ECOSs, os desafios apontados em diferentes cenários representam uma contribuição para a ES em um momento em que a área passa por questionamentos sobre seu processo de evolução enquanto campo de pesquisa (BOEHM, 2006). É importante considerar que a inovação

é constante em ECOSs e que outras tecnologias podem se unir para fortalecer as redes de produção de software centradas em uma plataforma e nos envolvidos. A integração de tecnologias e serviços através da *Web* e *Mobile* já é uma realidade, mas novas possibilidades vêm sendo exploradas (SANTOS *et al.*, 2013a). Por exemplo, o gerenciamento de informações evolui rumo à convergência digital para integrar diferentes mídias em um único ambiente, com a construção de novas funcionalidades como contribuições de desenvolvedores externos.

Além disso, a colaboração dentro das redes de produção de software pode ser ainda mais estimulada pela adoção de software social (e.g., *Wikis*, *blogs* etc.), que permite aos atores interagir e compartilhar conhecimento numa dimensão social, realçando o potencial humano de criação em vez de apenas privilegiar a transmissão um para muitos. Neste capítulo, apresentou-se a evolução da visão da ES a fim de contemplar os ECOSs sob o ponto de vista da reutilização, o que serviu de base para discutir o estado da arte dos ECOSs e os seus desafios. Esta motivação direcionou as primeiras pesquisas realizadas no tema dentro da ES (SANTOS, 2010). No entanto, as diferentes direções de pesquisa apontadas pelas publicações e discussões sobre temas de trabalho e casos na indústria mostram que a área tem muito a explorar sob diferentes perspectivas dentro da ES (Figura 3.10).

Tabela 3.6 – Roadmap de pesquisa para ECOSs em ES no Brasil. Fonte: (SANTOS *et al.*, 2012c)

DESAFIOS PARA ECOSs
(1) FOSS ou proprietário: qual é o melhor cenário? – É importante identificar os impactos do usuário final como um ator externo no ECOS, a fim de estender técnicas de ES para aplicar nas plataformas de ECOSs por meio da modelagem de relacionamentos entre organizações chave e externas.
(2) Qual é a estratégia mais inovadora no desenvolvimento de software globalizado? – É importante mapear os pontos fortes e fracos das plataformas de ECOSs para definir e automatizar um <i>framework</i> para o seu gerenciamento e monitoração, local e globalmente.
(3) Qual é a estratégia mais inovadora em LPS? – É importante analisar questões arquiteturais da plataforma, tais como estabilidade de interface, segurança, confiabilidade e estratégia de negócio, a fim de integrar e experimentalmente validar atividades de ECOS na engenharia da LPS.
(4) Como executar o processo de abertura da arquitetura da plataforma? – É importante investigar arquiteturas adaptáveis para os cenários de desenvolvimento FOSS e proprietário, além de avaliar quão aberta e flexível está a arquitetura para apoiar a gerência de contribuições comuns de diferentes plataformas.
(5) Como explorar ECOSs na ES Brasileira? – É importante modelar as plataformas nacionais, e.g., Software Público Brasileiro, elencado como o quarto ECOS mais estudado segundo MANIKAS & HANSEN (2013), além do Modelo MPS, a linguagem de programação Lua etc., bem como aquelas desenvolvidas por pequenas e médias organizações. O objetivo é identificar as suas particularidades diante das plataformas globais e dar suporte a diagnósticos baseados no ciclo de vida para analisar a co-inovação e a maturidade dos ECOSs.

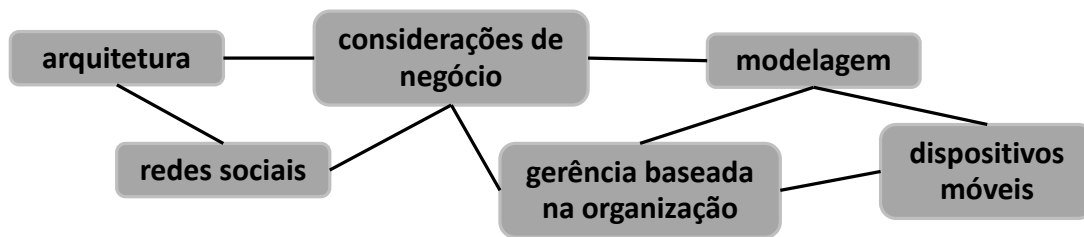


Figura 3.10 – Tendências de ECOSs na área de ES. Fonte: (WERNER *et al.*, 2012)

Como sumarização dos desafios dos ECOSs, pode-se destacar dois pontos de vista, como mostra a Figura 3.11 (BARBOSA *et al.*, 2013). Do ponto de vista do **gerenciamento**, tem-se: (i) como lidar com ecossistemas do tipo *open source*, i.e., plataformas abertas e as redes constituídas; (ii) como implementar a governança de ecossistemas, i.e., como gerir estratégias de sobrevivência e agregação da comunidade em torno de uma plataforma diante de possíveis competidores e/ou colaboradores; e (iii) como realizar a análise de ecossistemas, i.e., como identificar, coletar, relacionar e extrair informação a partir de redes constituídas em torno de uma plataforma. Por outro lado, do ponto de vista da **engenharia**, tem-se: (i) como abrir a plataforma, i.e., como permitir que organizações externas à organização chave contribuam para a evolução da plataforma, além de produtores e consumidores de componentes, serviços e aplicações, que alimentam o ECOS; (ii) como manter a qualidade da plataforma, i.e., como verificar se a plataforma continua atendendo às necessidades da comunidade constituída ao seu redor, bem como os atores e relacionamentos que a compõem, considerando suas funcionalidades e componentes; e (iii) como modelar a arquitetura da plataforma, i.e., como planejar o processo de construção e evolução da plataforma considerando questões técnicas, modelos transacionais e redes sociais.

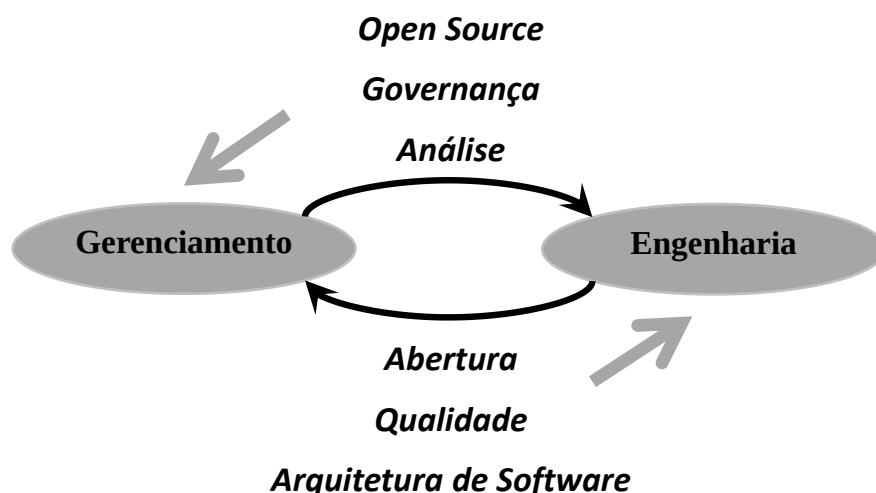


Figura 3.11 – Desafios em ECOSs enquanto área de pesquisa. Fonte: (BARBOSA *et al.*, 2013)

Capítulo 4 – Uma Abordagem para Engenharia e Gerenciamento de Ecossistemas de Software Baseada em Reutilização

4.1 Introdução

A criação de um novo produto de software frequentemente envolve grandes equipes de especialistas colocando em prática muitas habilidades, a um custo alto de coordenação. Obter um produto de sucesso requer comunicação próxima com usuários finais e clientes, incluindo gerentes e operadores (BOSCH, 2009). Isso transforma a criação de um software em um desafio *organizacional* e *de gerenciamento*, ao mesmo tempo em que *técnico* e *de engenharia* (SANTOS & WERNER, 2011b). Por um lado, um dos diferenciais entre os fornecedores de organizações de software está no seu processo de desenvolvimento, que pode afetar valor agregado, produtividade, custos e *time-to-market*. Ou seja, o sucesso de criar um software que seja viável, implantável e usado nem mesmo é assegurado frente ao envolvimento de terceiros. Por outro lado, uma quantidade significativa de grandes projetos de software falha, muitas vezes devido à carência de habilidades e à experiência das organizações para lidar com questões econômicas e sociais da ES, considerando o contexto global (BOEHM, 2006).

Nesse cenário, emergem as chamadas *aplicações sociotécnicas*, que consistem em componentes, serviços e aplicações construídos na indústria de software que, além de relacionar atividades de prototipação e de validação, contam com a participação dos usuários na sua divulgação, distribuição/venda e evolução. O usuário se transforma em mais um desenvolvedor de software, direto ou indireto, de modo que as organizações passam a lidar com o desafio de manter o controle sobre essas aplicações e suas tecnologias, na socialização da Engenharia de Software (ES) (SALMINEN & MIKKONEN, 2012). Isso impacta o uso de cadeias de valor e de resultados tradicionais, pois as primeiras modelam apenas um empreendimento de software por modelo de transferência de valor, de forma unilateral, como também acontece com as segundas, que identificam atores envolvidos em um objetivo a ser alcançado pela organização (Seções 3.1 e 3.3.6). Surge então o conceito de *rede de produção de software* (RPS) para modelar as organizações que desenvolvem e utilizam componentes *comercial-off-the-shelf* (COTS) e arquiteturas em camadas ou *plug-ins* e lidam com uma

rede complexa de fornecedores. Ou seja, as organizações não são mais monolíticas, i.e., não lidam somente com seus próprios clientes. Nas RPSs, são construídos modelos visuais para relacionar organizações de software, de hardware e de serviços que cooperam para satisfazer demandas de mercado, de forma multilateral (JANSEN, 2007).

Alguns estudos utilizam direta ou indiretamente as RPSs para criar notações que apoiem na compreensão da complexidade das relações na indústria diante dos impactos econômicos, afim de maximizar a percepção do *senso de negócio* (BOUCHARAS *et al.*, 2009; PETTERSSON *et al.*, 2010; ANGEREN *et al.*, 2011; YU & DENG, 2011; MCGREGOR, 2012). No entanto, pela socialização da ES, os impactos da comunidade também devem ser contemplados pelas RPSs, pois o ciclo de criação, fornecimento e operação de uma aplicação sociotécnica ocorre no contexto de um emaranhado de plataformas variadas e sobrepostas, que reúnem vários tipos de organizações e usuários, conhecidos ou não. Esses atores contribuem para (e dependem da) propagação, amplificação e expansão dessas plataformas para obterem sucesso com as aplicações sociotécnicas construídas pela indústria de software. Manter a sustentabilidade e a diversidade das plataformas é crítico para os interesses individuais e/ou coletivos dos envolvidos. O *senso de comunidade* alerta quanto à necessidade de estratégias para agregar diferentes atores, a fim de transformar modelos de negócio tradicionais para tratar transações e transferências em cadeias de valor clássicas e abertas (Seção 4.2.2).

Com uma indústria composta por elementos dinâmicos e instáveis, as organizações de software deixam de gerar valor somente em função da produtividade gerida internamente, sejam suas plataformas proprietárias ou abertas. Isso significa que elas vêm sendo forçadas a inovar e melhorar, além de tradicionalmente saberem “o que” e “como” inventar (GOLDMAN & GABRIEL, 2005). Consequentemente, elas devem reconhecer o valor e o talento existentes além de suas fronteiras e criar meios para utilizar a habilidade de atores externos e/ou desconhecidos, sem necessariamente incorporá-los²⁵, enquanto mantêm a sua vantagem competitiva. Conforme discutido nos Capítulos 3, pesquisadores e profissionais da prática da ES começaram a repensar o processo de desenvolvimento com a metáfora de *ecossistemas*, como aconteceu nas áreas de Administração e de Sociologia, sobretudo questões destas áreas representam

²⁵Uma vez que os atores, além de desenvolvedores, incluem organizações, a sua incorporação por uma organização corresponderia à aquisição ou fusão com outras organizações ou tecnologias. Por exemplo, a *Borland International* aos poucos gerou uma de suas plataformas de maior sucesso, o Turbo Pascal, a partir de movimentos de mercado em direção ao *Blue Label Software* (BLS) *Pascal*, desenvolvido e mantido pela *PolyData* (MESSERSCHIMMITT & SZYPERSKI, 2003).

desafios para a indústria de software (BOEHM, 2006). A abordagem dos Ecossistemas de Software (ECOSs) visa enxergar melhor o software como uma parte de um sistema ao estreitar suas relações com modelos de negócio e de colaboração, inovação aberta e proposição de valor, a partir de uma ou mais plataformas (HANSSEN & DYBA, 2012).

Sob o ponto de vista da ES, questões organizacionais (processo) e técnicas (produto) se entrelaçam ao considerar o elemento social (sistema). Isso permitiu que elementos da Teoria da Ecologia Organizacional (TRIST, 1977) fossem explorados para Ecossistemas de Software (ECOSs) e que HANSSEN (2012) derivasse as seguintes proposições: (1) organizações que pertencem a um ECOS se relacionam com uma organização chave, que atua como organização de referência, embora muitas delas estejam parcialmente sob controle ou ligadas por relações de interface; (2) ECOSs promovem a auto regulação; (3) ECOS possuem um caráter de rede; (4) ECOSs existem pela utilização de tecnologia; (5) ECOSs têm valores compartilhados; (6) o valor compartilhado em um ECOS é tanto proveniente do domínio do produto como do negócio; e (7) conforme o ECOS surge, o controle e a influência no desenvolvimento se tornam responsabilidade compartilhada entre as organizações chave e competidoras com o ambiente externo.

Os ECOSs vêm sendo apontados como o futuro da pesquisa em reutilização, por representarem um próximo passo para as LPSs, visando explorar o núcleo de artefatos além do contexto intra-organizacional (BOSCH, 2009). Dessa forma, expõe-se o potencial da reutilização a um ambiente ou mercado mais amplo e mais aberto para as aplicações, componentes e serviços (SANTOS & WERNER, 2010a; 2013a). Para SEICHTER *et al.* (2010) e JANSEN & CUSUMANO (2012), os ECOS correspondem a *frameworks* para reutilização que tratam: (i) *em alto nível*, os grupos de organizações criando software; e (ii) *em baixo nível*, ferramentas e artefatos funcionando como software. Não obstante as recentes pesquisas para analisar os ECOSs e contribuir para os desafios da ES, a Seção 1.3 mostrou que um problema anterior permanece em aberto, referente à compreensão de seus conceitos e relações, i.e., *elementos* (atores e artefatos) e *conexões* (relacionamentos). Ao lidar com o cenário interorganizacional, a análise de um ECOS passa a considerar não só a visão interna, mas também a visão externa. Nesse sentido, busca-se maximizar os sentidos de negócio e de comunidade na ES globalizada através da definição e modelagem destes aspectos como elementos de ECOSs.

Atualmente, isso não acontece, pois os estudos centram suas análises em um ECOS, o que não permite navegar entre as suas *fronteiras*, i.e., mercado, tecnologia,

infraestrutura ou organização, ou mesmo explorar questões de interesse em *subdomínios* ou *casos de ECOSs*, i.e., subconjuntos de elementos e de conexões do(s) ECOS(s). Em outras palavras, frente às divergências conceituais e às definições vagas no tema (HANSSEN & DYBA, 2012; MANIKAS & HANSEN, 2013; BARBOSA *et al.*, 2013), o domínio de ECOSs precisa ser construído, o que contribui seja do ponto de vista teórico, para melhor entendimento da academia, ou prático, para melhor visualização da indústria. As RPSs não apoiam a modelagem da *variabilidade de ECOSs*, i.e., das variações que atores e artefatos e relacionamentos podem sofrer em termos de papéis e responsabilidades. Assim como a variabilidade de elementos e suas conexões nos ECOSs interfere na compreensão em *nível conceitual*, em *nível concreto*, também impacta artefatos reutilizáveis manipulados por Engenharia de Domínio (ED), Linha de Produto de Software (LPS) e Desenvolvimento Baseado em Componentes (DBC), que passam a ser vistos como ativos de ECOSs ao incluírem os atores e incorporarem aspectos dos sentidos de negócio e de comunidade.

Nesse sentido, a abordagem de ECOSs pode utilizar de técnicas de reutilização para sua compreensão em nível conceitual (*domínio de ECOSs*), ao mesmo tempo em que contribui para a área de reutilização ao tentar elevar o nível de abstração de suas técnicas para lidar com as tradicionais barreiras “também técnicas”²⁶. Mais especificamente, o primeiro caso está associado ao problema enunciado nesta proposta de tese: *a compreensão dos conceitos e de suas relações entre diferentes ECOSs diante da abertura de suas plataformas na indústria de software*. Portanto, este capítulo propõe uma solução para o problema discutido, cujo objetivo é construir uma abordagem baseada em reutilização para dar suporte a Engenharia e Gerenciamento (E&G) de ECOSs. Denominada *Reuse-based Software Ecosystems Engineering and Management (ReuseSEEM)*, a abordagem explora, para isso, o conceito de domínio no ambiente de desenvolvimento para ECOSs (DPE). O foco está em criar uma abordagem integrada para definição, modelagem e análise de conceitos e suas relações na visão interna do ECOS, mas também nas redes que se constituem a partir da visão externa.

Para isso, o *framework* para E&G de ECOSs foi definido inicialmente com base nas iniciativas da comunidade de pesquisa no tema, como resultado de uma agregação

²⁶Segundo CUKIERMAN *et al.* (2007), a expressão “não técnico” agrega caráter pejorativo ao que “está de fora” do técnico, reforçando a dissociação ou faturação entre as questões que constituem a ES. Assim, nesta proposta de tese, será adotada a expressão “também técnico”, a fim de reforçar as imbricações de diversas questões envolvidas na ES.

de trabalhos publicados nas edições do IWSECO (2009-2012), refinado posteriormente por dois estudos secundários no tema (BARBOSA & ALVES, 2011; MANIKAS & HANSEN, 2013). O *framework* foi utilizado na análise de casos de ECOSs do Laboratório de Reutilização da COPPE/UFRJ (LENS/Reuse) com suas dimensões *técnica, transacional e social*. Os resultados permitiram a proposta da quarta dimensão de E&G, bem como de uma infraestrutura de apoio. Com esta pesquisa, espera-se gerar uma melhor compreensão e percepção do(s) ECOS(s) na indústria de software global e dinâmica por uma base de conhecimento para interessados, tratando suas três dimensões. O capítulo está organizado da seguinte forma: a Seção 4.2 apresenta o *framework* para E&G de ECOSs nas três dimensões, analisando casos do LENS/Reuse; a Seção 4.3 discute a proposta de ECOSs baseada em reutilização, exibindo a quarta dimensão do *framework* e uma infraestrutura de solução; a Seção 4.4 faz uma análise dos trabalhos relacionados; finalmente, a Seção 4.5 conclui o capítulo com algumas considerações finais.

4.2 Um *Framework* para Engenharia e Gerenciamento de Ecossistemas de Software

Compreender um ECOS a partir da estrutura de classificação em níveis de escopo de JANSEN *et al.* (2009b) requer foco no *escopo* do ECOS. Cada nível possui desafios de pesquisa distintos, i.e., do efeito das mudanças arquiteturais na plataforma ao desenvolvimento de métricas para medir sua saúde. Esses desafios podem ser articulados pela definição das propriedades gerais dos objetos de estudo de cada nível (Seção 3.3.2). Além do escopo, as diferentes *dimensões* que se sobrepõem aos níveis do ECOS devem ser consideradas para representar seus três pilares: *software, transações e relacionamentos*. Em outras palavras, muitas organizações atuam no mercado por meio da combinação de processos de ES, visando misturar modelos de negócio, envolvimento de desenvolvedores externos e arquitetura do produto aberta. Esses pilares foram discutidos em duas revisões *ad hoc* da literatura (SANTOS & WERNER, 2011b; 2011d; 2012a; HANSSEN & DYBA, 2012) e em dois estudos secundários (MANIKAS & HANSEN, 2013; BARBOSA *et al.*, 2013). Como primeira iniciativa, SANTOS & WERNER (2011b) realizam uma coleção, análise e classificação de conceitos extraídos das 16 publicações do IWSECO 2009 e IWSECO 2010, inspirado pela visão dos ECOS em três dimensões, enunciada por CAMPBELL & AHMED (2010). A seguir,

apresenta-se a classificação, estendida pela inclusão das 13 publicações do IWSECO 2011 e IWSECO 2012:

- *Arquitetura do ECOS*: 11 publicações exploram questões relativas ao processo decisório na manutenção de propriedades arquiteturais da plataforma, e.g., utilização de visualização de modelos e código, ou modelagem de elementos do ECOS (ALSPAUGH *et al.*, 2009; ANVAARI & JANSEN, 2010; BOSCH, 2010; CATALDO & HERBSLEB, 2010; PETTERSSON *et al.*, 2010; YU & DENG, 2011; MOLDER *et al.*, 2011; SANTOS & WERNER, 2011b; IYER, 2012; SALMINEN & MIKKONEN, 2012; HYRYNSALMI *et al.*, 2012);
- *Estratégias e Táticas do ECOS*: 14 publicações exploram a análise de perspectivas dos ECOSs, fazem analogias com outros tipos de ecossistemas ou proveem métodos e modelos para organizar, classificar e avaliar ECOSs e ferramentas para desenvolver software neste contexto (JANSEN *et al.*, 2009; WERNER, 2009; CAMPBELL & AHMED, 2010; DHUNGANA *et al.*, 2010; HUNINK *et al.*, 2010; MCGREGOR, 2010; SANTOS & WERNER, 2010a; BERK *et al.*, 2010; ANGEREN *et al.*, 2011; BARBOSA & ALVES, 2011; POPP, 2011; HANSSEN & DYBA, 2012; POPP, 2012; JANSEN & CUSUMANO, 2012);
- *Redes Sociais do ECOS*: 4 publicações exploram redes focadas nos envolvidos (e.g., gerência da comunidade do ECOS) e nos artefatos (e.g., gestão de conhecimento em requisitos), bem como as interações entre si (FRICKER, 2009; CAPURUÇO & CAPRETZ, 2010; SEICHTER *et al.*, 2010; MENS & GOEMINNE, 2011).

Partindo de desafios selecionados da Seção 3.4, como “por que os ECOSs aparecem e desaparecem?” e “como definir e monitorar escopo, tipos, papéis e características dos ECOSs?”, a pesquisa envolvida nesta proposta de tese focou *a priori* na compreensão da análise de ECOSs com base em conceitos de ES. Ou seja, frente às divergências conceituais, o primeiro passo deste trabalho foi reunir as iniciativas da comunidade de pesquisa e prática em ECOSs a fim de organizar uma base de conhecimento inicial para a definição e modelagem de ECOSs (SANTOS & WERNER, 2010a). Como exemplo de *framework* de reutilização, *Reuse in Software Ecosystem* (*ReuseECOS*) consiste em um *framework* “3+1” de E&G de ECOSs (Figura 4.1), desenvolvido a partir das dimensões *técnica*, *transacional* e *social*, integradas pela

quarta dimensão de *E&G*, que corresponde à solução desta proposta de tese. Buscou-se compreender os ECOSs construídos a partir de diferentes RPSs nas fases do seu ciclo de vida e considerou os níveis de escopo para identificar novos ECOSs.

Além disso, isso serviu como instrumento para criar uma proposta de ECOSs baseada em reutilização, i.e., adaptar e/ou aplicar em ECOSs os conceitos relacionados a domínios e a componentes. Assim, conforme discutido na Seção 1.5, a abordagem *ReuseSEEM* seria contemplada em suas três primeiras fases: definição, modelagem e análise de ECOSs. *ReuseECOS* se estrutura como um processo composto por *passos* e *atividades*, para extrair diagnóstico de um ou mais ECOSs (*gerenciamento*) e interferir em seus elementos (*engenharia*), detalhado nas subseções a seguir. O *framework* foi utilizado para analisar casos de ECOSs do LENS/Reuse (Tabela 4.1), cujos exemplos são mostrados durante a sua apresentação.

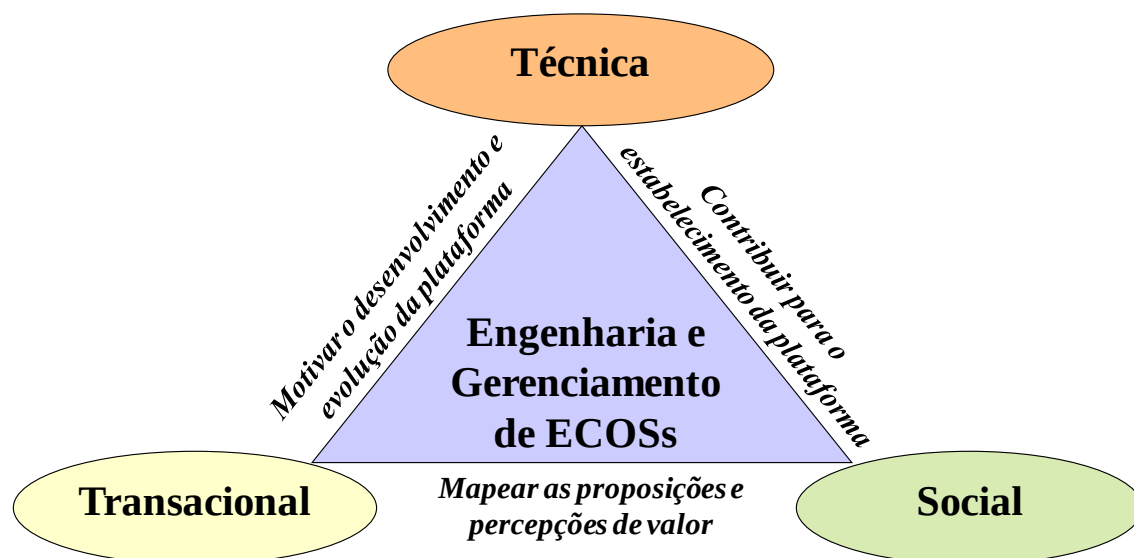


Figura 4.1 – Framework “3+1” para E&G de ECOSs. Fonte: (WERNER *et al.*, 2012)

4.2.1. Dimensão #1: Técnica

A primeira dimensão do *ReuseECOS* foca no elemento *plataforma*, i.e., base delimitada por um mercado, tecnologia, infraestrutura ou organização, referindo-se aos níveis de escopo *organizacional* e da *visão interna do ECOS* (JANSEN *et al.*, 2009a). A meta é entender como a ES pode ser aplicada à concepção, desenvolvimento e manutenção da plataforma de um ECOS a partir de produtos e serviços tradicionalmente construídos por projetos e processos, como sumariza a Figura 4.2. Primeiramente, a plataforma de um ECOSs deve ser selecionada para contextualizar a sua análise e projeto, i.e., o *caso de ECOS*, o que envolve entender a sua saúde a partir de indicadores

como produtividade, robustez e criação de nicho (HARTIGH *et al.*, 2006). Em seguida, parte-se para a análise e planejamento do processo de abertura da plataforma do ECOS, de acordo com seu modelo de negócio e/ou estilo arquitetural. Considera-se ainda elementos e fatores relativos a incerteza, complexidade e percepção de atividade, o que torna possível intervir no ECOS com princípios de Engenharia ao balancear modularidade e transparência para gerir visualização de informação durante a evolução e manutenção do ECOS, o que se chama *translucidez* (Seção 3.2).

Tabela 4.1 – ECOSs *free and open source software* (FOSS) do LENS/Reuse da COPPE/UFRJ

PROJETO	DESCRIÇÃO
ODYSSEY http://reuse.cos.ufrj.br/odyssey (WERNER <i>et al.</i>, 2000) (TEIXEIRA <i>et al.</i>, 2009)	Grande projeto na plataforma <i>Java Standard Edition</i> (JSE), desenvolvido desde 1997, corresponde a um IDE <i>stand-alone</i> acadêmico para apoiar engenharia de domínio e reutilização de software. Apresenta um <i>kernel</i> (Odyssey-Light), além de diversos <i>plug-ins</i> para apoiar subprojetos como tratar linhas de produtos de software (LPSs) e linhas de processos de software. Possui contribuições de vários grupos de ES do Brasil, entre eles, IFF, PUC-RS, UFJF, UNIRIO e UFF.
BRECHÓ http://reuse.cos.ufrj.br/brecho (WERNER <i>et al.</i>, 2007) (SANTOS <i>et al.</i>, 2010b)	Médio projeto na plataforma <i>Java Enterprise Edition</i> (JEE), desenvolvido desde 2005, corresponde a uma biblioteca <i>Web</i> de componentes e serviços para apoiar processos de gerência de reutilização e um mercado de componentes baseado em valor. Utiliza <i>frameworks open source</i> (<i>Struts</i> e <i>Hibernate</i>) e é uma plataforma acadêmica autocontida. Possui contribuições do grupo de ES da UFF.
PORTAL EDUES http://lab3d.coppe.ufrj.br/portaledues (SANTO <i>et al.</i>, 2009) (BORGES <i>et al.</i>, 2011)	Médio projeto na plataforma JEE desenvolvido desde 2009, corresponde a um ambiente <i>Web</i> para apoiar a pesquisa experimental (i.e., revisão sistemática, pesquisa de opinião e corpo de conhecimento). Utiliza um <i>framework open source</i> (<i>JBoss SEAM</i>) e apresenta-se como uma plataforma acadêmica autocontida. Possui contribuições do grupo de ES da UFLA.
PORTAL RPP http://www.rpp.ufrj.br (OLIVEIRA <i>et al.</i>, 2011) (SANTOS <i>et al.</i>, 2013a)	Médio projeto na plataforma JEE, desenvolvido desde 2010 em conjunto com o IPPUR/UFRJ²⁷ e com uma organização de software de Niterói/RJ, corresponde a um repositório <i>Web</i> para compartilhar e evoluir conteúdos de pesquisa de dez laboratórios sobre o tema políticas públicas, envolvendo usuários finais de diferentes áreas. Utiliza tecnologias <i>open source</i> (e.g., <i>HSQLDB</i>, <i>JSF2</i>, <i>Richfaces 4</i> e <i>EJB 3.1</i>) e vai entrar em produção futuramente.

A dimensão técnica visa também analisar os ECOSs por meio de analogias com outros tipos de ecossistemas, como biológicos, sociais e de negócio, para obter e derivar métodos e modelos de organização, classificação e avaliação dos ECOSs e suas plataformas (CAMPBELL & AHMED, 2010). Isso significa realizar a *engenharia de domínio* (ED), onde o domínio são os ECOSs, a fim de melhorar a compreensão do

²⁷ Instituto de Planejamento Urbano e Regional da UFRJ. Disponível em: <http://www.ippur.ufrj.br/>.

ciclo de vida da plataforma de um ECOS, gerenciar as similaridades e variabilidades de suas características e selecionar o seu estilo arquitetural. Conforme BARBOSA *et al.* (2013), esta dimensão trata de conceitos que aparecem diretamente em quatro tópicos de pesquisa investigados em ECOSs: arquitetura de software, sistemas operacionais, evolução de software e LPSs. Para atender a este propósito, os seguintes passos e atividades são considerados pela dimensão técnica:

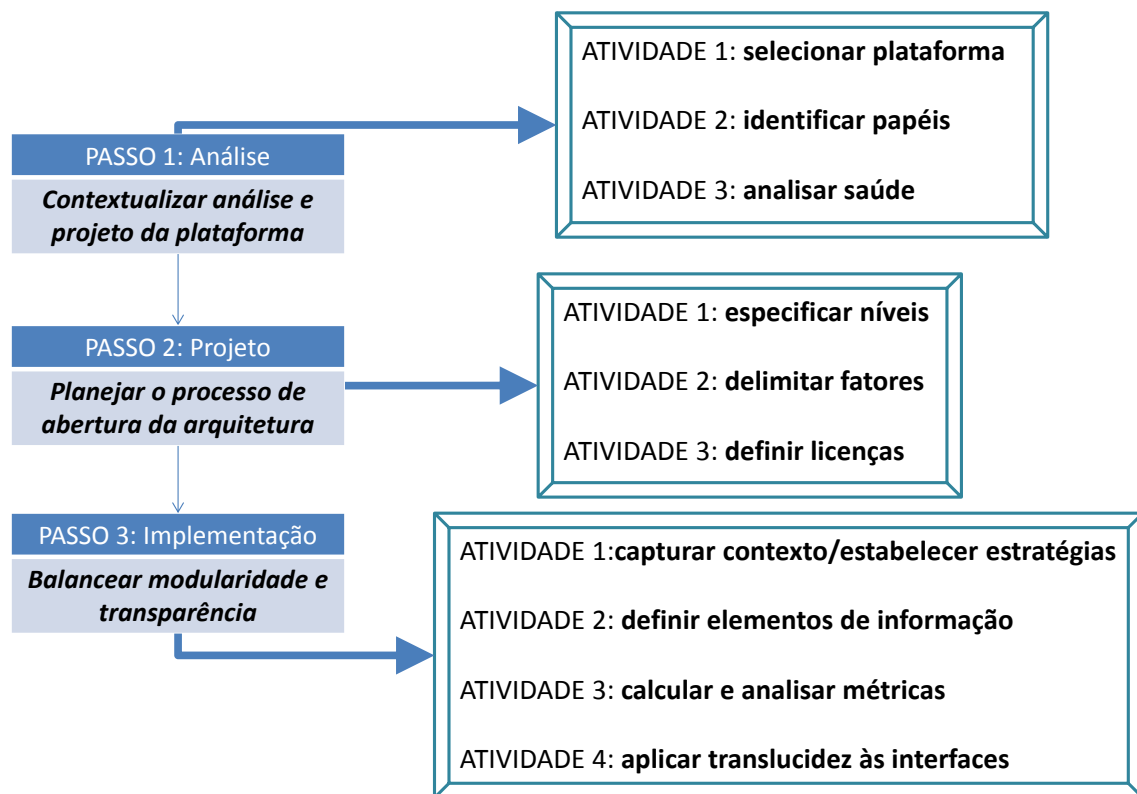


Figura 4.2 – Dimensão técnica do ReuseECOS. Fonte: (SANTOS & WERNER, 2011b)

Passo 1: *contextualizar análise e projeto da plataforma* – corresponde à compreensão da plataforma do ECOS por três atividades, que envolvem conceitos do modelo de domínio resultantes da especialização de ecossistemas de negócio (BERK *et al.*, 2010), combinada ao modelo de referência baseado no SPEM²⁸ (PETTERSSON *et al.*, 2010):

Atividade 1: *selecionar plataforma* – representa o momento da escolha da plataforma a ser estudada, dependendo da fronteira do ECOS. Esta seleção se dá pela caracterização dos elementos do caso do ECOS. No ECOS LENS/Reuse, os projetos Odyssey e Brechó foram selecionados para uma análise preliminar como ECOSs distintos, com suas respectivas plataformas Odyssey e Brechó. Os exemplos também mencionam o ECOS Portal EduES e o ECOS Portal RPP.

Atividade 2: *identificar papéis* – visa definir quem são os atores do ECOS, considerando os diferentes papéis e responsabilidades previamente tratados pela

²⁸Software Process Engineering Meta-model visa definir elementos para modelar um nível de detalhe relativo a processos de ciclo de vida e seus componentes, além de atividades e tarefas. Possui um perfil UML que agrega semântica ao modelo definido, além de uma notação customizada (OMG, 2008).

literatura de ecossistemas de negócio, pois os ECOSs podem ser tratados como uma especialização deles (IANSITI & LEVIEN, 2004a). Estes papéis são agrupados em duas categorias, *centralizador* e *agentes de nicho* (JANSEN *et al.*, 2009a; BERK *et al.*, 2010) (Seção 3.3):

- Os centralizadores representam as organizações que dão forma ao ECOS e são responsáveis por sua manutenção e evolução:
 - *organizações chave* são responsáveis por criar e compartilhar valor com todos os atores do ECOS, e.g., estudantes de graduação, mestrado e doutorado que trabalham nas plataformas do LENS/Reuse;
 - *organizações competidoras* são responsáveis por extrair valor e podem assimilá-lo ou eliminá-lo do ECOS, e.g., o ECOS Eclipse está atraindo os estudantes para que desenvolvam suas pesquisas fora dos ECOSs Odyssey e Brechó.
- Os agentes de nicho utilizam a plataforma para criar valor para ela e para si mesmos, ao desenvolver e melhorar as suas capacidades, mas almejam se diferenciar enquanto participantes do ECOS diante das organizações chave e competidoras:
 - *influenciador (influencer)*, que interfere na organização chave, e.g., Grupos de ES Experimental e de Qualidade de Software da COPPE/UFRJ e usuários finais de diferentes regiões do Brasil, que reportam problemas e fazem sugestões para as plataformas;
 - *atuante (hedger)*, que participa também das plataformas de organizações competidoras, a fim de minimizar os riscos, e.g., grupos de ES de outras universidades brasileiras como PUC-RS, UFF e IFF, que trabalham na plataforma Odyssey e na plataforma Eclipse, ou a organização de software de Niterói que trabalhou na implementação da Portal RPP por dois anos;
 - *discípulo (disciple)*, que participam somente da plataforma da organização chave, e.g., o grupo de ES da UFLA por cerca de dois anos se dedicou ao Portal EduES, devido à aprovação de um projeto de pesquisa nacional.
- Os agentes do nicho podem ser classificados como *autores*, *usuários* ou *especialistas no domínio* dos elementos da plataforma do ECOS, e podem ser responsáveis por atividades como *criar*, *reutilizar* e *implantar* (PETTERSSON *et al.*, 2010). Por exemplo, o Grupo de ES da UFLA possuía um estudante de graduação atuando como autor responsável por criar e implantar módulos no Portal EduES em 2011;
- Existem ainda os *usuários finais*, que se beneficiam de produtos e serviços oferecidos pela plataforma na forma de aplicações para a sociedade enquanto consumidora da indústria de software.

Atividade 3: analisar saúde – consiste em quantificar e qualificar indicadores de saúde do ECOS a partir de sua plataforma. Os três principais indicadores explorados pela literatura são (HARTIGH *et al.*, 2006; JANSEN *et al.*, 2009a;

DHUNGANA *et al.*, 2010, MCGREGOR, 2010; BARBOSA & ALVES, 2011; JANSEN & CUSUMANO, 2012):

- *produtividade*: descreve o nível de atividade do ECOS, i.e., volume de transações criadas (comerciais ou não), de valor agregado e de novos atores atraídos. Por exemplo, em maio de 2008, o ECOS Brechó atingiu um marco especial com um período interno de intenso desenvolvimento em sua plataforma, conforme relatório da ferramenta StatSVN²⁹ (Figura 4.3). Outras medidas interessantes, neste caso, podem ser pontos de caso de uso desenvolvidos, taxa de produção de componentes e serviços, número de relatórios gerados, qualidade das teses, dissertações e projetos de graduação etc.

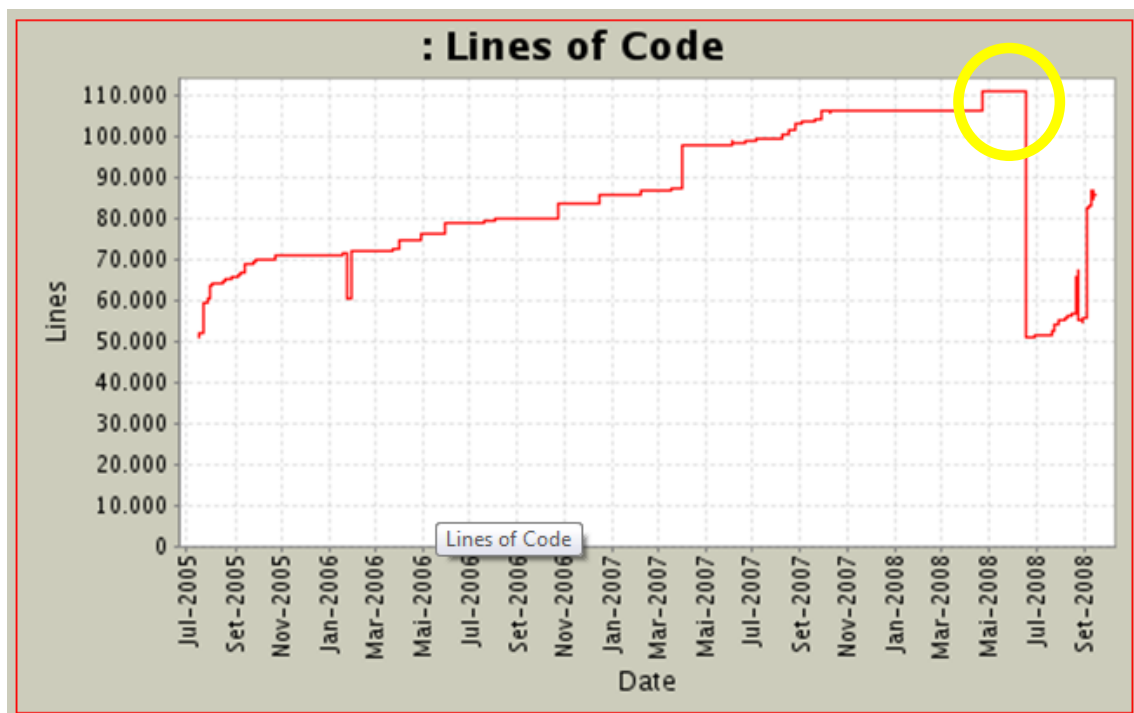


Figura 4.3 – Número de linhas de código (LoC) em *commits* SVN na plataforma do ECOS Brechó, extraído da ferramenta StatSVN. Fonte: (SANTOS & WERNER, 2011b)

- *robustez*: descreve quão bem um ECOS pode se recuperar após um forte estresse, e.g., a perda de participantes que compõem organização chave ou de agentes de nicho em lote, e avanços tecnológicos que afetam grande parte do ECOS, i.e., sua comunidade ou plataforma. Por exemplo, a saída ou perda de pesquisadores em nível de mestrado ou doutorado devido a oportunidades na indústria de software ou concursos públicos reduz o número de membros do LENS/Reuse que programam ou lideram novos projetos de *plug-ins* da plataforma Odyssey e extensões da plataforma Brechó. Outras medidas interessantes, neste caso, podem ser qualidade da pesquisa

²⁹StatSVN recupera informação a partir de um repositório SVN e gera várias tabelas e gráficos que descrevem o desenvolvimento do projeto. Disponível em: <<http://www.statsvn.org>>.

(impacto das publicações) e qualidade dos produtos gerados sobre as plataformas de acordo com normas como a ISO 25000³⁰.

- *criação de nicho*: descreve a capacidade do ECOS em criar oportunidades para que atores novos e antigos explorem novas possibilidades comerciais ou não. Por exemplo, a busca contínua por agências nacionais e regionais de apoio financeiro a projetos (e.g., CNPq, CAPES, FAPERJ e FAPEMIG) visando reunir grupos de pesquisa, como a UFLA e o IPPUR/UFRJ, e fortalecer novos ECOS construídos dentro do ECOS LENS/Reuse, no caso, o Portal EduES e a Portal RPP, respectivamente. Outro exemplo é a divulgação utilizada para promover os ECOS Brechó e Portal EduES por meio de minicursos e apresentação de artigos em eventos nacionais e internacionais. Outra medida interessante pode ser o número de novos atores como agentes de nicho autores e usuários da plataforma.

Passo 2: planejar o processo de abertura da arquitetura – corresponde à modelagem da abertura da plataforma do ECOS por meio de três atividades:

Atividade 1: especificar níveis – tem por objetivo identificar e separar módulos ou componentes da plataforma do ECOS, explorando os benefícios de uma arquitetura em camadas e facilitando o seu processo de abertura ao organizá-lo em tarefas e subtarefas, para que cada ator aja em um nível de abstração particular (ANVAARI & JANSEN, 2010). Por exemplo, tanto as plataformas dos ECOSs Brechó e Portal EduES (*Web*) quanto a do ECOS Odyssey (*Desktop*) são do tipo infraestrutura e apresentam arquitetura em (três) camadas:

- *aplicações estendidas*: construídas pelos desenvolvedores externos ao LENS/Reuse e instaladas pelos usuários das soluções de software;
- *aplicações nativas*: desenvolvidas pelo LENS/Reuse e, em alguns casos, não podem ser modificadas;
- *kernel*: desenvolvido pelos membros do LENS/Reuse, corresponde ao núcleo da infraestrutura, que trata de componentes e serviços de baixo nível como *drivers* de dispositivo, *segurança*, *frameworks* etc.

Atividade 2: delimitar fatores – tem por objetivo definir mecanismos de extensão e acessibilidade ao explicitar as condições que regem o acesso às diferentes camadas e componentes da plataforma (ANVAARI & JANSEN, 2010). Por exemplo, no ECOS Brechó, três fatores são utilizados para tornar clara a noção de abertura da arquitetura da plataforma:

- *integrar*: permite utilizar componentes de uma camada existente em uma aplicação de software externa via API, chamada de serviço, inclusão de código fonte, objeto de dados compartilhados ou outros mecanismos de extensão de software. Por exemplo, o mecanismo de integração da ferramenta Visual Studio TFS³¹ para publicação de

³⁰ISO 25000 é uma norma para ES – *Software product Quality Requirements and Evaluation*. Disponível em: <http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=35683>.

³¹O *Visual Studio Team Foundation Server* (TFS) é a plataforma de colaboração na essência da solução de gerenciamento do ciclo de vida de aplicativos da Microsoft. O TFS oferece suporte para práticas de desenvolvimento ágil para vários IDEs e plataformas localmente ou na nuvem e fornece as ferramentas

liberações de versões de componentes diretamente na plataforma Brechó e o módulo para extração e gerência de serviços Web (MARINHO *et al.*, 2009a; 2009b);

- *estender*: permite ampliar a funcionalidade dos componentes de uma camada, e.g., a evolução da plataforma para apoiar um mercado de componentes, originando a sua nova versão, chamada Brechó-VCM (SANTOS & WERNER, 2009);
- *modificar*: permite substituir ou alterar os componentes de uma camada, e.g., modificação do módulo de tarifação de componentes (MARINHO *et al.*, 2009c) para incorporar modelos de precificação (SANTOS *et al.*, 2009b).

Tabela 4.2 – Comparação entre estratégias de abertura das plataformas dos ecossistemas do ECOS LENS/Reuse, com base em (ANVAARI & JANSEN, 2010), onde: *P* = possibilidade, *L* = status da licença, *Po* = possível, *Pc* = possível para alguns componentes, *Np* = não permitido, *Pn* = permissão não é necessária, *Pa* = permissão é necessária em alguns casos, e *Ps* = permissão é sempre necessária. Fonte: (SANTOS & WERNER, 2011b)

FATOR	NÍVEL	Odyssey (1997-...)		Brechó (2005-...)		EduES (2009-...)		RPP (2011-...)	
		P	L	P	L	P	L	P	L
Integrar	aplicações estendidas	Po	Pn	Po	Pn	Po	Pa	Po	Pa
Estender		Po	Pn	Po	Pa	Po	Pa	Po	Pa
Modificar		Po	Pa	Po	Pa	Po	Pa	Po	Pa
Integrar	aplicações nativas	Po	Pa	Po	Pa	Po	Pa	Po	Pa
Estender		Po	Pa	Po	Pa	Po	Pa	Po	Pa
Modificar		Po	Pa	Po	Pa	Po	Pa	Po	Pa
Integrar	kernel	Pc	Ps	Pc	Ps	Po	Ps	Po	Ps
Estender		Pc	Ps	Pc	Ps	Po	Ps	Po	Ps
Modificar		Np	Ps	Pc	Ps	Po	Ps	Po	Ps

As relações entre níveis e fatores relativos à abertura das plataformas dos ECOSs do LENS/Reuse são resumidas na Tabela 4.2. Algumas observações: (i) a plataforma Brechó foi alvo de estudo técnico de viabilidade de reengenharia de sua versão Brechó-VCM, visando extrair as funcionalidades relativas ao mercado de componentes para gerar um *kernel* leve e voltado para atividades mínimas do processo de gerência de reutilização; (ii) a plataforma Odyssey passou efetivamente por este processo há quase 10 anos; (iii) a plataforma Portal EduES está em uma fase favorável para começar a pensar sobre isso, após duas liberações de versões; e (iv) o Portal RPP, por ser um sistema real em produção com vários usuários, deve se beneficiar da experiência de 15 anos do ECOS LENS/Reuse para ser estruturado em sua evolução evoluído. Nota-se que a possibilidade de participação de atores externos sobre as plataformas vem aumentando, ao passo que a aplicação de licenças vem se tornando mais controlada, sobretudo por um processo de gerência de configuração definido há mais de 10 anos.

Atividade 3: definir licenças – tem por objetivo facilitar e ao mesmo tempo restringir a atuação dos diversos atores do ECOS sobre a plataforma por meio da definição de direitos e deveres em seu processo de abertura (ALSPAUGH *et al.*, 2009). São consideradas questões como evolução e substituição de necessidades, componentes e serviços; evolução da arquitetura; e evolução da licença por mudanças nos direitos desejados ou deveres aceitáveis. Isso resulta na chamada “*co-evolução*” da plataforma, por mostrar que existe uma interdependência entre organizações chave, que produzem *para* a plataforma, e agentes de nicho, que produzem *com* a plataforma. Mecanismos de coordenação e de comunicação são requeridos para este cenário. A concepção das licenças deve levar em conta alguns tipos de elementos de software comuns às arquiteturas de quaisquer sistemas, e.g., componentes de código fonte, executáveis, serviços, APIs, conectores de software, métodos de conexão e arquiteturas configuradas de sistemas e subsistemas. Por exemplo, no ECOS Brechó, a permissão da LENS/Reuse (organização chave) é sempre necessária para integrar, estender ou modificar elementos de software do *kernel* da plataforma (Tabela 4.2). Nesse contexto, JANSEN & CUSUMANO (2012) identificam modelos de associação para *participantes* de ECOSs para reger as políticas de licença, que estão um nível acima dos agentes de nicho:

- *modelo de parceria (partnership)*: é de propriedade da organização chave e visa sustentar, gerenciar, agrupar e expandir o ecossistema dos parceiros e, conseqüentemente, o número de agentes do nicho. Oferece a oportunidade de maior engajamento de um ator do ECOS junto à organização chave ao preencher um ou mais papéis disponíveis, com benefício pré-definidos. Usualmente, o parceiro deve atender a um conjunto de requisitos e pagar taxas de associação;
- *modelo de sociedade (membership)*, normalmente oferecido por organizações chave em ECOSs FOSS devido à sua formação legal diferenciada, permite a criação de um ecossistema de membros com ideais de cooperação e colaboração, podendo envolver definição de responsabilidades e eventuais papéis sujeitos a taxas.

Passo 3: balancear modularidade e transparência – corresponde à implementação da abertura da plataforma do ECOS com base na transparência e no projeto modular de sistema (Seção 3.3), por meio de quatro atividades:

Atividade 1: capturar contexto/estabelecer estratégias – visa detalhar o escopo da plataforma do ECOS, classificando-o de acordo com o *nível de abstração*³² e o *tipo*³³ de conhecimento manipulado na rede de produção de software, bem como o perfil do *ator*³⁴. Com a arquitetura em camadas (*passo 2*), torna-se mais fácil aplicar o conceito de DBC ao utilizar interfaces. Por exemplo, a plataforma do ECOS Brechó consiste em uma aplicação de software Web desenvolvida sobre a plataforma do ECOS Java EE desde 2005, cuja arquitetura de software segue o padrão MVC (*Model-View-Controller*) e se baseia em *frameworks* pertencentes a ECOS FOSS conhecidos (*Struts* e *Hibernate*). A plataforma

³²Por exemplo, artefato em nível de requisito, modelo, código ou executável, recurso, informação.

³³Por exemplo, funcionalidade, componentes, interesse transversal.

³⁴Por exemplo, gerente da plataforma, engenheiro de requisitos, analista, desenvolvedor interno (membro/parceiro) ou externo.

utiliza ainda o *container* de aplicações Tomcat (ECOS Apache) e o servidor de banco de dados MySQL (ECOS Oracle). O sistema de controle de versões da plataforma Brechó é a plataforma do ECOS Subversion (SVN) e atualmente conta com 18 desenvolvedores do ECOS LENS/Reuse cadastrados (13, em 2008). A plataforma apresenta, em sua maioria, artefatos em nível de código, é baseada em componentes e tem quatro destes participantes mais ativos, um por 6 anos (gerente, doutorando), outro que terminou uma contribuição após um ano (estudante de engenharia), um terceiro recém-incorporado (mestrando), além de um agente de nicho discípulo, que estudou remotamente a estrutura da plataforma há dois anos (estudante de computação). A plataforma era acessada por duas localizações geográficas diferentes, Rio de Janeiro/RJ e Lavras/MG. Devido à carência de documentação do tipo industrial e à volatilidade dos participantes, uma estratégia vantajosa desde o começo do projeto foi o uso de *javadoc* e padrões de programação, bem como tecnologias livres, embora não haja recurso humano/financeiro para viabilizar qualquer migração ou atualização de elementos arquiteturais. Por exemplo, a manutenção dos *frameworks Struts* e *Hibernate* da versão baseada em *tags*, ao invés da versão baseada em anotações.

Atividade 2: definir elementos de informação – visa explicitar três elementos chave que constituem a arquitetura da plataforma, os dois primeiros relativos às *interfaces* da plataforma (que devem ser translúcidas) e o último relativo à *visibilidade* da evolução do trabalho de modelar, projetar e implementar um componente da plataforma, conforme CATALDO & HERBSLEB (2010):

- *incerteza*: é a probabilidade de mudanças ocorrerem na interfaces da plataforma, analisando as decisões a serem tomadas no *tempo* (e.g., evolução e correção) e no *espaço* (e.g., desenvolvimento distribuído). A meta de tratar este elemento é estabilizar as interfaces, o que impacta a rastreabilidade entre os diferentes níveis e tipos de abstração dos componentes. Incluem-se questões de segurança e confiabilidade, interface gráfica com usuário e o *trade-off* de inclusão de funcionalidade *versus* “emagrecimento”³⁵ da plataforma (BOSCH, 2010). Por exemplo, o ECOS Odyssey possuiu agentes de nicho desenvolvendo componentes para a camada do *kernel* e para a camada das aplicações nativas em diferentes localizações geográficas (Juiz de Fora/MG, Campos/RJ e Porto Alegre/RS). As atividades precisaram ser coordenadas por uma lista de discussão e de *workshops* presenciais para orquestrar as decisões a serem tomadas ao longo do tempo;
- *complexidade*: é a propriedade de uma plataforma possuir interfaces padronizadas e compreensíveis, explorando o princípio do encapsulamento para favorecer as atividades dos agentes de nicho nos diferentes níveis de abstração da plataforma. A meta de tratar este elemento é padronizar estilos de modelagem e de programação, o que impacta características do produto, como manutenibilidade e reutilizabilidade, e.g., custo e esforço para evoluir a plataforma. Isso também contribui para manter a variabilidade das características da plataforma (CAMPBELL & AHMED, 2010). Por exemplo, o código

³⁵Refatoração para remover e reorganizar funcionalidades da plataforma por sua erosão (i.e., perda da vantagem competitiva) devido ao excesso de funcionalidades na camada do *kernel* (BOSCH, 2010).

fonte da plataforma do ECOS Brechó foi construído seguindo os padrões estabelecidos pelo relatório *Java Code Conventions*, desenvolvido pela Sun, além de seguir o padrão MVC e ser baseado em dois *frameworks* bem conhecidos, o que favorece a entrada de agentes de nicho e novos participantes na organização chave;

- *percepção de atividade*: é a capacidade dos atores terem consciência do processo que rege a sua atividade, bem como as dependências e entraves de dois aspectos, *artefatos* e *papéis*. A meta de tratar este elemento é gerir coordenação e comunicação, o que impacta a compreensibilidade do conhecimento envolvido na arquitetura, e.g., custo e esforço de construir e gerenciar a plataforma. Isso impacta no *workflow* para desenvolvimento de casos de uso da plataforma, bem como no acesso aos dados para criação de componentes e serviços por organizações externas (BOSCH, 2010). Por exemplo, os quatro ecossistemas do ECOS LENS/Reuse têm plataformas sob controle de versões (SVN) e possuem um sistema de controle de modificações implantado (*Bugzilla*) (WERNER *et al.*, 2010). Isso permite aos agentes de nicho de cada ECOS colaborar e se comunicarem no desenvolvimento e manutenção de componentes das arquiteturas, o que inclui listas de discussão dos ECOS Yahoo e Google.

Atividade 3: calcular e analisar métricas – visa extrair informações sobre a arquitetura da plataforma com base nos elementos de informação definidos. A Figura 4.4 exhibe algumas das informações discutidas abaixo:

- para medir o elemento *incerteza*, alguns dados coletados são as experiências de arquitetos e desenvolvedores no entendimento dos requisitos, das características da plataforma (mapa de conhecimento) e dos modelos que identificam pontos de incerteza na história de projetos similares. Por exemplo, na plataforma do ECOS Brechó, o tempo e o esforço gastos para construir um novo componente para mostrar informações na forma de gráficos de barras ou de pizza pode ser baseada em dados de LoC do SVN. Isso inclui dados de componentes desenvolvidos no escopo dos mecanismos de *marketing*, de avaliação e de negociação para a Brechó-VCM (SANTOS & WERNER, 2011c, SANTOS *et al.*, 2013c);
- para medir o elemento *complexidade*, alguns dados coletados são número e tipos de parâmetros das interfaces de componentes nas diferentes camadas da plataforma, das restrições descritas na *Object Constraint Language* (OCL) para a arquitetura e de propriedades transversais à plataforma. Por exemplo, no código fonte da plataforma do ECOS Brechó, pode-se consultar o *javadoc* para verificar se a legibilidade e a manutenibilidade foram afetadas ao longo do tempo, aplicando critérios de qualidade de produto sobre as interfaces dos componentes (SANTOS *et al.*, 2009a);
- para medir o elemento *percepção de atividade*, alguns dados coletados são políticas ou contratos que gerem os elos entre artefatos e papéis, e relatórios de ferramentas de apoio a decisões de projeto arquitetural que capturam, documentam e rastreiam a evolução da plataforma ao longo do tempo. Pode-se citar o uso de novo recurso,

bloqueio de partes do código, estabelecimento de pré e pós-condições etc. Por exemplo, ferramentas como StatSVN podem ser utilizadas para coletar e analisar dados sobre a plataforma do ECOS Brechó, e.g., linha de tempo para o código fonte, pacotes e arquivos distribuídos por mudança realizada, contribuição de cada desenvolvedor (históricos de *commits*), atividades por hora por dia ou por semana etc. Isso contribui para a definição de licenças (*passo 2*) para as interações entre organizações chave, competidoras e externas.

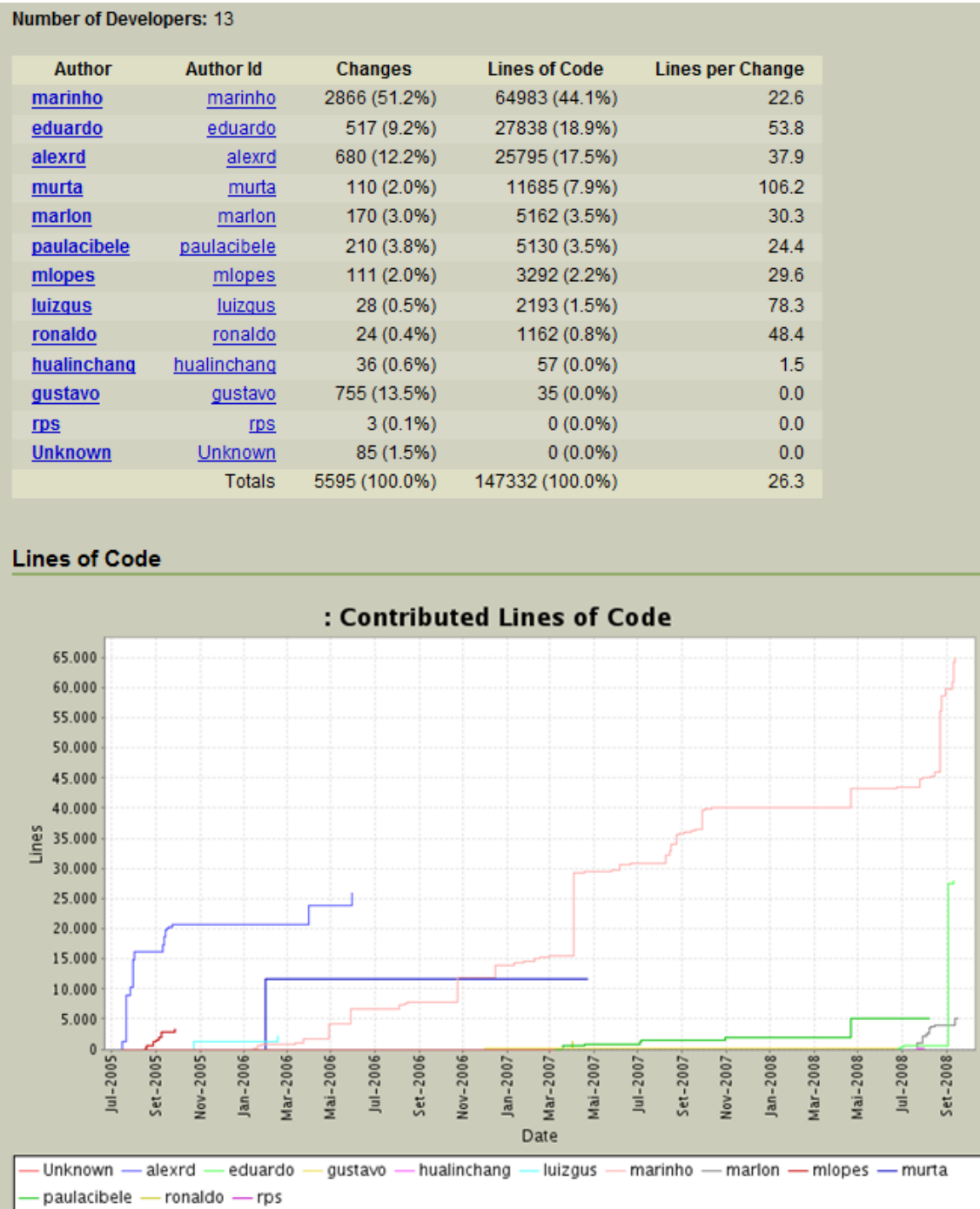


Figura 4.4 – LoC por mudança realizada (superior) e contribuição de LoC por desenvolvedor da plataforma do ECOS Brechó (inferior). Fonte: (SANTOS & WERNER, 2011b)

Atividade 4: *aplicar translucidez às interfaces* – visa contribuir para o mecanismo de coordenação e comunicação do ECOS, favorecendo a colaboração e a cooperação ao evitar a sobrecarga de informação, i.e., cada perfil tem acesso a um dado tipo e nível de abstração de conhecimento da plataforma, de acordo com o seu papel. Paralelamente, propriedade intelectual e segurança devem ser tratadas para agregar e manter a confiabilidade do conhecimento manipulado, seja nos modelos ou no código fonte. Por exemplo, a implementação da plataforma do ECOS Brechó é baseada em *javadoc* e utiliza SVN, o que permite executar mineração de dados para visualizar impactos de modificações em certos componentes da plataforma. Além disso, informação poderia ser filtrada e exibida para um determinado ator, considerando a sua tarefa e perfil (e.g., pacotes ou classes). Para CATALDO & HERBSLEB (2010), uma estratégia para isso seria explicitar elementos de informação por meio do uso de *tags* em descrições arquiteturais e de anotações em códigos baseados em *javadoc*, a fim de favorecer a translucidez das interfaces da plataforma. Assim, pode-se melhorar a visibilidade de elementos de informação/comportamentos, e ocultar outros, através da ligação entre papel técnico e sócio-organizacional em uma rede socio-técnica. A visualização de informação usa regras que relacionam conhecimentos de diferentes tipos e níveis de abstração com os perfis do ECOS e pode, então, apoiar o E&G. Entre as estratégias de mercado que devem ser analisadas para lidar com a aplicação da translucidez às plataformas, estão:

- *clopenness* (MOLDER *et al.*, 2011): estratégia relativa ao movimento contínuo entre os estados de fechamento (*closeness*) e de abertura (*openness*) da arquitetura da plataforma de um ECOS. Rompe a falsa dicotomia entre ECOS proprietário e FOSS e prima por ECOS híbridos (Seção 3.3.2). Isso requer uma análise do caso do ECOS para compreender que escala de *clopenness* é a mais adequada e manter a comunidade sustentável e diversificada (DHUNGANA *et al.*, 2010), aplicado nos elementos do ECOS ou nas conexões entre eles. Entre alguns atributos chave para essa avaliação, estão: (i) *oportunidades* para as organizações externas, como para os participantes do ECOS, (ii) *acessibilidade* ao ECOS para os diversos tipos de agentes de nicho, considerando o seu *status* no ECOS; (iii) *transparência*, como já explicado; (iv) *reciprocidade* baseada em políticas que regem as interações entre os atores do ECOS; e (v) *permissão* segundo níveis e fatores que impactam a arquitetura da plataforma (*passo 2*);
- *mashups* (SALMINEN & MIKKONEN, 2012): estratégia caracterizada por aplicações que combinam recursos (dados, código etc.) a partir de diferentes serviços disponíveis na *Web* permitindo uma experiência integrada. Pode-se combinar conteúdo de maneira nova e imprevisível, criando novos serviços *Web* ou provendo novas visualizações para aqueles existentes. Isso requer uma análise da organização chave sobre a estabilidade e usabilidade das aplicações para atrair e manter usuários finais e fornecer meios para que eles desenvolvam com ECOSs, na categoria programação de usuário / plataformas *Web* e *Mobile* (Seção 3.3.2) (BOSCH, 2009). Por exemplo, um *mashup* consiste em combinar o serviço de publicação

de imagens no perfil do site de rede social *Facebook* com a aplicação *GoogleMaps*, a fim de criar um componente para geração de álbuns;

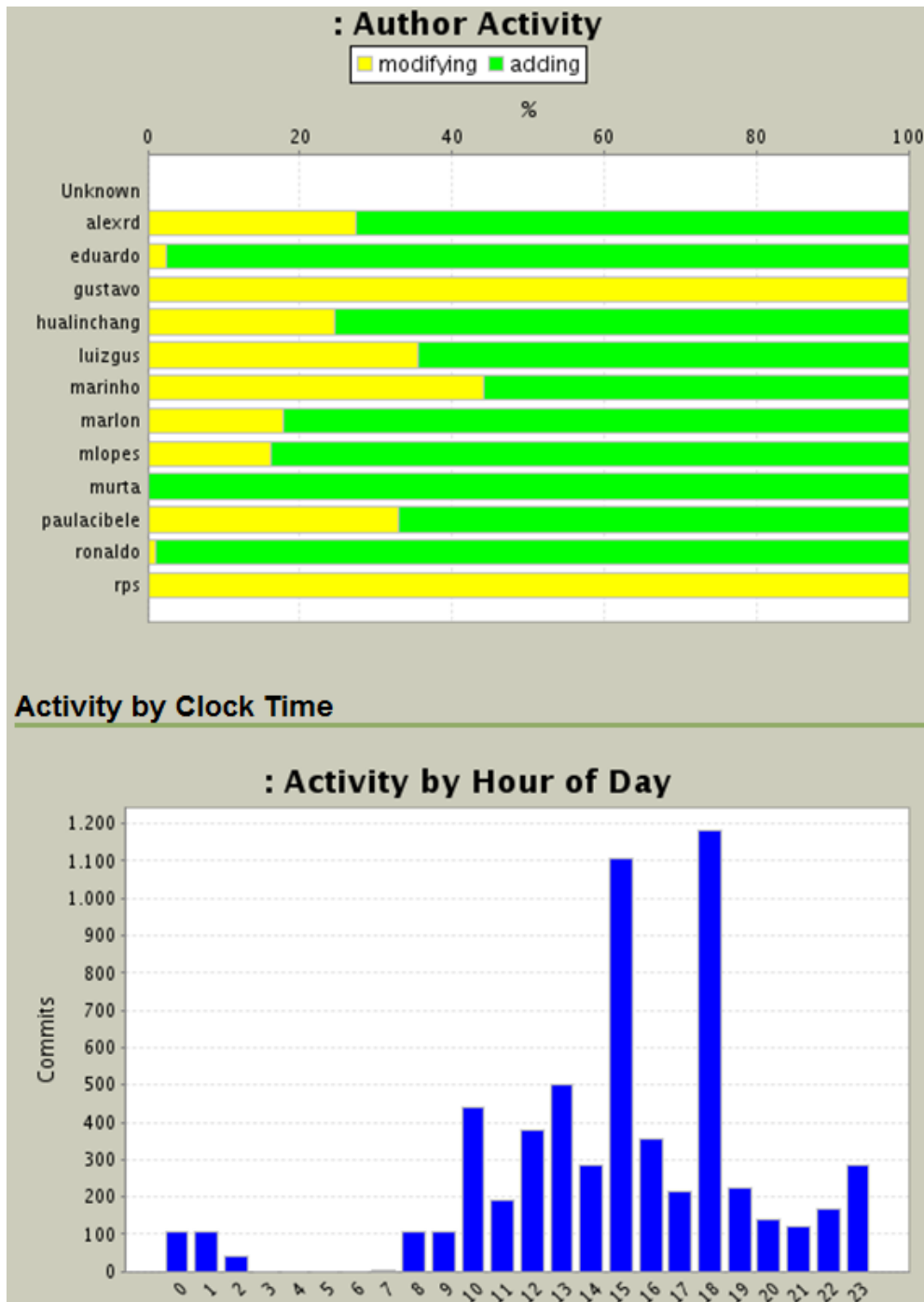


Figura 4.5 – Tipo de atividade (modificação/correção) (superior) e taxa de atividade por hora por dia, na plataforma do ECOS Brechó. Fonte: (SANTOS & WERNER, 2011b)

- *multi-homing* (HYRYNSALMI *et al.*, 2012): estratégia em que a organização externa publica aplicações, componentes e serviços em múltiplas plataformas de ECOS concorrentes, i.e., um desenvolvedor externo oferece aplicações para as plataformas Windows Phone e Android com base em seu produto de interesse. Isso requer análise da organização chave para gerenciar a estabilidade e usabilidade das

interfaces da plataforma e manter/ atrair organizações externas desconhecidas, além de oferecer oportunidades para agregar o valor do ECOS direta ou indiretamente. Por exemplo, no ECOS Eclipse, não existe obrigatoriedade de publicação de *plug-ins* no *marketplace* e a comunidade pode não ser totalmente mapeada.

4.2.2. Dimensão #2: Transacional

A segunda dimensão do *ReuseECOS* foca no elemento *conhecimento*, i.e., recursos, artefatos e informações, referindo-se aos níveis de escopo da *visão interna do ECOS* e da *visão externa do ECOS* (JANSEN *et al.*, 2009a), mais do que *organizacional*. A meta é entender como fatores naturais do processo de ES podem interferir em mecanismos auto regulatórios de um ECOS comercial ou não, como sumariza a Figura 4.6. Primeiramente, os elementos que compõem o contexto de concepção, gerenciamento e monitoramento do ECOS e de sua plataforma devem ser identificados/capturados e registrados. Pode-se aferir aspectos da saúde do ECOS pela aplicação da abordagem GQM (*Goal-Question-Metric*) (BASILI *et al.*, 1999): (i) *selecionar as metas do ECOS*, e.g., ser competitivo; (ii) *especificar questões que apoiem na verificação das metas*, e.g., “qual é o valor do ECOS?” e “como é o nicho de mercado?”; e (iii) *definir, coletar e analisar métricas relacionadas*, e.g., tipos e quantidades de atores, países e usuários finais envolvidos, taxa de percepção de benefícios da plataforma, índices de riscos e volume de oportunidades atingidos pelo ECOS etc. A abordagem GQM apoia a coleção, manipulação e apresentação de informação acerca da sustentabilidade e da diversidade como indicadores de saúde do ECOS (Seção 3.2). A partir do conjunto de indicadores, um *framework* para gerenciamento e monitoramento do ECOS pode ser modelado e instanciado com base na definição de cadeias de valor e de resultados (Seções 3.1 e 3.3.6, respectivamente).

A dimensão transacional visa também analisar os ECOSs por meio de analogias com outros tipos de ecossistemas, como biológicos e de negócio, para obter e derivar métodos e modelos de organização, classificação e avaliação da circulação de conhecimento e de valor *no ECOS* e *a partir do/para o ECOS* (CAMPBELL & AHMED, 2010). Isso significa realizar *transações nos ECOSs* ao criar metas e planos de ação para seus elementos por meio de programas e projetos, impulsionar inovação (fechada e aberta) junto à indústria de software e estabelecer planejamento estratégico para entender como, quando, onde e quem realizará essas metas. Conforme BARBOSA *et al.* (2013), esta dimensão trata de conceitos que aparecem diretamente em três tópicos

de pesquisa investigados em ECOSs: modelagem de ECOSs, co-inovação em software e negócios. Para atender a este propósito, os seguintes passos e atividades são considerados pela dimensão transacional:

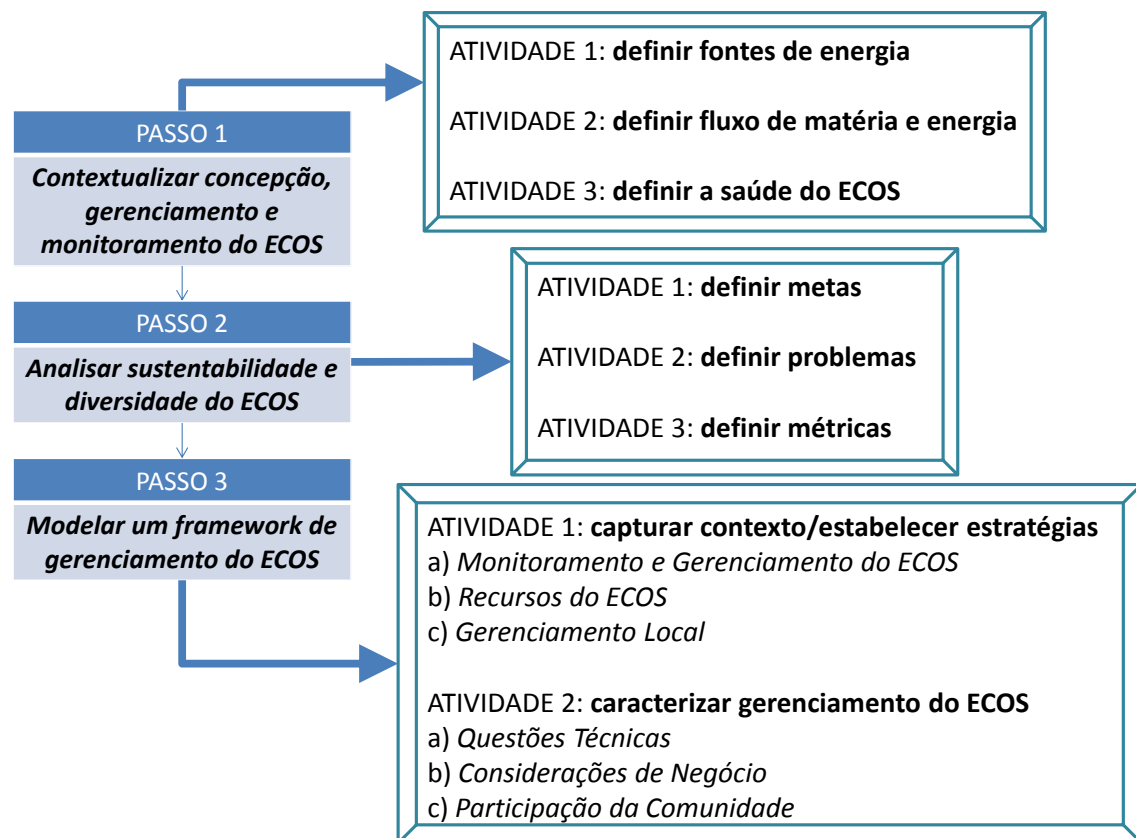


Figura 4.6 – Dimensão transacional do *ReuseECOS*. Fonte: (SANTOS & WERNER, 2011d)

Passo 1: *contextualizar concepção, gerenciamento e monitoramento do ECOS* – tem por objetivo mapear conceitos, características e comportamentos de outros ecossistemas para o contexto do ECOS e enriquecer as fases e as propriedades do seu ciclo de vida estabelecido na dimensão técnica (DHUNGANA *et al.*, 2010), por três atividades:

Atividade 1: *definir fontes de energia* – similar à constituição da flora nos ecossistemas biológicos que é responsável pela produção de energia primária pela fotossíntese, esta atividade visa identificar todos os *participantes fundamentais* da organização chave do ECOS. Por exemplo, no ECOS LENS/Reuse, alguns estudantes de doutorado são selecionados para gerenciar os ECOSs existentes, considerando a sua pesquisa e experiência (conhecimento), bem como afinidade e relacionamento (rede social);

Atividade 2: *definir fluxo de matéria e energia* – similar à constituição da cadeia alimentar em ecossistemas biológicos, esta atividade visa organizar estrutural e dinamicamente a *cadeia de valor* do ECOS. Visa também mapear as cadeias de resultados que considerem as diferentes suposições a serem investigadas nas decisões cotidianas (BIFFL *et al.*, 2006). Por exemplo, a proposta da nova versão da plataforma do ECOS Brechó, a Brechó-VCM, consiste em prover um mercado/ambiente para desenvolvimento de ECOSs competitivos/colaborativos. A plataforma define uma cadeia de valor (Figura

4.7) e uma cadeia de resultados (SANTOS, 2010), partindo de pesquisas na literatura e de avaliações com especialistas em Reutilização de Software. Pela imaturidade do mercado de componentes (WERNER *et al.*, 2009), uma estratégia para iniciar o ciclo de vida do novo ECOSs Brechó-VCM precisa ser definida, além da calibragem das cadeias mencionadas. Nesse sentido, é importante entender que existem dois tipos de cadeias de valor (MCGREGOR, 2010; YU & DENG, 2011):

- *cadeia aberta*, para ECOSs FOSS tradicionais, onde a organização chave fornece a plataforma, mas deixa nas mãos das organizações externas a responsabilidade de criar valor orientado pela demanda dos usuários, o que divulga e atrai novos atores. Normalmente, utiliza o conceito de *transferência* para os fluxos entre atores, i.e., sem taxas financeiras;

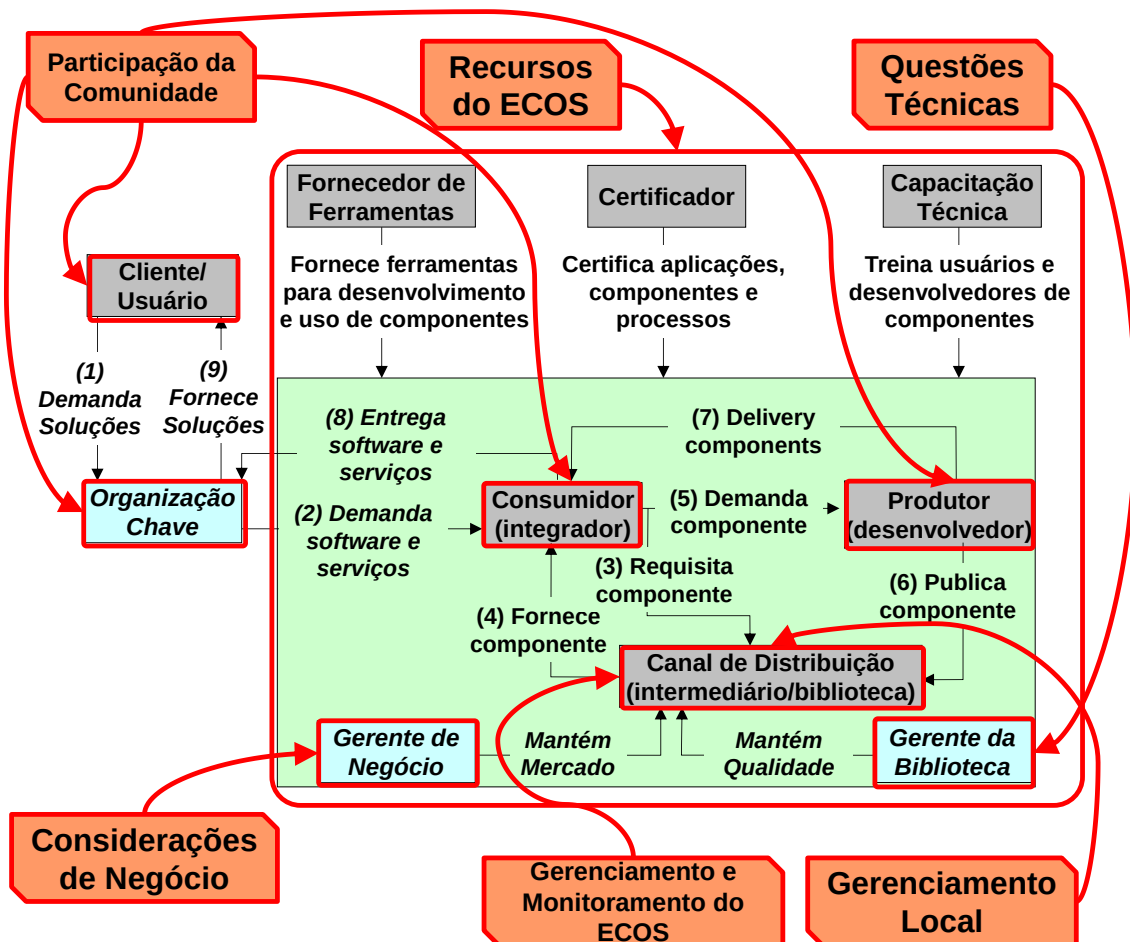


Figura 4.7 – Cadeia de valor da indústria de software (externa) e do mercado/ambiente de DBC (interno) implementada na Brechó-VCM, analisada e marcada com elementos de gerenciamento de ECOSs. Os retângulos representam os papéis dos agentes e as setas, as suas ações. Nos retângulos mais claros, estão os elementos que adicionados pela proposta de SANTOS (2010). Fonte: (SANTOS & WERNER, 2011d)

- *cadeia de valor clássica*, para ECOSs proprietários estritos, onde há um relacionamento linear entre a organização chave e os usuários finais, e entre as organizações chave e externas, embora em alguns modelos de negócio possa existir uma relação entre os usuários e os

fornecedores do ECOS. Na maior parte das vezes, utiliza o conceito de *transação* para os fluxos entre atores, i.e., comunicação padronizada, contabilizada e compensada de alguma forma.

Atividade 3: definir a saúde do ECOS – similar ao entendimento da reciclagem de nutrientes nos ecossistemas biológicos, esta atividade visa estabelecer uma *rede social* voltada para os artefatos, recursos e informações que circulam no ECOS, assim como os atores que os manipulam, considerando seus respectivos papéis. O enfoque deve estar no tratamento dos ciclos de conhecimento do ECOS. Por exemplo, o ECOS Brechó possui um fluxo de conhecimento com as seguintes etapas: (i) um novo participante do ECOS LENS/Reuse ou agente de nicho recebe treinamento; (ii) a seguir, este ator desenvolve um projeto piloto para conhecer a tecnologia e a organização chave, como meta inicial; (iii) o ator troca experiências com um participante fundamental mais experiente, para estabelecer uma nova meta e um plano de ação, e.g., desenvolvimento de novo módulo ou projeto de pesquisa na plataforma; (iv) o ator desenvolve suas atividades de modo a agregar valor e trocar conhecimento com o ECOS Brechó e com o ECOS LENS/Reuse; por fim, (v) o ator disponibiliza o conhecimento ao ECOS Brechó e o externaliza para outros ecossistemas internos e externos ao ECOS LENS/Reuse por meio de publicações.

Passo 2.1: analisar sustentabilidade do ECOS – tem por objetivo cuidar do crescimento e manutenção da comunidade formada pela organização chave e externas, e usuários do ECOS ao longo do tempo. A ideia é favorecer a sobrevivência da plataforma frente a mudanças naturais na indústria de software, e.g., evolução de tecnologia ou ataque/sabotagem do ECOS por organizações competidoras. O elemento crítico para este passo é a *organização chave*, cuja sustentabilidade pode requerer um dos modelos de governança para o ECOS (JANSEN & CUSUMANO, 2012):

- *plataforma de software dirigida pela comunidade*: grandes produtos FOSS, onde existem atores com papel de *coordenador* para lidar com agentes do nicho que utilizam a plataforma para seus próprios interesses. Às vezes, o código fonte pode estar fechado a um consórcio, e.g., IntelliCAD. Os coordenadores devem fazer com que o processo de desenvolvimento da plataforma continue ativo e saudável, tratar a divulgação para atrair atores externos e gerir questões técnicas de interface e de usabilidade;
- *plataforma de software privada*: organizações chave como Microsoft, Apple e Autodesk querem maximizar seu valor ao avançar no mercado de software e serviços relacionados, associando-se com participantes específicos de um domínio ou nicho para levar soluções que os usuários não conseguiriam adquirir. O papel do coordenador é similar ao descrito acima, com a diferença de que neste cenário, atores com este papel almejam lucros.

Os seguintes parâmetros podem ser caracterizados (DHUNGANA *et al.*, 2010):

1. **Como são tratados os recursos finitos ou críticos, ou o seu reservatório?**: caracterizar arquitetura, código, tempo, dinheiro, usuários etc.;
2. **Como se dá o controle da população?**: caracterizar o incentivo para compra, uso e/ou extensão de produtos e serviços da plataforma;
3. **Quais são os tipos de interação entre os atores?**: caracterizar colaboração e competição no (entre) ECOS(s);

4. **Como é o processo de transferência de energia?:** caracterizar a troca e a tradução de conhecimento no ECOS;
5. **Qual é a base da definição dos processos?:** caracterizar ciclos de mercado e avanços tecnológicos, incluindo em ECOS FOSS;
6. **Como se dá o processo de adaptação ambiental?:** caracterizar a resposta dos atores em aceitar, rejeitar e/ou aprender a gostar do ECOS.

Por exemplo, analisando o ECOS Brechó:

1. Tem apenas bolsas de estudo para apoiar um participante do ECOS. O outro participante é estudante, mas profissional de indústria (sem bolsa). Não há outros tipos de recursos imediatos para apoiar agentes de nicho que desejam contribuir para a plataforma, o que requer uma definição clara de modelo FOSS para não ameaçar a sustentabilidade do ECOS;
2. Tem controle sobre o desenvolvimento da plataforma via repositório SVN, seguindo a estrutura linha de desenvolvimento principal (*trunk*), ramos (*branches*) e etiquetas para liberações (*tags*);
3. Possui atores que colaboram entre si para agregar valor ao ECOS, embora não atuem de forma cooperativa, no mesmo módulo;
4. Apoia o fluxo de conhecimento por meio de listas de discussão e realização de seminários sobre as atividades conduzidas na plataforma;
5. Apresenta um processo de desenvolvimento leve, focado na atividade de implementação, com documentação baseada em *javadoc* e em documento de requisitos e diagramas de classe;
6. Apesar de grande potencial para utilização na indústria de software, é um ECOS acadêmico, cuja plataforma consiste em um projeto de pesquisa (protótipo) e cujo processo de aceitação de atores externos ainda é rodeado de incertezas (SANTOS *et al.*, 2010a). Normalmente, novos participantes ou agentes de nicho procuram o ECOS Brechó pelo interesse em pesquisar na área de reutilização ou para desenvolver projetos na plataforma existente.

Passo 2.2: analisar diversidade do ECOS – tem por objetivo gerir o envolvimento de organizações chave, competidoras e externas, e de usuários, buscando dar suporte a (i) diferentes linguagens, infraestruturas e dispositivos, e (ii) diferentes grupos de usuários, papéis e domínios. A ideia é compensar a perda de atores do ECOS como acontecem as catástrofes em regiões do planeta onde algumas espécies têm que fazer o papel de outras que estão comprometidas, ou intensificar seu papel para manter o equilíbrio do ecossistema biológico, o que requer planos e/ou ações situadas (CUKIERMAN *et al.*, 2007). No caso do ECOSs, a vantagem está no fato de que o conhecimento sai da rede de produção de software pela perda de um ator, mas também permanece no ECOS, diferentemente dos nutrientes em ecossistemas biológicos. O elemento crítico para este passo é a *plataforma*, cuja diversidade pode impactar na combinação de *modelos de negócio híbridos* no ECOS, gerados a partir de escalas entre os seguintes fatores de negócio (POPP, 2011; 2012):

- **modelo de negócio versus modelo de recompensa:** um modelo de negócio determina que bens e serviços são fornecidos e por qual organização, apresentando um ou mais padrões de negócio para ser genérico e não definir como o negócio deve executar. Por outro lado, um modelo de recompensa

mostra como uma organização é compensada por cada um de seus padrões de negócio, não necessariamente envolvendo pagamento pois cada organização é livre para criar seus modelos. Os papéis que aparecem relacionados ao produto são: *criador* (que desenvolve), *distribuidor* (que disponibiliza), *vendedor de licenças* (que explora serviços) e *intermediário* (que agrega valor);

- ***open source comunitário versus open source comercial***: um modelo *open source* comunitário deve ser gerido por uma comunidade que cria, mantém e dá suporte para a plataforma FOSS e, às vezes, disponibiliza atividades de pré-venda e venda para organizações que oferecem produtos com uma versão *open source* e outra comercial, sendo o serviço provido de forma gratuita (*open license*). Por outro lado, um modelo *open source* comercial deve ser gerido por uma organização chave, que distribui a plataforma de modo customizado para seus clientes com licença padrão e cuja consultoria e manutenção devem ser prestadas por organizações externas profissionais.

Os seguintes parâmetros podem ser caracterizados (DHUNGANA *et al.*, 2010):

1. **O que significa diversidade para o ECOS?**: caracterizar a variedade da comunidade de desenvolvedores externos e de usuários;
2. **Por que ela é importante?**: caracterizar formas de aumentar o valor da plataforma por estender um mercado horizontal;
3. **Como ela ajuda o ECOS?**: caracterizar a exploração de novas oportunidades quando plataformas competitivas tornam um nicho obsoleto;
4. **Como assegurá-la no ECOS?**: caracterizar o apoio a diferentes linguagens, infraestruturas e dispositivos de hardware, além de novos projetos, domínios e grupos de usuários e de atores externos;
5. **Caso contrário, o que acontece?**: caracterizar porque a vantagem competitiva se degrada, resultando na redução ou extinção da comunidade.

Por exemplo, analisando o ECOS Brechó, seja do ponto de vista da plataforma de desenvolvimento ou de utilização:

1. Possui uma comunidade muito reduzida de participantes (passando de três para dois participantes, em 2013), além de agentes do nicho influenciadores e usuários (e.g., observadores ou ainda seguidores da plataforma);
2. Fez uma segunda tentativa de utilização na indústria ao evoluir de uma versão para o processo de gerência de reutilização para um ambiente ou mercado de componentes, podendo se aplicar para ECOSs, embora não tenha sido pensada a sua abertura para permitir projetos FOSS;
3. Revigorou-se com as novas direções da indústria de software brasileira, com as expectativas do Modelo MPS e as orientações da ISO 12207 da exigência de processos de reutilização nas organizações;
4. Além de ser uma plataforma para desenvolvimento de um ECOS, também pode ser utilizada como repositório de reutilização para os mais diferentes domínio e áreas do conhecimento, ao tratar o conceito de componente como encapsulamento de um conceito. Isso abre oportunidades para vários usuários, como acontece com o Portal EduES, tido como um arcabouço estruturante para um corpo de conhecimento de qualquer área;

5. Apesar das oportunidades e pontos fortes, apresenta riscos e pontos fracos, tais como: uso/dependência de versões de tecnologias mais antigas, o que é um problema típico das organizações consumidoras em ECOSs; inexistência de uma proposta clara de licença para desenvolvimento e utilização; falta de documentação completa; volatilidade de atores (voluntários ou bolsistas); e inexistência de uma estratégia de negócio para expandir e ousar cenários em larga escala, mesmo possuindo uma estratégia (Figura 4.8).

Passo 3: *modelar um framework de gerenciamento do ECOS* – tem por objetivo sumarizar os passos 2.1 e 2.2 ao modelar elementos de ECOSs, graficamente e conceitualmente, com base em (DHUNGANA *et al.*, 2010), os quais são destacados na Figura 4.7 e na Figura 4.8:

Atividade 1: *capturar contexto/estabelecer estratégias* – consiste em identificar as seguintes *questões* em cadeias de valor e em arquiteturas de ECOSs a partir da *visão de negócio dirigida pela inovação aberta* (CHESBROUGH, 2003; CAMPBELL & AHMED, 2010):

1. **Monitoramento e Gerenciamento do ECOS:** definir aspectos como segurança da plataforma, produtos e serviços; adaptabilidade sob demanda; viabilidade de mercado; saúde do ECOS; competição e produtividade; gestão de recursos, entre outros. Na plataforma Brechó-VCM, isso acontece através do próprio repositório de referência, com um banco de dados históricos;
2. **Recursos do ECOS:** definir aspectos como artefatos de software manipulados; licenças; acordos; documentação, entre outros, distribuídos em múltiplos bancos de dados. Na plataforma Brechó-VCM, isso acontece pelo mecanismo de apoio ao canal de distribuição ao incorporar elementos de negócio (Seção 2.5). Acontece também pela compreensão dos papéis dos atores envolvidos na plataforma (níveis de escopo da visão interna do ECOS) e na indústria de software (nível de escopo da visão externa do ECOS), utilizando uma cadeia de valor (Figura 4.7);
3. **Gerenciamento Local:** fornecer conhecimento para alimentar a estratégia de gerenciamento e monitoramento global do ECOS. Na plataforma Brechó-VCM, isso acontece pela existência de repositórios locais.

Atividade 2: *caracterizar gerenciamento do ECOS* – consiste em identificar as seguintes *perspectivas* em cadeias de valor e em arquiteturas de ECOSs para *integrar as dimensões técnica e social e melhorar o planejamento estratégico do ECOS* (PRADO, 2002; CAMPBELL & AHMED, 2010):

1. **Questões Técnicas:** definir a arquitetura da plataforma e de seus produtos visando integrar componentes desenvolvidos por organizações externas. Na plataforma Brechó-VCM, isso acontece pela participação de atores com papel de gerente da biblioteca, responsáveis por manter a qualidade da plataforma e de aplicações, componentes e serviços publicados;
2. **Considerações de Negócio:** definir planos para explorar os recursos do ECOS e cuidar das questões econômicas. Na plataforma Brechó-VCM, isso acontece pela participação de atores com papel de gerente de negócios, responsáveis por apoiar a cadeia de valor do ECOS ao longo do tempo, automaticamente ou não;

3. **Participação da Comunidade:** definir como diferentes atores interagem e como a comunidade criada usufrui dos lucros e/ou benefícios gerados a partir dos recursos disponíveis. Na plataforma Brechó-VCM, isso acontece pela participação de demais atores do ECOS, i.e., de usuários e organizações chave, a produtores e integradores de componentes.

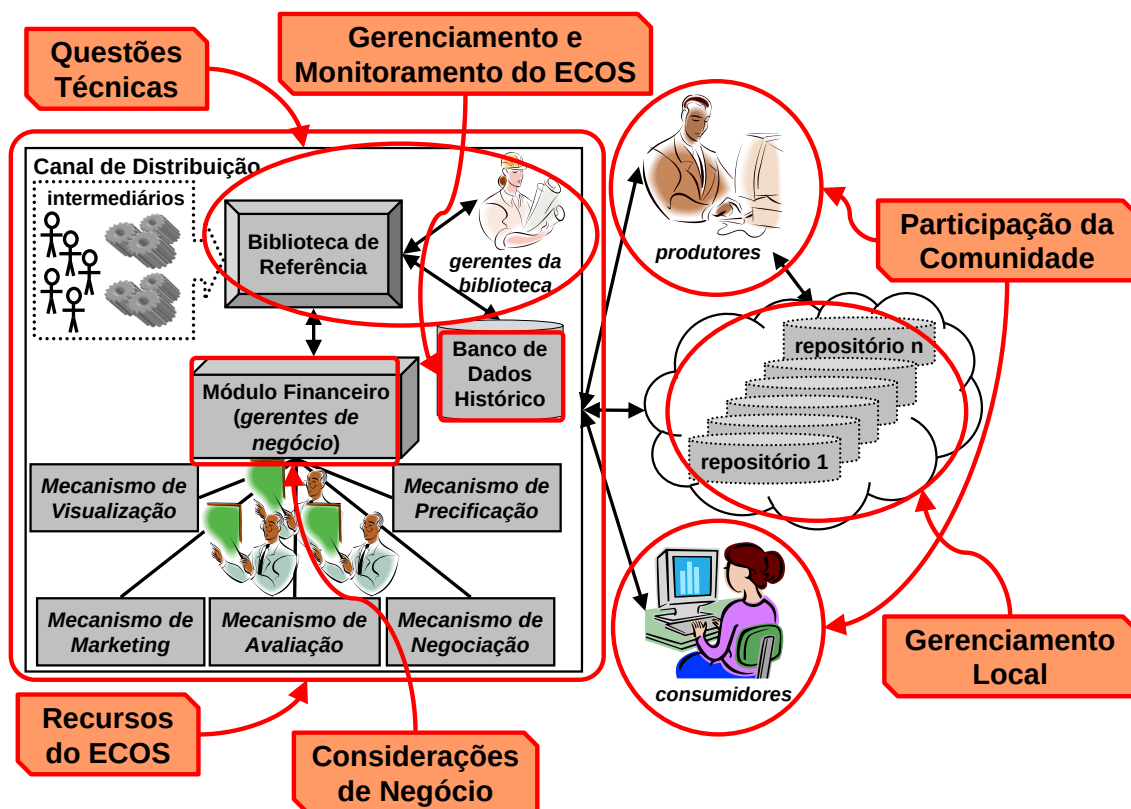


Figura 4.8 – Arquitetura da plataforma Brechó-VCM, analisada e marcada agora com elementos de gerenciamento de ECOSs. Elementos de negócio são incorporados por mecanismos que apoiam o canal de distribuição e o mercado de componentes (SANTOS, 2010). Fonte: (SANTOS & WERNER, 2011d)

4.2.3. Dimensão #3: Social

A terceira dimensão do *ReuseECOS* foca no elemento *envolvidos*, i.e., *stakeholders* do ECOS, referindo-se igualmente aos níveis de escopo *organizacional*, da *visão interna do ECOS* e da *visão externa do ECOS* (JANSEN *et al.*, 2009a). A meta é entender como a criação, organização e manutenção de redes sociais podem interferir nas comunidades que participam dos ECOSs, levando em consideração o conhecimento adquirido, trocado e transmitido nestas redes, o que gera as redes socio-técnicas, como sumariza a Figura 4.9. Consequentemente, identificar e analisar requisitos, bem como criar e gerenciar sistemas de software, podem ser consideradas atividades sociais na ES (CUKIERMAN *et al.*, 2007; SILVA & CÉSAR, 2009). Primeiro, *relacionamentos* entre atores e artefatos do ECOS devem ser identificados e modelados, extraídos de redes

sociais, IDEs, repositórios de controle de versão, repositórios de gerência de reutilização, entre outros, a fim de compreender a circulação de conhecimento e de valor sob a dimensão social. Em seguida, para cada ator envolvido, os *papéis* devem ser mapeados, considerando os aspectos temporal e contextual, a fim de contemplar a evolução e a distribuição no processo de ES das plataformas. Por fim, aplicar mineração de dados para extrair conhecimento das redes socio-técnicas, tomando por base um conjunto de indicadores estratégicos da organização chave, externa ou ainda usuária (dimensão transacional) e métricas de projeto da plataforma, aplicações, componentes e serviços (dimensão técnica). Assim, pode-se interferir no gerenciamento de um ECOS.

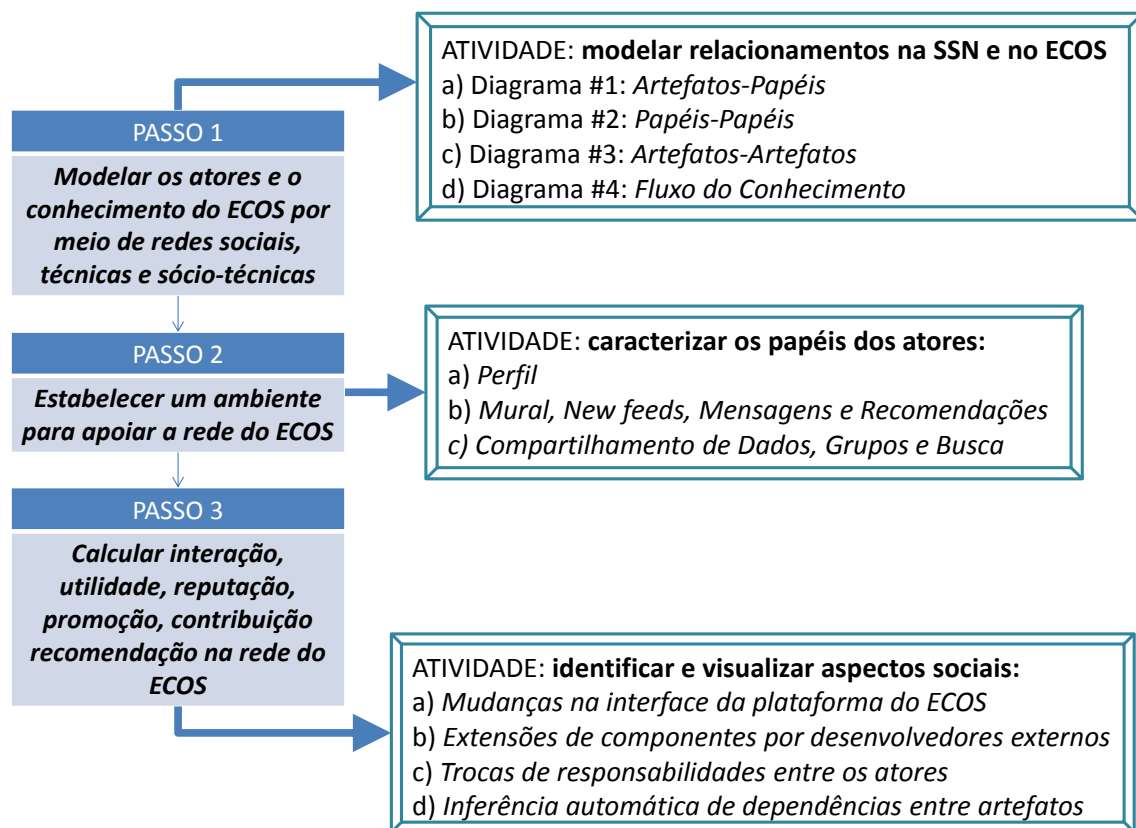


Figura 4.9 – Dimensão social do ReuseECOS. Fonte: (SANTOS & WERNER, 2012a)

A dimensão social visa também analisar os ECOSs por meio de analogias com outros tipos de ecossistemas, como biológicos e sociais, para obter e derivar métodos e modelos de organização, classificação e avaliação dos relacionamentos e papéis das redes socio-técnicas do ECOS (CAMPBELL & AHMED, 2010). Isso significa *balancear proposições e percepções de valor* dos envolvidos sob os pontos de vista de: (i) *utilidade* – por que os envolvidos integram, estendem e modificam conhecimento em um ECOS, e interagem entre si em um ambiente globalizado?; (ii) *promoção* – como a capacidade e o engajamento dos envolvidos são reconhecidos implícita e explicitamente

nesse ambiente?; e (iii) *conhecimento* – como o desenvolvimento *open source* e outras oportunidades na rede social do ECOS contribuem para os seus envolvidos?. Conforme BARBOSA *et al.* (2013), esta dimensão trata de conceitos que aparecem diretamente em um tópico de pesquisa investigado em ECOSs: *open source*. Para atender a este propósito, os seguintes passos e atividades são considerados pela dimensão social:

Passo 1: modelar os atores e o conhecimento do ECOS por meio de redes sociais, técnicas e socio-técnicas – consiste em utilizar um modelo para descrever todos os atores que participam da rede, acrescentando o *conhecimento* (i.e., artefatos como código, arquitetura, casos de teste, requisitos, bem como informações e recursos relacionados) e suas interações durante a transferência no ECOS. Isso pode se dar por meio da aplicação da teoria dos grafos, conforme propõem CAPURUÇO & CAPRETZ (2010), que exploram a recomendação social por meio de módulos que gerenciam a dependência com o tempo e a sensibilidade do contexto social (i.e., consciência da presença e da interação entre os membros). O modelo deve acessar o repositório de dados históricos e de relatos de experiência para estar constantemente atualizado e refletir o *status* do ECOS. Isso favorece o estabelecimento de três tipos de redes de relacionamento: (i) *rede socio-técnica*, entre atores e conhecimentos, e.g., diagramas papel-artefato; (ii) *rede social*, entre atores, e.g., diagramas papel-papel; e (iii) *rede técnica*, entre conhecimentos, e.g., diagramas artefato-artefato. A Tabela 4.3 exemplifica as relações entre os nós desta rede, segundo SEICHTER *et al.* (2010) e SANTOS *et al.* (2013b), e a Figura 4.10 ilustra um exemplo de abordagem para modelar redes técnicas, sociais e socio-técnicas. Em cada caso, o objetivo é compreender a circulação de conhecimento e de valor no ECOS do ponto de vista social, estruturando métodos de visualização de ECOSs (IYER, 2012). No caso dos ecossistemas do ECOS LENS/Reuse, não existe um modelo que descreve as diferentes redes, exceto diagrama de classes e de componentes da *Unified Modelling Language* (UML) coerentes com o código fonte (rede técnica), que descrevem alguns conhecimentos, e.g., o ECOS Brechó possui um diagrama de classes que descreve os relacionamentos entre as suas entidades. Adicionalmente, SEICHTER *et al.* (2010) analisam como os artefatos podem se relacionar com os atores do SECO e estabelecem quatro modos:

- **possuído por:** relaciona um artefato ao seu proprietário, ator responsável ou que possui o conhecimento a respeito, tornando-o visível para demais envolvidos e diminuindo a perda de conhecimento com a sua saída;
- **interessado em:** permite ao ator ou a grupos de atores serem informados sobre as atualizações de *status* ou quando novos dados forem publicados por um dado artefato, através de *new feeds*, mensagens e comentários de outros atores que possuem o mesmo interesse;
- **depende de:** ocorre no desenvolvimento de plataformas, onde muitos artefatos dependem de outros e, com as redes sociais, pode-se visualizar essas dependências, agregando valor pela utilização de *new feeds* ou mensagens;
- **membro de:** agrupa os atores dos ECOSs por interesse, como acontece nos sites de redes sociais *on-line*, mas com a participação de artefatos.

Tabela 4.3 – Tipos de interação entre atores e conhecimentos nos ECOSs.
Fontes: (SEICHTER et al., 2010) e (SANTOS et al., 2013b)

INTERAÇÃO	EXEMPLO
<i>Ator → Ator</i>	Atores podem se comunicar por publicações, compartilhamento de dados, mensagens e grupos (comunidades), influenciando o ECOS com sugestões, avaliações e comentários.
<i>Ator → Conhecimento</i>	Atores podem interagir com artefatos como interagem com outros atores em um <i>site</i> de rede social, incluindo classificação e comentários na mudança de <i>status</i> , enriquecidos com informação pública e provendo retorno a interações diretas com organizações chave, externas e usuários.
<i>Conhecimento → Ator</i>	A promoção do artefato permite que qualquer mudança de seu <i>status</i> possa também ficar visível pela notificação de novas comunicações e mensagens para os atores conectados, e.g., artefatos provenientes de repositórios de dados históricos de ECOSs ou de gerência de configuração.
<i>Conhecimento → Conhecimento</i>	Interação disseminada pela divulgação de informações e interpretação automática da dependência entre artefatos.

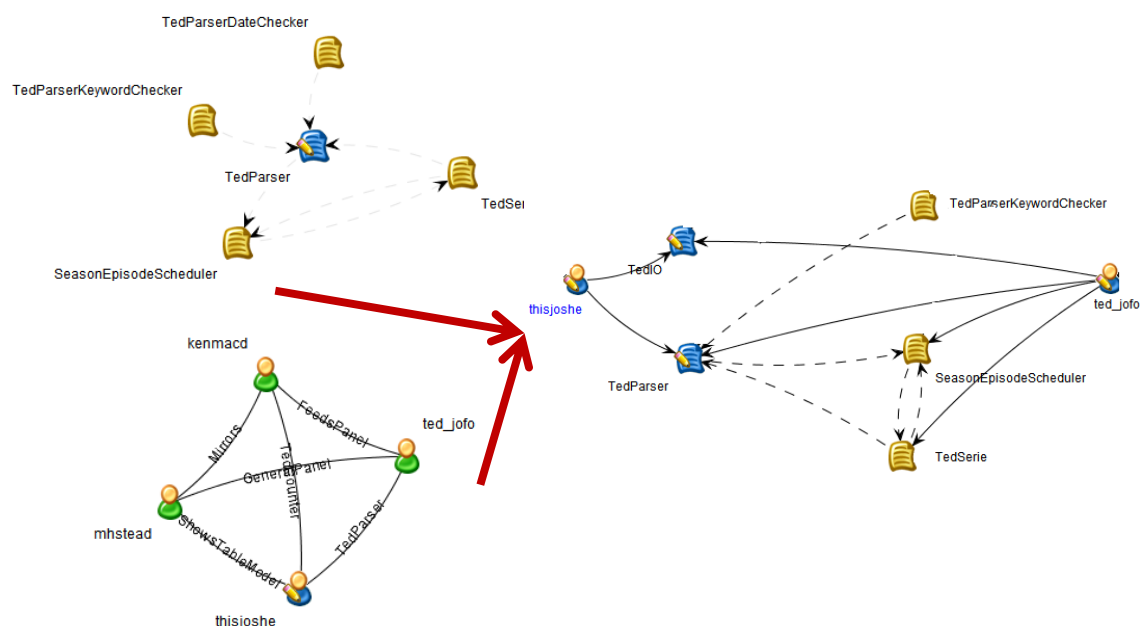


Figura 4.10 – Modelagem de redes em ES. Os atores são representados por nós em formato de avatar e os artefatos, por nós em formato de folha de papel. Fonte: (VAHIA et al., 2011)

Passo 2: estabelecer um ambiente para apoiar a rede do ECOS – consiste em estabelecer uma infraestrutura para apoiar a construção e manutenção das redes em um ambiente de desenvolvimento de software com ECOSs, como ilustra a Figura 4.11. Isso leva às combinações de tipos de interação entre os atores e conhecimentos, onde o foco é reter conhecimentos no ECOS. Por exemplo, no ECOS Brechó, esse ambiente existe pela combinação de diferentes tecnologias como lista de discussão *Yahoo*, utilização do repositório SVN (i.e., conhecimento na forma de artefatos da plataforma) e do *Bugzilla* (i.e., conhecimento na forma de solicitações sobre a plataforma). Pode-se ainda considerar os primeiros elementos sociais para apoiar a estrutura da rede sócio-técnica (SEICHTER et al., 2010; SANTOS et al., 2013b):

- **perfil:** registro de dados de atores ou conhecimentos para as redes dos ECOSs;
- **mural:** comentários que podem ser anexados aos dados de atores e/ou de conhecimentos da rede;

- **novas comunicações:** informações atualizadas sobre novos relacionamentos, novos conhecimento e novos comentários;
- **compartilhamento de dados:** *download* e *upload* de conhecimentos no ECOS e em sua plataforma (e.g., *commits* e *checkouts* do SVN);
- **agrupamento:** relacionamentos inferidos ou escolhidos entre atores, bem como dependências entre conhecimentos dos ECOSs;
- **pesquisa:** busca por atores e conhecimentos nas redes dos ECOSs;
- **sugestões:** sugestões de relacionamentos e de contatos entre atores nas redes dos ECOSs, assim como conhecimentos de interesse com base em recomendação;
- **mensagens:** comunicação pessoal entre os atores e conhecimentos das redes;

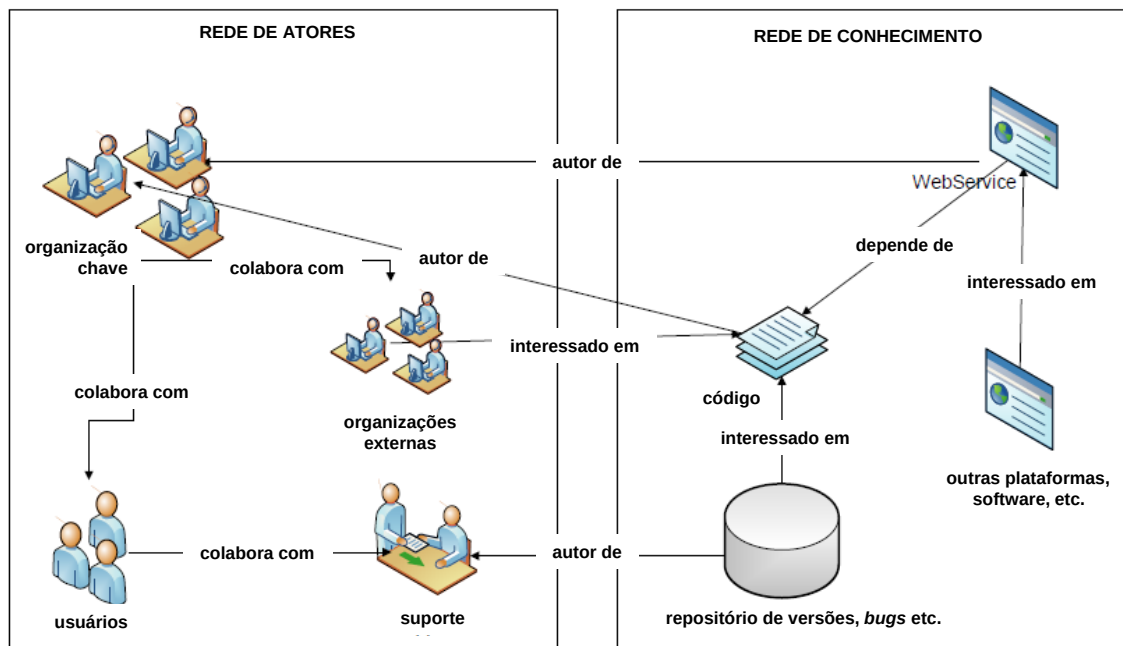


Figura 4.11 – Rede social de atores e conhecimentos e seus relacionamentos em um ambiente de desenvolvimento de software com ECOSs. Fonte: (SEICHTER *et al.*, 2010)

Passo 3: *calcular interação, utilidade, reputação, promoção, contribuição e recomendação na rede do ECOS* – consiste em manipular dados das redes a fim de extrair informação para apoiar os passos das dimensões técnica e transacional, tratando novos elementos sociais. Nesse sentido, é importante considerar os aspectos *temporal*, i.e., informação do ECOS em determinados momentos, e *contextual*, i.e., informação de contextos nos quais um ator ou conhecimento está inserido (CAPURUÇO & CAPRETZ, 2010). O modelo da rede socio-técnica deve explicitar as interações entre elementos (i.e., atores, conhecimentos e relacionamentos), bem como qualificar e quantificar a reputação destes e inferir ligações ou afinidades entre eles. Os impactos dessas manipulações afetam decisões referentes a: (i) mudanças de interface da plataforma do ECOS; (ii) extensões de artefatos por organizações externas, (iii) mudanças de responsabilidade entre atores e (iv) inferência automática de dependências entre conhecimentos do ECOS (MENS & GOEMINNE, 2011). A utilização da ferramenta StavSVN permite visualizar as interações entre um participantes da organização chave e o código fonte da plataforma do ECOS Brechó, e.g., atividade nos

pacotes da plataforma, percentual de modificações e impactos no código e artefatos alterados nos últimos *commits* realizados no repositório SVN, como mostram a Figura 4.12 (interação) e a Figura 4.13 (contribuição). No entanto, utilidade, reputação promoção e recomendação não representam as preocupações do ECOS Brechó até o momento. A Figura 4.14 apresenta uma arquitetura de solução, derivada da proposta descrita em (SANTOS & WERNER, 2012b) com o objetivo de (i) propiciar um ambiente para tratar o ciclo de vida social dos ECOSs (Seção 3.3.2), (ii) considerar o ambiente de desenvolvimento de software com ECOS (*Passo 2*), (iii) focar na rede socio-técnica subjacente e (iv) gerenciar o processo de inovação aberta. Esta solução está centrada em um *data warehouse*³⁶ estruturado em torno dos elementos da plataforma, bem como os relacionamentos com os envolvidos. Os intermediários podem analisar o repositório de dados históricos e de relatos de experiência para entender a evolução dos ECOSs que disponibilizaram suas informações e interferir na plataforma de um ECOS selecionado, e.g., como e quando tornar aberto um módulo da arquitetura da plataforma, como gerenciar as permissões de acesso etc. Além disso, pode-se administrar atributos como utilidade e promoção a partir da análise das redes sociais de um ECOS, visando contribuir para a sua saúde quanto a sustentabilidade e diversidade (CAMPBELL & AHMED, 2010).

Most Recent Commits		
marlon		
2008-09-16 20:51 Rev.: 410	16 lines of code changed in 1 file:	kernel/trunk/src/br/ufjf/cos/reuse/brecho/component/web/struts: ReleaseAction.java (+16)
marlon		
2008-09-13 13:49 Rev.: 409	1 lines of code changed in 1 file:	kernel/trunk/web/component: editComponentBody.jsp (+1 -1)
marlon		
2008-09-13 12:13 Rev.: 408	1086 lines of code changed in 3 files:	kernel/trunk/docs: dump.sql (+3 -3), dump2.sql (new 1080) kernel/trunk/web/component: editComponentBody.jsp (+3 -3)
marlon		
2008-09-11 19:20 Rev.: 396	1 lines of code changed in 1 file:	kernel/branches/Marlon: build.xml (+1 -2)
marlon		
2008-09-10 16:54 Rev.: 394	4 lines of code changed in 1 file:	kernel/branches/Marlon/src/br/ufjf/cos/reuse/biblioteca/category/web/struts: CategoryAction.java (+4 -4)
marlon		
2008-08-28 20:40 Rev.: 357	0 lines of code changed in 2 files:	kernel/branches/Marlon/src/br/ufjf/cos/reuse/biblioteca/template/field: Field.java (changed)
marlon		
2008-08-19 22:41 Rev.: 290	56 lines of code changed in 3 files:	kernel/branches/Marlon/src/WEB-INF/struts: messages.properties (+2 -1), messages_en.properties (+2 -1) kernel/branches/Marlon/src/br/ufjf/cos/reuse/biblioteca/category/web/struts: CategoryAction.java (+52 -1)

Figura 4.12 – Interações entre atores e artefatos na plataforma Brechó, extraídos pela ferramenta StatSVN, em formato de lista. Fonte: (SANTOS & WERNER, 2012a)

³⁶Data Warehouse corresponde a um sistema de computação utilizado para armazenar informações relativas às atividades de uma organização em bancos de dados, de forma consolidada. O desenho da base de dados favorece os relatórios, a análise de grandes volumes de dados e a obtenção de informações estratégicas que podem facilitar a tomada de decisão (WIKIPEDIA, 2013).

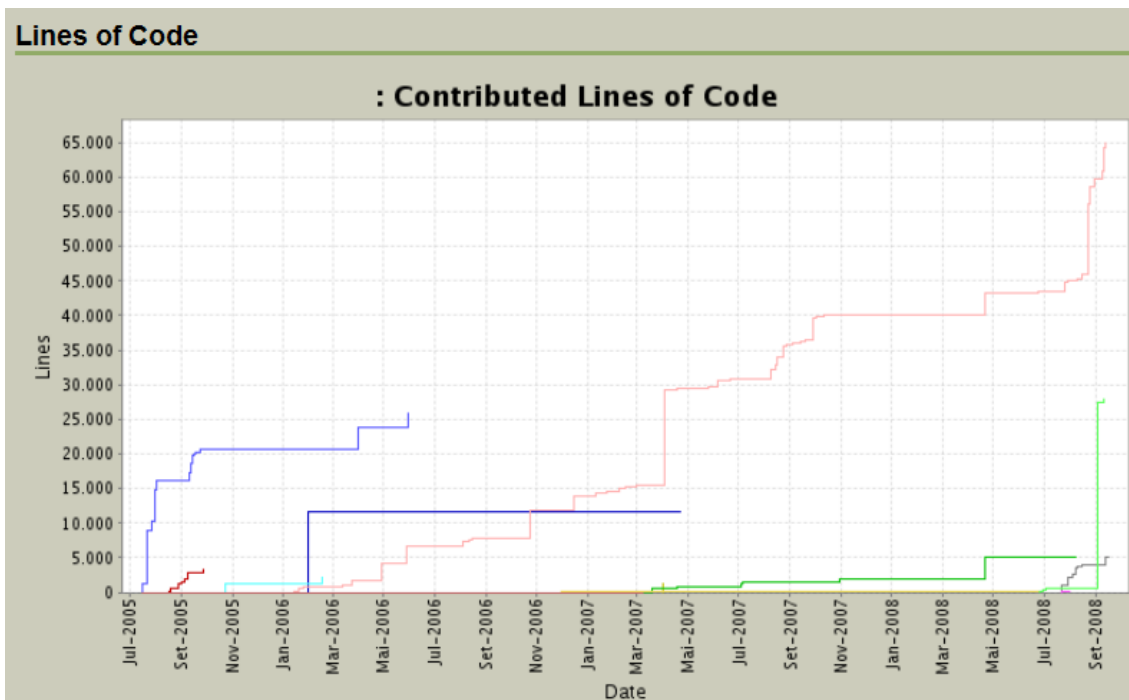


Figura 4.13 – Interações entre atores e artefatos na plataforma Brechó (LoC por modificações e contribuições), extraídos pela ferramenta StatSVN. Fonte: (SANTOS & WERNER, 2012a)

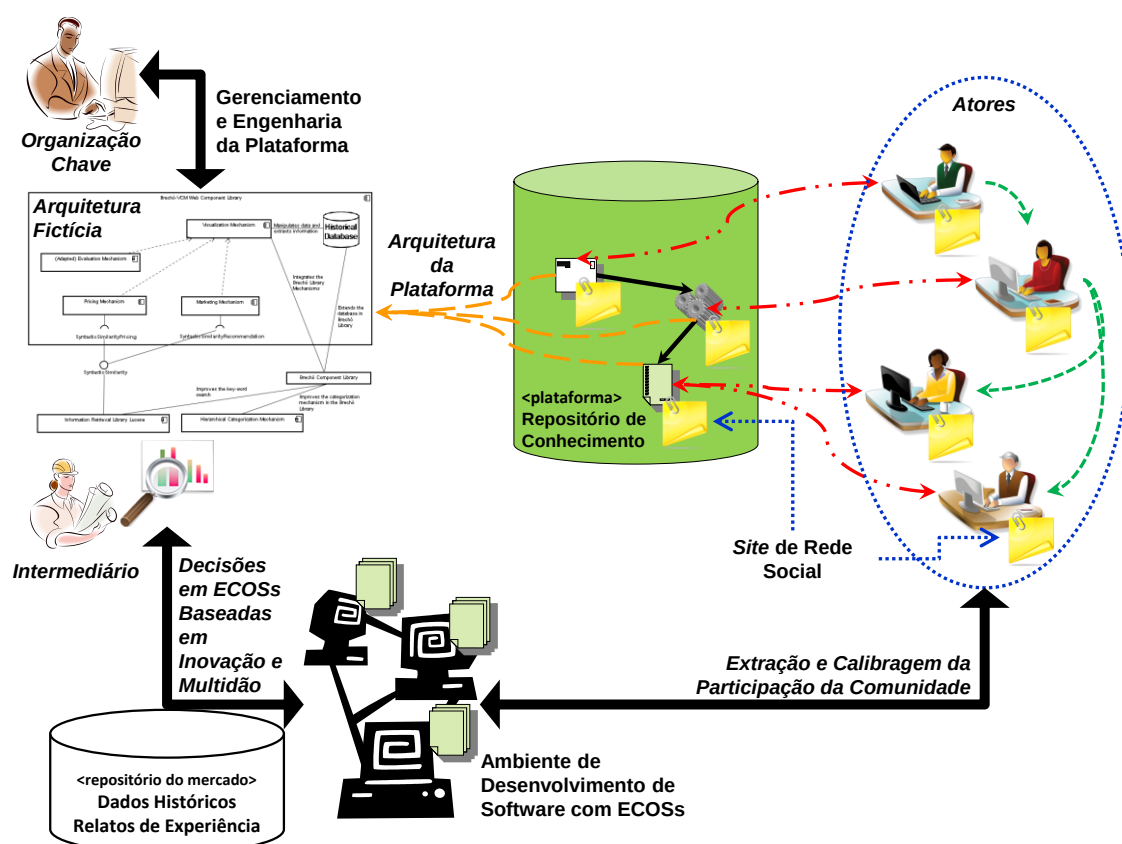


Figura 4.14 – Arquitetura para apoiar o ciclo de vida social de ECOSs, baseado em (SANTOS & WERNER, 2012b). Fonte: (SANTOS *et al.*, 2013b)

4.3 Proposta de Ecossistemas de Software Baseada em Reutilização

A fundamentação teórica sobre conceitos de reutilização realizada no Capítulo 2 permitiu a elaboração de uma linha do tempo com suas quatro gerações (Seção 3.2) e contribuiu para a opinião compartilhada por vários pesquisadores de que ECOSs pode estressar o potencial da reutilização na indústria de software, almejado desde os anos de 1980 (FRAKES & KANG, 2005). Considerando ainda a base de conhecimento inicial construída pelo *framework ReuseECOS*, esta seção discute alguns conceitos de reutilização a fim de adaptá-los e aplicá-los no contexto de ECOSs. Primeiramente, o pilar central consiste de uma ou mais **plataformas de ECOSs** que, sob o ponto de vista de ES, consistem em vários conjuntos de produtos e serviços de software relacionados, cujas fronteiras podem ser estabelecidas por mercados, tecnologias, infraestruturas ou organizações (Seção 3.3.2). Disso, deriva-se o conceito de **arquiteturas de plataformas**, que estendem arquiteturas de sistemas para tratar a governança de elementos de negócio, organizacionais, de TI e/ou de software conhecidos na literatura e na prática, de acordo com as fronteiras definidas.

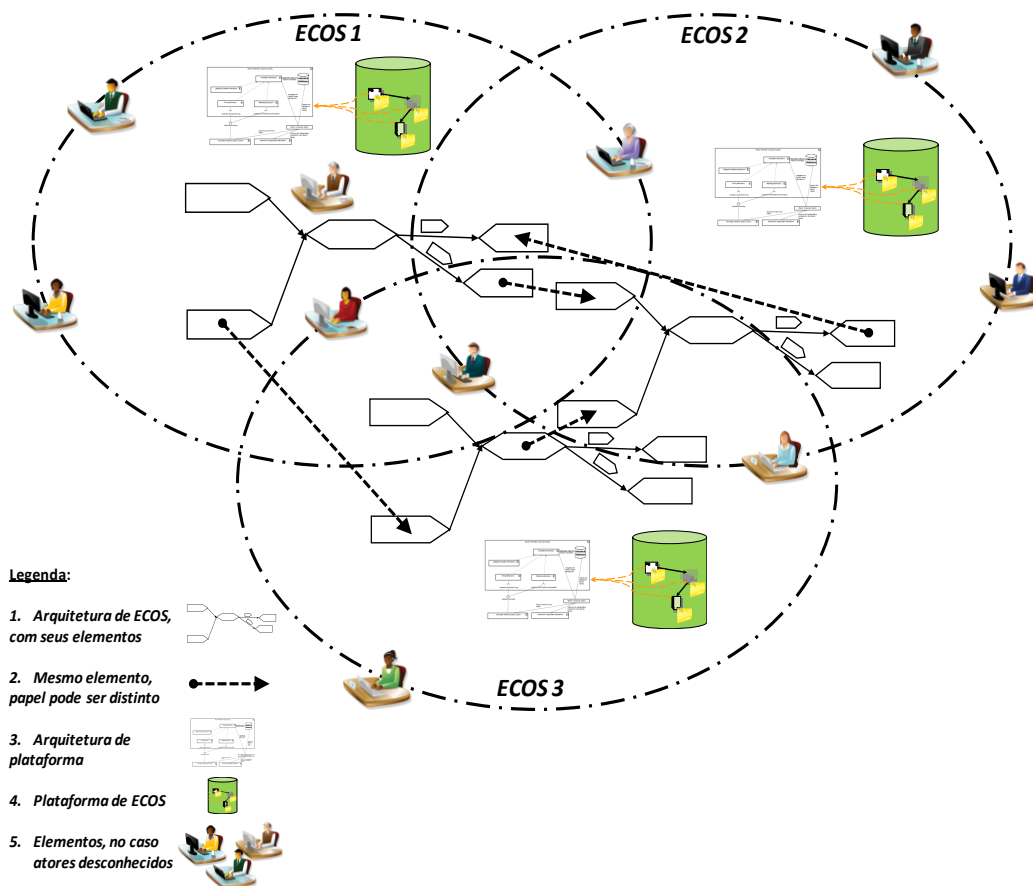


Figura 4.15 – Exemplo dos conceitos de ECOSs

Uma arquitetura de plataforma corresponde a uma arquitetura de software específica para domínio, pois pertence ao nível concreto da compreensão dos ECOSs (Seção 1.4) e, portanto, é o elemento chave de ED, LPS e DBC. Por outro lado, no nível conceitual da compreensão dos ECOSs, surge o conceito de **arquiteturas de ECOSs**, que consistem nas RPSs modeladas para organizar os diferentes elementos que constituem os ECOSs ao definir papéis, responsabilidades e relacionamentos, conforme as perspectivas de ECOSs (ver Figura 3.4). Neste caso, os **elementos de ECOSs** são seus atores e artefatos, bem como seus relacionamentos, cuja variabilidade é definida justamente por papéis e responsabilidades. A Figura 4.15 sumariza os conceitos apresentados nesta seção, flexibilizando a notação para modelar RPSs proposta por BOUCHARAS *et al.* (2009) ao exibir atores aparentemente desconhecidos nos ECOSs (anônimos). São utilizadas também linhas pontilhadas para representar os diferentes recortes das fronteiras dos ECOSs, bem como setas pontilhadas para mostrar as dificuldades de representar a variabilidade, e.g., de um mesmo ator que assume diferentes papéis. As subseções a seguir os expandem com definições mais elaboradas para a construção desta proposta de tese.

4.3.1. Domínios em Ecossistemas de Software

As conclusões obtidas pelos Capítulos 3 confirmam o que estudos secundários sobre ECOSs expuseram recentemente, sobre o fato que a terminologia e as definições da ECOSs variam amplamente na academia e na indústria, e mesmo o conceito de ECOS ainda é vago e diverso. A área de reutilização pode contribuir nesta direção com a definição de domínio de ECOSs. O **domínio de ECOSs** consiste em determinar como objeto de estudo a própria abordagem de ECOSs, a fim de estabelecer um vocabulário comum para os interessados, que desejam compartilhar informações, realizar inferências ou reutilizar conhecimento sobre ECOSs. Estendendo a ideia de (SHAPER, 1974), a construção do domínio de ECOS deveria considerar as *relações* compreensíveis e existentes entre os itens de informação ou *conceitos* identificados na classe de ecossistemas em geral (e.g., de negócio, sociais e biológicos). No entanto, isso deve ser validado pela comunidade acadêmica e industrial de ECOSs para que se torne o conhecimento necessário para viabilizar a modelagem de ECOSs no contexto da área de ES (SANTOS & WERNER, 2010a).

O conhecimento de ECOSs pode ser armazenado em uma base que utiliza modelos conceituais (e.g., ontologias, uma vez que esta abordagem apoia a ED, como discutido na Seção 2.2.4). Assim, a partir de uma **base de conhecimento**, um determinado ECOS pode ser instanciado por meio de uma atividade que visa apoiar um interessado a traduzir entidades de um ECOS real para um modelo conceitual composto por atores e artefatos atrelados a papéis, responsabilidades e relacionamentos. Esta atividade é denominada de **análise de domínio (AD) de ECOS**, cuja entrada é o conhecimento da realidade e a saída é o conhecimento formalizado através do *modelo de domínio de ECOSs*. Por sua vez, o modelo conceitual produzido pela AD de ECOSs corresponde à entrada de outra atividade, que utiliza os conceitos e relações instanciados em um dado ECOS para construir a sua RPS, consultando a base de conhecimento para evitar que atores e artefatos comuns a outros ECOSs previamente instanciados sejam duplicados. A RPS resultante consiste na arquitetura do ECOS e a atividade responsável pela sua produção é denominada **projeto de domínio de ECOS**.

No entanto, devido ao conceito de aplicações sociotécnicas apresentado na Seção 4.1, o objetivo passa a ser a construção de um modelo que enriqueça uma RPS com mais aspectos do senso de negócio e que introduza o senso de comunidade. Por exemplo, ao invés de exibir apenas os papéis e os relacionamentos entre as organizações de software em um ECOS como acontece de maneira estática (BOUCHARAS *et al.*, 2009), integrar elementos que permitam a atualização da RPS de maneira dinâmica. Isto significa adicionar aos atores e aos artefatos representados: (i) atributos de negócio *crédito*, *ponto*, *credibilidade*, *perfil* (de mercado) e *insight*, manipulados por mecanismos de precificação, de *marketing*, de avaliação, de negociação e de visualização, conforme definido na Seção 2.5; e (ii) atributos de comunidade *interação*, *utilidade*, *reputação*, *promoção*, *contribuição* e *recomendação*, manipulados por mecanismos de redes sociais, i.e., perfil, mural, novas comunicações, compartilhamento de dados, agrupamento, pesquisa, sugestões e mensagens, definidos na Seção 4.2.3.

Dessa forma, a saída da atividade de projeto de domínio de ECOS consistiria na **rede sociotécnica do ECOS**, cujo conceito depende da definição de ECOSs, que está sendo construída ao longo deste capítulo, explicada a seguir (Seção 4.3.3). Adicionalmente, a partir da base de conhecimento, pode-se extrair um **caso de ECOS**, i.e., uma seleção de um conjunto de elementos e conexões que representa um subdomínio do ECOS instanciado na base de conhecimento. Isso permite a realização de análises de ECOSs para o entendimento de sua história, após configurações e

recortes de elementos e suas conexões do ponto de vista de um interessado, no nível organizacional, da visão interna e/ou da visão externa. Um caso de ECOS pode utilizar repositórios que tratam de elementos específicos, e.g., (i) a AD de ECOS acessa a base de conhecimento para verificar o subdomínio através do modelo conceitual e mantê-la (CORCHO *et al.*, 2003); e (ii) o projeto de domínio de ECOSs acessa repositórios de reutilização para recuperar informações de artefatos e, eventualmente, *sites* de rede social para recuperar informações de atores. Como um todo, o processo composto pelas atividades de análise e de projeto de domínio de ECOSs é denominado **ED de ECOSs**. Essas atividades fazem com que a abordagem *ReuseSEEM* seja utilizada como uma nova abstração para as técnicas de reutilização como ED, LPS e DBC, visando tratar suas questões econômicas e sociais em nível mais alto.

Além disso, a partir dos casos de ECOSs, métodos, técnicas e ferramentas podem ser construídos para possibilitar melhor compreensão dos seus impactos na ES tradicional. Como discutido na Figura 3.4, nos estudos de caso/provas de conceito publicados, um elemento tem um papel ou responsabilidade em um caso de ECOS ou em outros ECOSs, que pode ser alterado pela configuração ou recorte feito por um novo ponto de vista do interessado. Este fato ressalta a importância do conceito de rede em ECOSs (JANSEN *et al.*, 2009a; CAPURUÇO & CAPRETZ, 2010; SEICHTER *et al.*, 2010; POPP, 2011; YU & DENG, 2011). Além disso, a constituição da rede expõe o conceito de **variabilidade de ECOS**, enunciado na Seção 4.1, que pode servir para lidar com as diferenças entre os ECOSs, bem como explorar as características comuns entre eles, o que contribui para a sua manutenção. Por exemplo, um ator desconhecido (e.g., organização externa) pode se transformar em um participante do ECOS através de modelos de associação (ANGEREN *et al.*, 2011), tais como parceria ou sociedade (Seção 4.2.1), dessa forma, alterando o seu papel no ponto de vista deste ECOS, mesmo que represente a mesma entidade.

4.3.2. Componentes em Ecossistemas de Software

Como no caso do conceito de domínio em ECOSs, o conceito de componentes em ECOSs também pode ser enunciado. Por exemplo, a organização chave do ECOS *SWCompany*³⁷ resolve modelar seu ecossistema e realiza a atividade de AD de ECOS para instanciar seus elementos e respectivas conexões a partir do modelo de domínio de

³⁷Nome fictício.

ECOSs previamente mantido. A título de ilustração, a Figura 4.16 ilustrar parte de um modelo conceitual de ECOSs representado pelo diagrama de classes da UML, contemplando alguns atores e a variabilidade de seus papéis. Este modelo foi gerado como tentativa inicial de refinar um conjunto mínimo de atores a partir do *framework ReuseECOS* e dos estudos secundários disponíveis (BARBOSA & ALVES, 2011; MANIKAS & HANSEN, 2013). A partir desse modelo e dos conceitos relacionados a seguir, a RPS do ECOS *SWCompany* pode ser gerada, como mostra a Figura 4.17 (ver notação na Seção 3.3.2):

- <<ECOS>> Nome: *SWCompany*
O ECOS tem o nome da própria organização.
- <<plataforma>> Sistema: *SWCompany*
A plataforma é o portfólio de projetos do domínio de atuação.
- <<papel>> Organização Chave: *SWCompany*
- <<papel>> Fornecedor: SAP, Organização Externa
SAP é conhecida e Organização Externa não é.
- <<artefato>> Produto: P1
Produzido pela Organização Externa.
- <<artefato>> Serviços: S1
Provido pela SAP

Apesar da visão interna do ECOS ter sido mostrada pela RPS, na Figura 4.18, percebe-se que o ator SAP, com papel *fornecedor* (*supplier*), pode ter sido registrado na base com o papel *organização chave* (*keystone*), durante a instanciação do ECOS SAP, cuja plataforma SAP é um artefato que compõe o ECOS *SWCompany* e que mantém uma relação comercial por meio de serviços relacionados (*S1*). Por representar o mesmo ator, mas com papéis distintos, o elemento SAP não deve ser duplicado, mas **relacionado** ou **marcado** pelo ECOS *SWCompany* na sua instanciação. A organização pode selecionar um conjunto de elementos e conexões do ECOS para analisar um subdomínio específico, o que consiste em explorar a variabilidade do ECOSs. Por exemplo, a organização chave pode querer alterar um fornecedor, ou bloquear um desenvolvedor externo. A meta está em gerar produtos e serviços de software relacionados a partir do **repositório de ativos do ECOS**, i.e., base preparada para armazenar e recuperar produtos e serviços que compõem ou derivam da plataforma de

ECOS, estendida com atributos de negócio e de comunidade. A combinação da base de conhecimento com o repositório de ativos do ECOS constitui o **repositório de ECOS**.

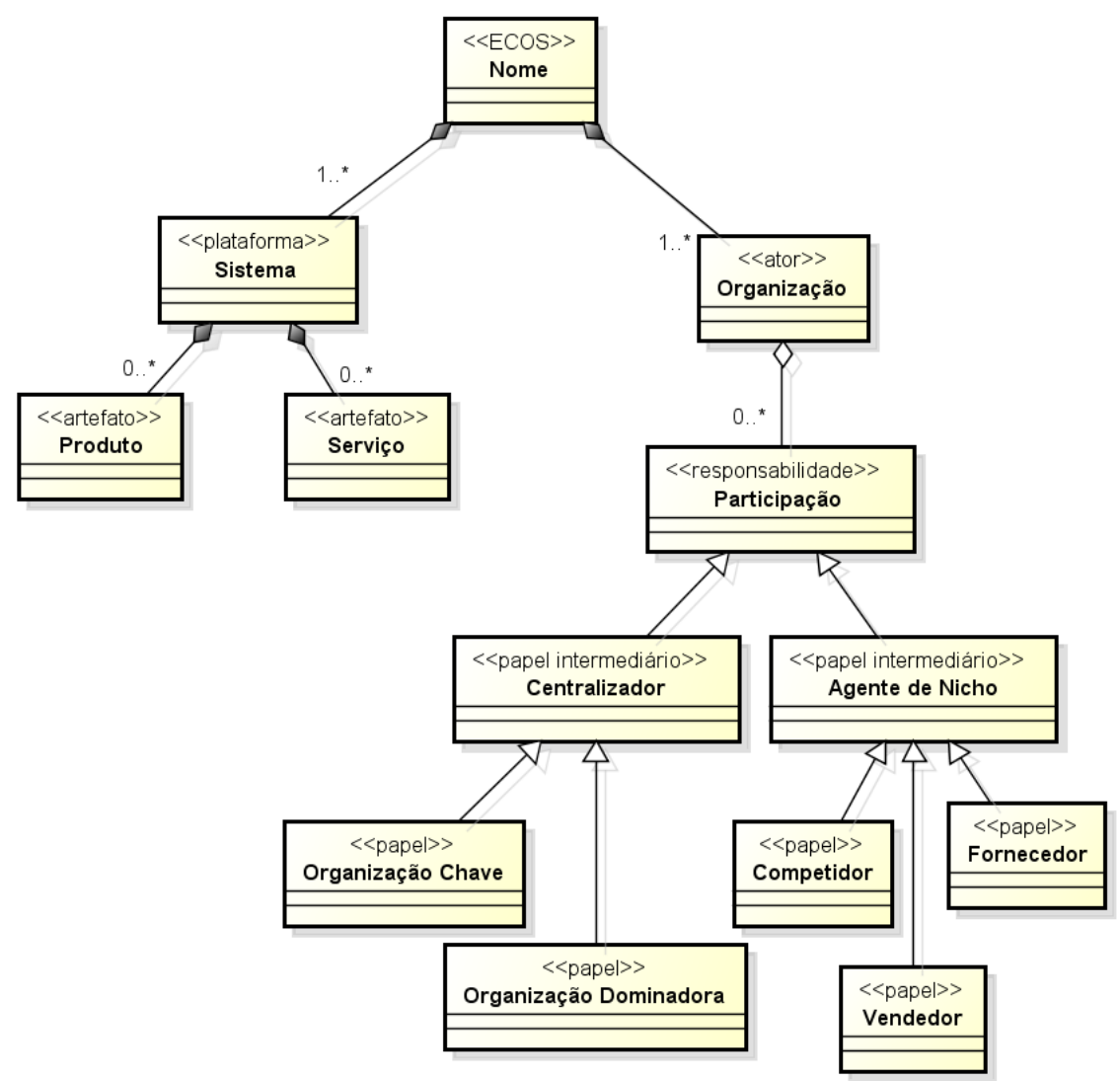


Figura 4.16 – Parte de um modelo conceitual de ECOSs preliminar

ECOS SWCompany

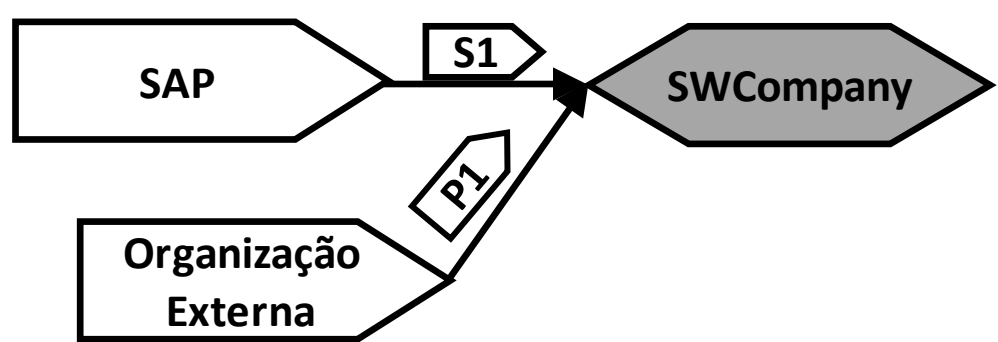


Figura 4.17 – Parte da RPS do ECOS SWCompany

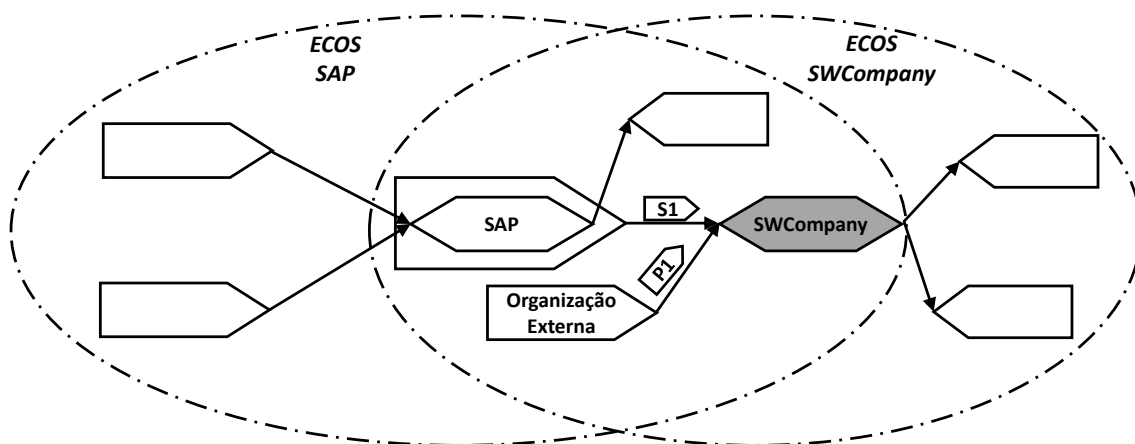


Figura 4.18 – Variabilidade em ECOSs, onde um ator pode assumir diferentes papéis dependendo do ponto de vista do interessado. Optou-se pelo interesse sobre o ECOS *SWCompany*, cujo papel do ator SAP é de fornecedor, após uma adaptação da proposta de BOUCHARAS *et al.* (2009). O ajuste corresponde a: (1) nível de escopo interno do ECOS representado por linha tracejada pela visão externa do ECOS (ao invés de linha contínua); e (2) organização SAP representada por dois símbolos, indicando seu papel sujeito ao ponto de vista (ao invés de um símbolo)

O conjunto formado em diferentes níveis de abstração do ECOSs, i.e., base de conhecimento, modelo conceitual instanciado, dados históricos, relatos de experiência, arquitetura de ECOS, atores, artefatos, relacionamentos, papéis, responsabilidades, plataforma de ECOS e ativos e ECOS, são chamados **componentes de ECOSs**. Estendendo WERNER & BRAGA (2005), por serem tratados como componentes, os elementos e suas conexões no ECOS que devem ser definidos de forma autocontida, com identificação clara, com descrição do papel ou responsabilidade específico, em conformidade com o modelo do respectivo nível de abstração, documentação apropriada e um grau de reutilização definido, independentemente do seu tipo. Por fim, outros conceitos podem ser entendidos após definição de componentes de ECOSs, na atividade de projeto do domínio de ECOS. Por exemplo, a arquitetura do ECOS segue um **estilo**, i.e., um conjunto de regras que identificam os componentes de um ECOS a partir da rede sociotécnica. Além disso, para apoiar a definição e compreensão de novos ECOSs, pode-se recorrer a **padrões**, i.e., disposição ou arranjo dos componentes de um ECOS que refletem suas variações e similaridades, regras e orientações para boa formação da rede sociotécnica, através de uma linguagem de padrões.

4.3.3. Definição de Ecossistemas em Engenharia de Software

À luz da proposta de ECOSs baseada em reutilização discutida nessa seção, esta proposta de tese apresenta uma definição de ECOSs que tenta levar em consideração um olhar sociotécnico da ES (Seção 1.1). O olhar sociotécnico aparece quando a academia e

a indústria focam nas aplicações sociotécnicas (Seção 4.1), o que permite construir redes “sociotécnicas”, e não somente “socio-técnicas” como é feito tradicionalmente. As redes socio-técnicas são redes técnicas, modeladas por grafos, onde os nós maiores correspondem aos artefatos de software e as arestas são as dependências existentes entre eles. Essas redes são estendidas para que os desenvolvedores sejam representados por nós menores e a aresta entre um desenvolvedor e um artefato tenha alguma semântica, e.g., criação ou atualização do artefato (MAGDALENO, 2010). Estendendo a discussão de CUKIERMAN *et al.* (2007), as **redes sociotécnicas dos ECOSs** ainda consistem em grafos, como na socio-técnica, mas com uma visão interdisciplinar, incluindo outros elementos além de artefatos de software e de desenvolvedores. Com isso, busca-se criar meios para analisar fatos científicos e artefatos tecnológicos a partir de sua complexidade, em função de quatro características (SANTOS, 2010):

- *o local, o situado (resistência ao global, ao universal), o caso a caso, a contingência*: promover trocas de informação entre repositórios de reutilização (i.e., locais, específicos de domínio e de referência) e o repositório de ECOSs, para ampliar a compreensão da rede;
- *os conhecimentos não formalizáveis*: reconhecer a importância da sinergia entre conhecimento explícito e conhecimento tácito nos ECOSs, a fim de contemplar conhecimentos transferíveis e não-transferíveis entre as entidades, e/ou armazenáveis e não-armazenáveis nos repositórios;
- *a complexidade (em vez de simplificações)*: ao analisar um caso de ECOS, evitar a divisão da complexidade do objeto de estudo, i.e., determinado ECOS, em questões técnicas ou não, abrangendo-as de forma integrada e de diferentes perspectivas, como organizacional, de mercado, tecnológica e de infraestrutura (JANSEN *et al.*, 2009a);
- *os transbordamentos (em vez de enquadramentos)*: apesar da possibilidade de fazer uma configuração ou recorte de um caso de ECOS, a variabilidade deve funcionar como um aspecto que amplia o enquadramento realizado. Deve-se viabilizar um **recorte fluido**, i.e., manutenção de um mecanismo de visualização que explore o potencial da graficabilidade, da interação e da navegabilidade indeterminada nas redes sóciotécnicas, para ampliar a representação da ES.

A combinação dessas características da rede sóciotécnica conduz a uma visão que mistura os componentes de ECOSs e contempla a imbricação sóciotécnica, o que foi mostrado, inclusive, a partir do uso de linhas pontilhadas que estabelecem as fronteiras entre os níveis de escopo da visão interna e externa dos ECOSs na Figura 4.18 (e não com linhas contínuas, como na notação original). Os interessados em analisar e compreender os ECOSs devem ser capazes de depreender as múltiplas combinações entre os componentes dos ECOSs, e não que estas sejam somente estabelecidas por um modelo estático, como vem sendo feito (BOUCHARAS *et al.*, 2009). Sejam três ou dois elementos básicos para definição (Seção 3.3.1), os ECOSs devem tratar novos modelos de negócio, inovação aberta e valor de TI na indústria de software. O termo “rede”, de caráter geral, tem menos impacto que o termo “comunidade”, mais utilizado para o contexto de FOSS, uma vez que os ECOSs não devem ser enxergados como extremos, e sim como híbridos, dado a vascularidade dos relacionamentos e a sobreposição dos papéis. As entidades podem ser conhecidas ou não, o que impacta o desenvolvimento de software em modelos de negócio híbridos. Outro ponto importante é manter o software como elemento básico, por se tratar de ECOSs, e expandir as atividades de engenharia e de gerenciamento para lidar com este contexto emergente na indústria de software. A Figura 4.19 expõe a definição adotada.

Ecossistemas de Software são redes sociotécnicas de desenvolvimento de produtos e serviços de software relacionados, formadas por um conjunto de componentes, conhecidos ou não, que se relacionam para realizar a engenharia e o gerenciamento de uma ou mais plataformas e para gerar valor e inovação na indústria de software

Figura 4.19 – Definição de ECOSs para esta proposta de tese

4.3.4. Requisitos da Abordagem *ReuseSEEM*

A proposta de ECOSs baseada em reutilização e o *framework ReuseECOS* permitiram a identificação dos principais requisitos (*Ri*) para o desenvolvimento da abordagem *ReuseSEEM*, considerando as fases de definição, modelagem e análise de ECOSs, foco desta proposta de tese, exibidos a seguir:

- R1: A abordagem deve refinar um conjunto mínimo de conceitos e suas relações no domínio de ECOSs, consensuais sob o ponto de vista da literatura explorada e corroborados pelos especialistas no tema;

- R2: A abordagem deve construir uma base de conhecimento, composta por um glossário de termos de ECOSs para conter os conceitos e respectivos significados;
- R3: A abordagem deve construir ainda um modelo conceitual, utilizando, para isso, um modelo conceitual de apoio para a manipulação do glossário de termos de ECOSs;
- R4: A abordagem deve identificar os principais tipos de atores e de artefatos, para representar a variabilidade através de redes sociotécnicas de ECOSs, ao explorar papéis, responsabilidades e relacionamentos;
- R5: A abordagem deve construir um repositório de ativos de ECOSs, composto por uma biblioteca de reutilização, que gerencia os artefatos identificados no domínio *de* ECOSs (i.e., produtos e serviços), e por um *site* de rede social, que gerencia a participação dos atores;
- R6: A abordagem deve construir um ambiente para automatizar as atividades de análise e projeto de domínio *de* ECOSs, a partir da realização de R1 a R5;
- R7: A abordagem deve criar um mecanismo para configuração ou recorte de casos de ECOSs para a extração de subdomínio de um ECOS, a partir da realização de R6;

4.3.5. Infraestrutura da Solução

A quarta dimensão do *framework ReuseECOS*, denominada *E&G de ECOSs*, foi introduzida por SANTOS & WERNER (2012b) e foca em intercambiar e integrar os passos e atividades das três dimensões apresentadas a fim de viabilizar a infraestrutura de apoio à abordagem proposta (Figura 4.20). Para isso, utiliza as três relações entre elas para estender o DPE e contemplar um ambiente de apoio à ES globalizada (Figura 4.1). Estas relações correspondem a:

- **Relação #1:** *motivar o desenvolvimento e evolução da plataforma* – aparece pela relação entre as dimensões técnica e transacional. Foca na compreensão de relacionamentos entre elementos dos ECOSs e modelos de sistema mantidos na plataforma de modo mais amplo, incluindo elementos do desenvolvimento de software globalizado (e.g., terceirização e internacionalização);
- **Relação #2:** *contribuir para o estabelecimento da plataforma* – aparece pela relação entre as dimensões técnica e social. Foca em realçar o envolvimento e atenção da comunidade existente em torno da plataforma do ECOS, analisando os potenciais e fraquezas para um ambiente globalizado;

- **Relação #3: mapear as proposições e percepções de valor** – aparece pela relação entre as dimensões transacional e social. Foca em mapear as considerações de valor aos níveis de escopo organizacional, da visão interna e externa do ECOS sob o ponto de vista da ES globalizada. Neste caso, *proposição de valor* é o conceito de valor gerado sob o ponto de vista de cada um dos envolvidos no ECOS, e *percepção de valor* consiste no entendimento do valor capturado por cada envolvido no ambiente e no contexto do ECOS.

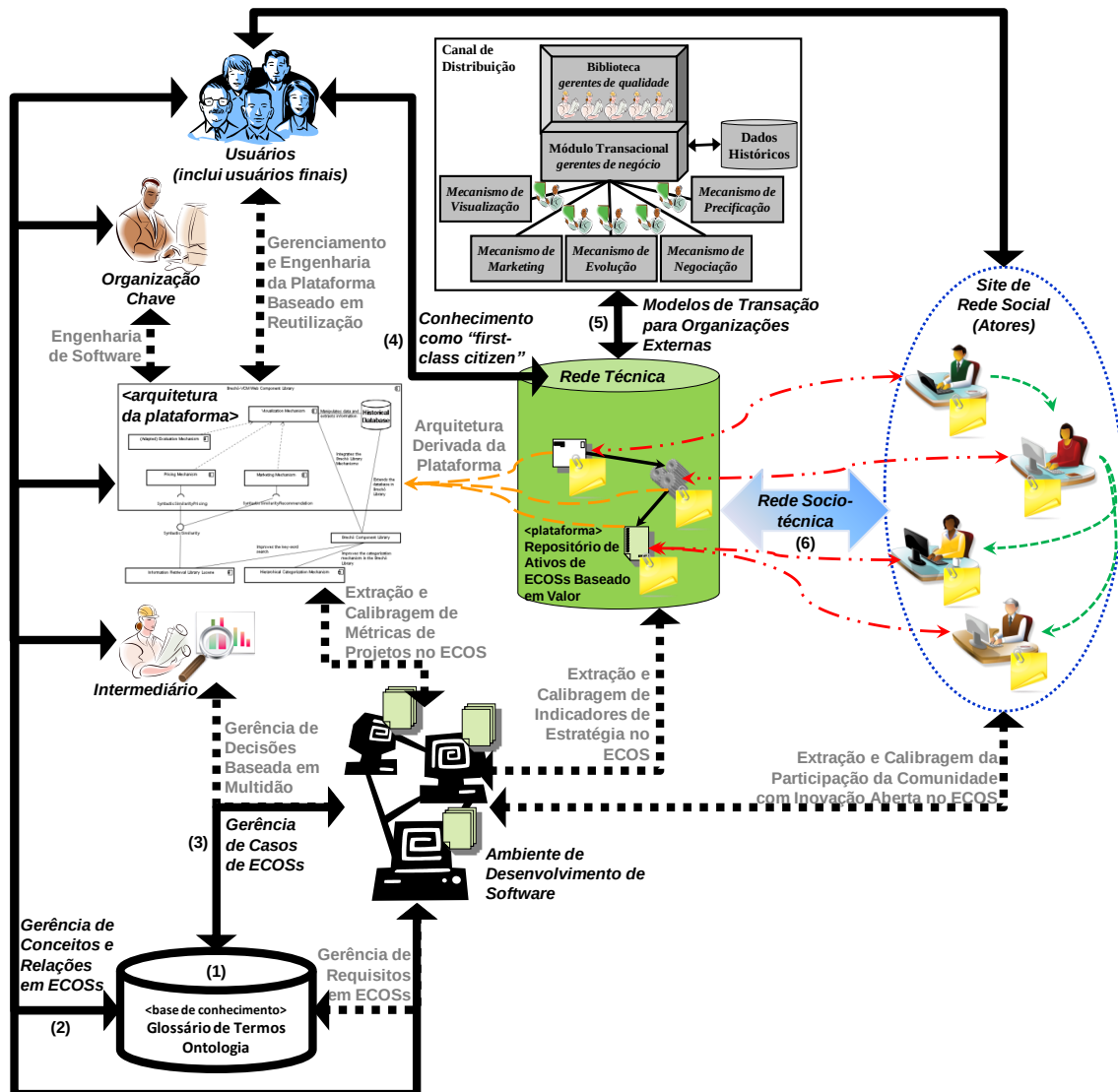


Figura 4.20 – Infraestrutura de apoio à abordagem ReuseSEEM.
Fonte: (SANTOS & WERNER, 2012b)

Conforme discutido na proposta de ECOSs baseada em reutilização, o conhecimento (i.e., artefatos) tem *status* equivalente a um ator, conforme SEICHTER *et al.* (2010) chama de “*first class citizens*”, o que representa uma fraqueza da representação por RPS. Partindo destes elementos de ECOSs, três ferramentas contemplariam as três relações na área de ES e implementariam as fases de *definição* (AD de ECOSs) e *modelagem* (projeto de domínio de ECOSs):

- **Ferramenta #1:** *base de conhecimento de ECOSs*, uma ferramenta para apoiar a gerência de conceitos e relações em ECOSs, bem como a sua instanciação. Para isso, utiliza conceitos de ED e de DBC (WERNER & BRAGA, 2005). Baseia-se na coleta, análise e visualização de dados acerca da saúde de ECOS sob a perspectiva de seus **artefatos**, tais como (SANTOS & WERNER, 2012b):
 - *métricas de projeto*: modularidade e transparência de artefatos pertencentes a projetos que envolvem os atores de diferentes ECOSs;
 - *indicadores de estratégia*: sustentabilidade e diversidade sob níveis de escopo da visão interna e externa do ECOS no desenvolvimento de software globalizado;
 - *participação da comunidade*: potencial de ambientes colaborativos apoiados por redes sociais e seus impactos na indústria de software.
- **Ferramenta #2:** *repositório de ativos de ECOSs baseado em valor*, uma extensão de um repositório de DBC para apoiar o tratamento de elementos de negócio em ECOSs. Para isso, utiliza cadeias de valor e cadeias de resultados (BOEHM & JAIN, 2007). Baseia-se na coleta, análise e visualização de dados acerca da saúde de ECOS sob a perspectiva de suas **informações** (elementos de negócio), tais como (SANTOS, 2010):
 - *visualização* de movimentos de artefatos das plataformas em ambientes e mercados, e.g., ofertas, necessidades, novos nichos etc.;
 - *marketing* realizado sobre os ECOSs, bem como sua análise, e.g., organizações, plataformas e/ou produtos *outliers* (que aumentam ou reduzem o valor do ECOS), recomendação de parceiros/membros etc.;
 - *avaliação* dos ECOSs, das plataformas, dos produtos e dos atores, considerando critérios como experiência e reputação para a construção da credibilidade dos envolvidos;
 - *negociação* de requisitos e de artefatos entre os atores do ECOS, e.g., orquestração da customização solicitada por organização consumidora de componentes para uma organização externa no ECOS;
 - *precificação* de artefatos de software em plataformas de ECOSs, considerando uma visão baseada em valor para contemplar as suas diferentes facetas, e.g., tempo, riscos, benefícios, oportunidades etc.
- **Ferramenta #3:** *rede sóciotécnica*, uma extensão de um *site* de rede social para apoiar o estabelecimento, combinação, controle de dependências e evolução de artefatos e papéis nos ECOS, mas também tratar elementos de comunidade (Seção 4.3.3). Isso resulta em redes baseadas em características comuns e perfis de usuários para gerenciar as relações do tipo ator-conhecimento, e.g., “amizade” em *sites* de redes sociais, dado que o conhecimento tem *status* equiparado ao de um ator. Para isso, utiliza teoria de grafos e de ator-rede (LATOUR, 2000), incluindo a RPS (BOUCHARAS *et al.*, 2009). Baseia-se na coleta, análise e visualização de dados acerca da saúde de ECOS sob a perspectiva de seus **recursos**, tais como:
 - *redes técnicas*, estruturadas em torno dos repositórios de artefatos da plataforma do ECOS para visualizar, e.g., dependências entre artefatos (SANTANA & WERNER, 2013);

- *redes socio-técnicas*, derivadas das relações entre atores e artefatos do ECOS para visualizar, e.g., mapas de reutilização³⁸ entre os envolvidos nas transações dos ECOSs (WERNER *et al.*, 2009);
- *redes sociais*, instanciadas a partir de perfis dos atores dos ECOSs disponíveis em *sites* na Internet para visualizar, e.g., tabelas de competências e habilidades para alocação de recursos humanos (SANTOS *et al.*, 2012a; 2013b).

Para implementar a fase de *análise* e refinar o *framework ReuseECOS*, outra ferramenta é necessária para apoiar a gerência de casos de ECOSs e, assim, os intermediários no processo de abertura dos ECOSs. Permite-se assim extrair informação e gerar conhecimento a partir do nível de escopo da visão externa de um ECOS, por haver uma base de casos de ECOSs para mineração. Ressalta-se que a base de conhecimento e o repositório de ativos de ECOSs baseado em valor devem estar integrados, para que troquem dados da indústria de software globalizada. Ao mesmo tempo, as relações entre as três dimensões são apoiadas por este repositório:

A proposta de solução para o *framework ReuseECOS* permite a construção de contribuições de pesquisa e prática em ECOSs como protótipos que estendam ferramentas utilizadas na ES tradicional e métodos adaptados de outras áreas do conhecimento. Por exemplo, o repositório de ativos de ECOSs pode ser apoiado pela plataforma Brechó-VCM (Seção 2.5). Para a rede sociotécnica, um *site* de rede social como *Facebook* pode ser integrado à plataforma Brechó-VCM e a repositórios de ES (e.g., controle de versões, controle de solicitações, gerência de projetos etc.) visando relacionar a rede social e a rede técnica. Finalmente, os requisitos definidos na Seção 4.3.4 foram confrontados com infraestrutura de apoio à abordagem *ReuseSEEM*. A análise permitiu selecionar um conjunto de seis relações entre as suas partes, mantidas em linha preta contínua e numeradas na Figura 4.20. A seguir, elas são discutidas:

- (1) *Repositório de ECOS*, diretamente relacionado aos requisitos R2 e R5, por tratar da construção da base de conhecimento e do repositório de ativos de ECOSs;

³⁸Um mapa de reutilização consiste em um mecanismo que permite aos produtores conhecerem seus consumidores e necessidades, criando um canal de contato entre as partes. Este canal viabiliza a notificação de atualizações, demandas de modificações, correção de defeitos e novas funcionalidades, contribuindo para o amadurecimento do próprio componente (LOPES *et al.*, 2006). Por sua vez, este gerenciamento incentiva a criação de uma comunidade de usuários, facilitando a interação, o suporte e uma classificação da reputação dos produtores quanto à qualidade dos componentes e dos serviços oferecidos. Para estabelecer esse vínculo, é necessário instanciar acordos entre produtores e consumidores a partir da seleção de licenças associadas ao componente, incluindo questões de propriedade intelectual e de adaptações, responsabilidades, suporte, tarifaç  o, garantias etc. (SANTOS, 2010).

- (2) *Gerência de Conceitos e Relações em ECOSs*, diretamente relacionada ao requisito *R1*, *R3* e *R4*, por tratar da organização do glossário de termos e da identificação de atores, artefatos, papéis, responsabilidades e relacionamentos;
- (3) *Gerência de Casos de ECOS*: diretamente relacionada aos requisitos *R7*, por tratar o mecanismo de extração de subdomínios de ECOSs e seus impactos na base de conhecimento;
- (4) *Conhecimento como “First-Class Citizen”*: diretamente relacionado aos requisitos *R5* e *R6*, por tratar artefatos da mesma forma que os atores nas redes sociotécnicas de ECOSs;
- (5) *Modelos de Transação para Organizações Externas*: diretamente relacionados aos requisitos *R4* e *R5*, por tratar a implementação de mecanismos para incorporar atributos de negócio nos ECOSs;
- (6) *Rede Sociotécnica*: diretamente relacionado aos requisitos *R4* e *R5*, por tratar a implementação de mecanismos para incorporar atributos de comunidade nos ECOSs;

4.3.6. Projeto e Implementação

A fim de modularizar o ambiente de DPE, os seguintes módulos foram identificados na arquitetura exibida na Figura 1.2:

- **Módulo #1: Suporte à AD de ECOSs**: atende aos requisitos *R1*, *R2*, *R3* e *R8*, consistindo na construção da base de conhecimento, composta por quatro subsistemas: (1) um glossário de termos de ECOSs refinado partir do *framework ReuseECOS* por meio de uma pesquisa de opinião, e posteriormente catalogado eletronicamente; (2) um modelo de domínio de ECOSs implementado em uma linguagem gráfica apropriada; e (3) um protótipo de ferramenta *Web*, do tipo *wizard*, para apoiar o processo de instanciação dos conceitos de ECOSs;
- **Módulo #2: Suporte ao projeto de domínio e de subdomínios de ECOSs**: atende aos requisitos *R4*, *R6* e *R7*, consistindo na identificação de elementos e suas conexões em ECOS, bem como a adaptação de modelos para a sua representação por meio de um protótipo de

ferramenta *Web*. O primeiro subsistema extrai atores e artefatos, bem como papéis, responsabilidades e relacionamentos, a partir do glossário de termos (*extrator de elementos e conexões*) para gerar: (1) um modelo de representação estendido da RPS (JANSEN, 2007; BOUCHARAS *et al.*, 2009) e, em seguida, (2) um modelo de rede sociotécnica de ECOSs, estendido da rede sócio-técnica (MAGDALENO, 2010). O segundo subsistema permite a configuração ou recorte de casos de ECOSs a partir da seleção de elementos e conexões de interesse do ponto de vista de um interessado (*extrator de subdomínios de ECOSs*).

- **Módulo #3: Extensão de um Repositório de Reutilização para criar o Repositório de Ativos de ECOSs:** atende ao requisito *R5*, consistindo na adaptação de um repositório de componentes para contemplar ativos de ECOS ao tratar aplicações (produtos e serviços) que compõem ou derivam da plataforma de ECOS. Deve-se implementar mecanismos de negócio e de comunidade que devem ser acoplados para facilitar a percepção de valor pelos interessados. No caso do senso de negócio, estes aspectos correspondem aos seguintes mecanismos, explorados na dissertação de mestrado que foi evoluída por esta proposta de tese (SANTOS, 2010): precificação, *marketing*, avaliação, visualização e negociação (Seção 2.5). Por sua vez, no senso de comunidade, estes aspectos correspondem aos seguintes mecanismos, explorados em um trabalho desenvolvido na COPPE/UFRJ (SANTOS *et al.*, 2012a; 2013b) e estendido de (SEICHTER *et al.*, 2010): oito elementos do *site* de rede social e seis indicadores sociais enunciados na Seção 3.3.2.

Para implementar o Módulo #1 do ambiente *ReuseSEEM*, algumas alternativas tecnológicas foram exploradas: (i) para o glossário de termos, foi analisada a tecnologia *Wiki*, pelas facilidades de divulgação para a comunidade e de entendimento de conceitos, e pela experiência do pesquisador (SANTOS *et al.*, 2010c; BORGES *et al.*, 2011); e (ii) para o modelo conceitual, foram identificadas alternativas, como ontologias³⁹, e.g., ambiente de gerência de ontologias *Protégé*, linguagem para

³⁹Outras opções também foram avaliadas. Inicialmente, foi escolhido o modelo de características da ED, mas se percebeu que o seu uso será mais apropriado com os desdobramentos desta pesquisa, uma vez obtidos os resultados desta proposta de tese. Apesar das inúmeras vantagens das ontologias e suas aplicações para ED previamente publicadas na literatura, mais recentemente, os diagramas da UML

representação formal *OWL* e linguagens para representação visual *LINGO* e *OntoUML*, utilizadas recentemente por duas dissertações de mestrado no LENS para o propósito de construção e instanciação de ontologias a partir de conceitos de domínio (LOPES, 2010; SANTO, 2012). Para o Módulo #2, pretende-se desenvolver um sistema *Web* utilizando a linguagem de programação Java para automatizar os subsistemas da atividade de projeto de domínio de ECOSs, que deve ter interface com os Módulos #1 e #3.

Dado que esta proposta de tese evolui uma dissertação de mestrado no contexto do Projeto Brechó (SANTOS, 2010), cujo propósito envolve explorar o senso de negócio em bibliotecas de componentes pela abordagem Brechó-VCM, o Módulo #3 deve ser implementado como uma extensão desta biblioteca, chamada Brechó-EcoSys (SANTOS & WERNER, 2011c). O primeiro passo foi implementar o mecanismo de negociação e estender o mecanismo de avaliação para contemplar também produtores e consumidores, e não apenas componentes, realizado em 2012 (SANTOS *et al.*, 2013c). Para o senso de comunidade, está sendo investigado o ambiente *Facebook Developers*, pois ele atende aos atributos de comunidade (Seção 4.3.1) e pode ser integrado com a biblioteca (SANTOS *et al.*, 2012a). No entanto, este módulo depende dos resultados obtidos no Módulo #1, a fim de incorporar elementos e suas conexões em ECOSs.

4.4 Trabalhos Relacionados

Apesar do número crescente de publicações relacionadas a ECOSs e do interesse da comunidade de ES em sua aplicabilidade industrial, os quatro estudos de revisão da literatura realizados apontam que o tema carece de pesquisas sobre modelos analíticos e sobre a execução de estudos de caso na indústria, possuindo um número elevado de publicações do tipo relatos e/ou casos de ECOSs (SANTOS & WERNER, 2010a; BARBOSA & ALVES, 2011; HANSEN & DYBA, 2012; MANIKAS & HANSEN, 2013). Conforme discutido, um complicador para as contribuições de pesquisa baseadas em experimentação está no fato de que o conhecimento dentro da comunidade varia amplamente e mesmo os conceitos de ECOS ainda são vagos e diversos, sendo muitas vezes utilizado de maneira *ad hoc* pela indústria de software. Este contexto motivou a abordagem *ReuseSEEM*, pois a contribuição desta proposta de tese surge pela carência de trabalhos formalizados para integrar definição, modelagem e análise de ECOSs. A

passaram a representar uma possibilidade para representar modelos conceituais de ECOSs. Após o refinamento dos conceitos de ECOSs e a construção do glossário de termos, as duas opções serão avaliadas novamente.

seguir, são apresentadas algumas abordagens que lidam minimamente com essas três questões, mesmo que destaquem apenas uma delas.

Motivados pelo fato de que as organizações têm problemas em distinguir os ECOSs onde atuam e em obter vantagem competitiva deles, BOUCHARAS *et al.* (2009) propõem uma abordagem formal para modelagem do contexto de implantação do produto de software e dos componentes da RPS, denominada *Software Ecosystem Modeling* (SEM). O objetivo de aplicar a abordagem SEM é permitir que as organizações se comuniquem através de suas ligações na RPS, para que possam propor e avaliar teorias a respeito de relações frágeis em seus modelos de negócio e antecipar os próximos desafios do ECOS. A abordagem se baseia nos níveis de escopo de ECOSs de JANSEN *et al.* (2009a), enfocando a visão interna do ECOS, e busca melhorar a visão de diversos tipos de atores (e.g., gerentes, engenheiros etc.).

Os autores combinam alguns aspectos de métodos de modelagem existentes com o desenvolvimento de novos *insights*, a fim de posicionar o produto de software na rede de produção. A abordagem SEM possui uma notação visual e está sedimentada por um conjunto de regras e relacionamentos (lógica de predicados), além de um meta-modelo (diagrama de classes da UML). Apresenta-se um estudo de caso na *CubicEyes*, uma organização focada no desenvolvimento de soluções *Web*. Entretanto, os autores não apresentam uma ferramenta para apoiar a abordagem, embora tenham utilizado uma ferramenta existente para a atividade de modelagem (*MetaEdit*). Ainda, a abordagem SEM foca na visão interna do ECOS, ou seja, não explora a variabilidade do domínio de ECOSs. Assim, não oferece um mecanismo para extrair subdomínios do ECOS a partir da visão externa (e.g., entender os papéis distintos que o mesmo ator pode nos ECOSs).

BERK *et al.* (2010) relatam as dificuldades das organizações, parceiros e desenvolvedores em conseguir *insights* na indústria de software baseada em ECOSs, o que impede uma avaliação de desempenho frente às suas ações individuais. Nesse sentido, propuseram um modelo que descreve um conjunto de características chave de ECOSs para permitir que as organizações avaliem o *status* de um ECOS, ou obtenham vantagens competitivas no sucesso de decisões, denominado *SECO Strategy Assessment Model* (SECO-SAM). Para criar o modelo, os autores exploraram a literatura de ecossistemas de negócio, mapeando conceitos e indicadores da área de Administração para a ES. Este modelo foi avaliado em um estudo de caso na *Open Design Alliance* (ODA), uma organização que promove padrões abertos em formatos CAD, particularmente *dwg*. O foco do estudo foi na plataforma em si, que consiste em várias

bibliotecas de software utilizadas pelos membros da ODA para a construção de aplicações gráficas. Algumas questões foram identificadas, e.g., “qual a interferência da ODA nos *insights* a respeito do ECOS?” e “que características quantitativas foram observadas no ECOS?”, e incluiu também uma avaliação da própria aplicação do SECO-SAM. Entre os principais problemas, não há garantias quanto à completude ou generalidade de aplicação do modelo devido à execução de um estudo de caso apenas; o modelo é de natureza ampla e trata de diversas características igualmente ponderadas, algumas delas específicas dos ecossistemas de negócio, o que pode desviar o foco dos ECOSs; e se baseia apenas no preenchimento e sumarização de questionários.

HUNINK *et al.* (2010) discutem a falta de métodos para apoiar o processo de criação de uma taxonomia de indústria dentro de um domínio específico, o que dificulta a existência de um diretório de pesquisa compartilhado por organizações interessadas. Para isso, desenvolvem um método para construir taxonomias de indústria baseado nas teorias da ciência comportamental e da ciência de projeto, denominado *Industry Taxonomy Engineering Method* (ITAXEM). O uso da primeira teoria busca tornar claro e/ou prever a conduta organizacional, ao passo que o uso da segunda visa estender as capacidades organizacionais ao criar novos conhecimentos e artefatos. O objetivo do método ITAXEM é servir de base para o desenvolvimento de modelos de dados e propiciar *insights* para os envolvidos na indústria de software.

Uma análise dos resultados obtidos em um estudo de caso realizado com um ECOS europeu gerou com sucesso a *European Software Industry Taxonomy* (EUSOIT), mas apontou alguns desafios, fraquezas e limitações do método. Por exemplo, (i) *qualidade das fontes*: um número limitado de taxonomias foi coletado e selecionado, e.g., *Amadeus*, *ITEuropa*, *Truffle 100 2008*, e bancos de dados como *Capterra* e *Software Network*; e (ii) *identificação dos envolvidos*: o método não define explicitamente como os envolvidos precisam ser identificados entre fornecedores, organizações governamentais e pesquisadores. Dessa forma, sua contribuição para o domínio de ECOSs não é clara, exceto pelo processo de criar uma taxonomia da indústria de software baseada em ECOSs, ou seja, o método aparenta somente utilizar o contexto do ECOSs para sua execução, sem ferramental de apoio.

Por outro lado, PETTERSSON *et al.* (2010) identificaram um *gap* em ECOSs, referente à necessidade da modelagem de processo mais precisa para entender o papel do software e da reutilização com a aproximação do usuário final como envolvido direto no desenvolvimento de software. Os autores utilizaram sua experiência para elaborar

um modelo de referência inicial para o domínio de aprendizagem móvel e para criar um método para análise de ECOSs específicos de domínio. A ideia é criar o domínio de *Mobile Learning Ecosystems* (MLES), resultante da combinação dos conceitos de três domínios: aprendizagem, *mobile* e ECOSs. Para isso, os autores empregaram as estratégias de avaliação de agregação de modelo de domínio via *walkthroughs* para criar um modelo de referência inicial, que contou ainda com a identificação de componentes de processo em um cenário no domínio de aprendizagem móvel. Ou seja, impactos da participação dos usuários finais na demanda e desenvolvimento de aplicações para plataformas de ECOSs. Estes ensaios apoiaram a geração de um modelo refinado, com base no *Software Process Engineering Meta-model* (SPEM).

O modelo refinado foi instanciado e verificado por 7 passos estabelecidos pelos autores, com relação aos componentes de processo identificados *a priori*: definir escopo, definir cenários de uso, realizar ED, combinar os três domínios, avaliar o modelo de domínio gerado, refinar este modelo e anotar conceitos adicionais ao modelo. A abordagem proposta pelos autores não foi completamente avaliada e foca em processo de software, como mostra o desejo dos autores em explorar auto adaptação de processo em ECOSs. Apesar de usarem conceitos de reutilização, o domínio de ECOSs não foi identificado ou modelado por estudos experimentais. Além disso, parece ter sido aplicado estritamente para inserir abordagens mais formais de ES (e.g., SPEM) sobre o processo de desenvolvimento de aplicações para aprendizagem móvel. Como nos outros trabalhos discutidos, nenhuma infraestrutura tecnológica foi apresentada.

Mais direcionados para compreender modelos de associação, ANGEREN *et al.* (2011) afirmam que os orquestradores de grandes ECOSs têm que lidar com *clusters* de organizações, embora não esteja claro como este ator funciona na prática. Os autores apresentam uma visão geral sobre modelos de associação de parceria e de sociedade em ecossistemas, discutindo os papéis assumidos pelos participantes dos ECOSs, assim como a dimensão e os benefícios, requisitos e custos envolvidos. Em seguida, os autores propõem uma visão conceitual de um modelo de associação em ECOSs, utilizando um diagrama de classes da UML. Uma avaliação deste modelo é realizada em três estudos de caso, que consideram os ECOSs SAP, ODA e Eclipse. Os ECOSs mencionados são comparados por meio de uma tabela de classificação, que enumera um conjunto de características de modelos de associação derivadas do modelo conceitual proposto. Apesar de contemplar elementos de ECOSs, o trabalho também não se preocupou com a

definição de domínio e permanece no nível de escopo da visão interna de ECOSs, de modo que cada ECOS deve ser analisado separadamente e sem suporte computacional.

Preocupados com a transição das estruturas e relacionamentos de uma indústria de software tradicional para um ecossistema, YU & DENG (2011) reforçam os impactos que o cenário de ECOSs pode gerar sobre negócios e decisões técnicas. Os autores realizam a tentativa inicial de utilizar uma abordagem de modelagem estratégica para apoiar a compreensão dos ECOSs, baseada no *framework* de modelagem i*. Os modelos i* descrevem dependências estratégicas entre organizações, desenvolvedores externos e usuários finais a fim de propiciar *insights* sobre alternativas para alcançar as metas estratégicas de cada ator. Utilizam, ainda, os conceitos de objetivo, sub-objetivo, tarefa e recurso. Para sua construção, os autores comparam os relacionamentos entre vendedor-comprador na cadeia de produção de software tradicional e na cadeia aberta dos ecossistemas, utilizando um exemplo relacionado à perspectiva de uma organização de software para dispositivos móveis fictícia.

Uma discussão sobre a dinâmica da mudança de uma cadeia de produção fechada para outra aberta permite observar que a complexidade dos relacionamentos multilaterais pode ser observada em dois caminhos: (i) o número maior de dependências, principalmente aquelas relativas aos sub-objetivos associados, e (ii) maior número de dependências recíprocas entre os atores do ECOS. Apesar de representar uma nova iniciativa de modelagem e análise de ECOSs, a abordagem não foi avaliada em um estudo de caso e contempla apenas a comparação dos aspectos de produto nos relacionamentos vendedor-comprador. Assim, desconsidera outros elementos, e.g., parceiros e serviços relacionados, ou uma base de conceitos para evitar isso. O modelo apresentado foca apenas no nível de escopo da visão interna do ECOS, não tratando outros níveis de abstração do ECOSs, sem ferramenta de apoio.

Por fim, do ponto de vista de reutilização, MCGREGOR (2010; 2012) investiga como informações, artefatos, clientes, moedas e produtos se movimentam entre as organizações, especialmente aquelas que implementam LPSs. No contexto de ECOSs, o autor propõe um método para modelar e analisar *ecossistemas de negócio de software*, explorando suas transações, o que produz três visões de ecossistema (i.e., negócio, software e inovação) e identifica ferramentas para cada uma delas, chamado *Strategic Ecosystem Analysis Method* (STREAM). Na visão de negócio, foca-se nas organizações, informações associadas e relacionamentos, representados por um modelo de ecossistema similar a um diagrama de rede, que visa identificar os agentes de um

ECOS. Na visão de software, considera-se o produto e um conjunto de modelos, testes e dados associados, com foco na arquitetura. Por sua vez, na visão de inovação, os elementos das outras visões são mostrados a partir de um esquema de classificação em quatro partes: produto, processo, experiência do cliente e modelo de negócio.

O método também possui um modelo conceitual para ECOSs (perfil da UML). O autor apresenta um estudo de caso no ECOS Eclipse e conclui que este ecossistema possui um magnetismo forte para desenvolvedores externos, é saudável em quase todas as medidas da saúde (produtividades, robustez e criação de nicho) e tem alta capacidade de inovação, apesar da medida clássica de LoC estar diminuindo anualmente. Conforme observado, o método STREAM não se preocupa com a construção de uma base de conhecimento para ECOSs, mas em avaliar um ECOS através de um processo, voltando-se mais para um ambiente de DCE (Seção 1.4). Além disso, os conceitos de ECOSs utilizados pelo método derivam fortemente de ecossistemas de negócio, até pelo foco em consultoria. Não existe um mapeamento entre os elementos de ECOSs em diferentes níveis, nem um ferramental para automatizar o método.

4.5 Considerações Finais

O processo de desenvolvimento de software cada vez mais se aproxima dos usuários finais e da sociedade, o que impacta a área de ES (MESSERSCHIMITT & SZYPERSKI, 2003). Com o contexto dos ECOSs, o enfoque passa do desenvolvimento de software tradicional para a evolução de plataformas e de seus produtos relacionados, o que envolve repensar os obstáculos da reutilização e as interferências das redes de atores e de conhecimento dos ECOSs (WERNER *et al.*, 2012). No entanto, o estado da arte do tema ainda impede o seu pleno estabelecimento como área da ES (SANTOS *et al.*, 2012c). As pesquisas deveriam prover um conjunto mais coeso de classificação dos ECOS e de seus atores, não apenas com o viés de ecossistemas de negócio, mas tratando a hibridização entre os diferentes tipos de ECOSs e seus resultantes (BARBOSA, 2012): reconhecimento de pares, senso de comunidade, ideia de propriedade, aprendizagem, recompensa, engajamento com novas tecnologias e intenções altruístas.

Dado o problema da compreensão de conceitos e relações em ECOSs diante da abertura de suas plataformas na indústria de software, este capítulo apresentou, como proposta de solução, uma abordagem baseada em reutilização para dar suporte ao E&G de ECOSs, denominada *ReuseSEEM*. A base da solução está no uso do conceito de

domínio para criar um processo de definição, modelagem e análise de ECOSs. Para isso, uma base de conhecimento inicial foi definida com base nas iniciativas da comunidade de pesquisa no tema, como resultado de uma agregação de trabalhos publicados nas edições do IWSECO (2009-2012). Começando por três dimensões, o *framework ReuseECOS* foi utilizado na análise de casos de ECOSs do LENS/Reuse. Os resultados permitiram a proposta da quarta dimensão de E&G e de uma infraestrutura de apoio. Além disso, a abordagem incorpora elementos dos sentidos de negócio e de comunidade para explorar ECOSs como caminho para o futuro da reutilização.

A proposta é concretizada por meio de um ambiente que contempla as três primeiras fases: (i) definição de domínio de ECOS pela utilização de um glossário de termos e de um modelo conceitual na base de conhecimento; (ii) construção de um ambiente de DPE, que envolve modelagem das RPSs e, em seguida, das redes sociotécnicas de ECOSs; e (iii) uso de (ii) para realizar análises de ECOSs a partir da extração de seus subdomínios. Como principais contribuições desta proposta de tese, espera-se: (i) melhorar a compreensão e percepção do(s) ECOS(s) na indústria de software globalizada ao tratar o domínio de ECOSs; e (ii) investigar um novo caminho de pesquisa para explorar a reutilização interorganizacional.

Até o presente momento, a principal contribuição da pesquisa foi a organização do *framework ReuseECOS* para perceber oportunidades que possam contribuir para o tema ECOSs sob o olhar da ES, de maneira a não dissociá-la de suas questões econômicas e sociais. Isso foi fundamental para confirmar as duas questões que embasaram esta proposta de tese: (1) *definição e modelagem de ECOSs*, expandindo a visão até então orientada pelos ecossistemas de negócio para incorporar visão da tecnologia e da sociedade, e (2) *análise de ECOSs*, gerenciando e combinando casos de ECOSs para analisar o caso de uma organização chave a partir de dados de repositórios de ativos e de projetos de software, à luz da interferência da comunidade.

A Tabela 4.4 mostra uma sumarização da análise dos trabalhos relacionados realizada na Seção 4.4, que apresenta iniciativas para compreensão de ECOSs, expostas aos requisitos da abordagem proposta (Seção 4.3.4). As iniciativas tratam minimamente a definição, modelagem e análise de ECOSs, mesmo que destaquem apenas uma delas. Observou-se que estes trabalhos não mostram ligações entre um modelo conceitual de ECOSs e um processo de análise, o que impacta a sua compreensão ao desconsiderar um glossário de termos. A grande maioria dos trabalhos não explora técnicas de reutilização ou mesmo uma interface com ambientes de redes sociais, exceto um deles,

que aplica estes conceitos para o domínio de MLES (PETTERSSON *et al.*, 2010), e outro, que explora ferramentas de desenvolvimento para apoiar ECOSs (MCGREGOR, 2010; 2012). Em todos os trabalhos, existe uma preocupação em identificar ou refinar os conceitos de domínio de ECOSs, embora não se preocupem em criar ou buscar uma base de conhecimento que contemple um conjunto consensual mínimo de conceitos importantes para os ECOSs. Isso justifica o fato de modelarem e analisarem os ECOSs sob o nível de escopo da visão interna e, conseqüentemente, de não se preocuparem em prover um repositório para gerenciar elementos de ECOSs e a sua visão externa.

Tabela 4.4 – Resumo dos trabalhos relacionados em relação aos requisitos da abordagem proposta, onde  = atende;  = atende parcialmente;  = não atende

	BOUCHARAS <i>et al.</i> (2009)	BERK <i>et al.</i> (2010)	HUNINK <i>et al.</i> (2010)	PETTERSSON <i>et al.</i> (2010)	ANGEREN <i>et al.</i> (2011)	YU & DENG (2011)	MCGREGOR (2010; 2012)
R1. Identifica e refina conceitos de domínio de ECOSs							
R2. Fornece um glossário de termos para ECOSs							
R3. Utiliza modelo conceitual para representar ECOSs							
R4. Identifica atores e artefatos, bem como relacionamentos							
R5. Utiliza repositórios de reutilização para apoiar ECOSs							
R5. Utiliza site de rede social para apoiar ECOSs							
R6. Oferece apoio para processos de ED de ECOSs							
R7. Oferece apoio para tratar subdomínios em ECOSs							

Capítulo 5 – Conclusão

5.1 Contribuições

Esta proposta de tese tem como objetivo construir uma abordagem baseada em reutilização para dar suporte a engenharia e gerenciamento (E&G) de Ecossistemas de Software (ECOSs), explorando, para isso, o conceito de domínio em um ambiente de desenvolvimento para ECOSs. Entre as suas principais contribuições, destacam-se:

- No contexto do problema:
 - Melhorar a compreensão de ECOSs na indústria de software globalizada ao explorar o domínio *de* ECOSs durante as etapas de definição e modelagem;
 - Apoiar os envolvidos na indústria de software a terem uma base para realizar a análise de ECOSs ao expor informações disponibilizadas em um repositório de casos;
 - Enveredar por um novo caminho para tratar a reutilização interorganizacional ao explorar o senso de negócio e o senso de comunidade na Engenharia de Software (ES);
 - Refletir sobre os desafios da ES e seus impactos na indústria nos próximos anos ao investigar questões de natureza econômica e social.
- No contexto da solução:
 - Construir uma base de conhecimento para o domínio de ECOSs com o intuito de servir de referência para a comunidade de pesquisa em ECOSs;
 - Identificar os principais elementos e modelos para representação de ECOSs, visando organizar um repositório de ativos de ECOSs orientado por técnicas de reutilização;
 - Construir um ambiente para definição, modelagem e análise de ECOSs composto por um processo de Engenharia de Domínio (ED) de ECOSs, bem como uma infraestrutura de apoio às suas atividades;

- Planejar e executar estudos experimentais, a fim de avaliar as partes que compõem a abordagem proposta e representar iniciativas baseadas em evidência para ECOSs.
- No contexto das comunidades acadêmicas relacionadas:
 - Elaborar uma linha do tempo com o objetivo de analisar a trajetória da reutilização e entender como os ECOSs podem ser apontados como um caminho potencial para o futuro da pesquisa na área;
 - Realizar uma revisão da literatura acerca das principais definições, estruturas, características, benefícios e dificuldades, sinais de teorização, impactos e grupos e áreas de ECOSs a fim de destacar a novidade do tema;
 - Organizar um *framework* para E&G de ECOSs na forma de uma agregação das iniciativas especificamente relacionadas ao tema, visando explicitar as afinidades entre os pesquisadores e profissionais interessados;
 - Divulgar os resultados obtidos até o momento, bem como aqueles que forem gerados, por meio de publicações científicas no tema.
- No contexto do Laboratório de Reutilização de Software da COPPE/UFRJ (LENS/Reuse):
 - Desenvolver um sistema *Web* para gerenciar o glossário de termos de domínio de ECOS (e.g., *Wiki*);
 - Desenvolver um sistema *Web* para instanciar um modelo conceitual de ECOSs (e.g., ontologia ou diagramas da *Unified Modelling Language* – UML);
 - Desenvolver um sistema *Web* para viabilizar a ED de ECOSs, a fim de implementar mecanismos de instanciação, modelagem e extração de subdomínios de ECOSs;
 - Estender a biblioteca Brechó para construir o repositório de ativos de ECOSs ao implementar mecanismos para tratar o senso de negócio em ECOSs e integrar um *site* de rede social para tratar o senso de comunidade;

- Expandir e solidificar uma nova frente de pesquisa no grupo, iniciada em meados de 2008, cujas origens se relacionam diretamente com a dissertação de mestrado do autor.

5.2 Metodologia da Pesquisa

A metodologia utilizada para a execução deste trabalho de pesquisa inclui as atividades realizadas antes desta proposta de tese, para apoiar a sua elaboração, e as atividades a serem conduzidas após a defesa do exame de qualificação. Dessa forma, pode-se refinar o desenvolvimento da pesquisa proposta e melhor nortear as direções para a sua conclusão.

5.2.1. Atividades Realizadas

As atividades realizadas antes desta proposta de tese e que apoiaram a sua elaboração são relacionadas a seguir:

- *Pesquisa bibliográfica geral sobre o tema ECOSs*, para identificar terminologia, definições, estruturas de classificação, características, benefícios e dificuldades, sinais de teorização, impactos na ES e grupos e tópicos de pesquisa;
- *Pesquisa bibliográfica geral sobre técnicas e ambientes de Reutilização de Software*, para apoiar a abordagem proposta, considerando o Desenvolvimento Baseado em Componentes (DBC): componentes e sua classificação, tipos de repositórios, arquiteturas, estilos e padrões e aspectos e mercado, biblioteca Brechó e processo Gerência de Reutilização (GRU). Mais recentemente, ED: domínio, linha de produto de software, ciclo de desenvolvimento, variabilidade, modelos de representação e ontologias;
- *Construção do framework para E&G de ECOSs*, para agrupar as iniciativas da comunidade de pesquisa no tema por meio de uma agregação dos trabalhos publicados nas primeiras edições do IWSECO (2009-2012), denominado *framework “3+1” ReuseECOS*. O *framework* analisa casos de ECOSs sob as dimensões técnica, transacional e social, integradas pela dimensão de E&G, e será a base para a definição e modelagem de ECOSs;

- *Elaboração de uma linha do tempo para a Reutilização de Software*, para mostrar as suas quatro gerações e posicionar efetivamente o conceito de ECOSs, que vem sendo apontado pelos pesquisadores e profissionais como um potencial caminho para a área;
- *Evolução da biblioteca Brechó*, para implementar o mecanismo de negociação e uma extensão do mecanismo de avaliação, visando concluir os cinco aspectos do senso de negócio, além de estudos sobre os dois mecanismos referentes aos aspectos do senso de comunidade explorando a ferramenta *Facebook Developers*;
- *Outras atividades*, como interações com outros grupos de pesquisa no Brasil e no exterior (e.g., CIn/UFPE, Utrecht University, Linha de BD do PESC/COPPE/UFRJ); e participações em diferentes eventos para permitir o contato com comunidades distintas: (1) ECOSs, Ecossistemas Digitais e Negócios de Software; (2) ES, Reutilização, Desenvolvimento Distribuído, Qualidade e Arquitetura; (3) Sistemas de Informação e Análise de Redes Sociais; e (4) Planejamento Urbano e Regional.

5.2.2. Atividades Previstas

Este trabalho de pesquisa deverá continuar com o seguinte conjunto de atividades, após a defesa do exame de qualificação:

- *Construir a infraestrutura para análise de domínio de ECOSs* (base de conhecimento), o que requer:
 1. refinar os principais conceitos de ECOSs;
 2. planejar e executar um estudo experimental do tipo pesquisa de opinião com os especialistas em ECOSs, selecionados inicialmente a partir dos comitês de programa do IWSECO e da *International Conference on Software Business*, visando avaliar esses conceitos;
 3. desenvolver o glossário de termos para organizar os conceitos e respectivos significados usando *Wiki*;
 4. escolher uma tecnologia para construir um modelo conceitual a fim de representar o domínio de ECOSs;

5. implementar o modelo conceitual para representar as relações entre os conceitos, propriedades, funções, atributos e restrições;
 6. desenvolver uma prova de conceito a fim de verificar e validar o modelo implementado.
- *Construir a infraestrutura para projeto de domínio e subdomínios de ECOSs, o que requer:*
 1. identificar os elementos de ECOSs a partir do glossário de termos: atores, artefatos, papéis, responsabilidades e relacionamentos;
 2. estender modelos para sua representação: rede de produção de software (RPS) e rede sóciotécnica de ECOSs;
 3. estender a biblioteca Brechó para se transformar em um repositório de ativos de ECOSs, gerenciando esses elementos e adaptando ou implementando mecanismos para incorporar os elementos dos senso de negócio e de comunidade;
 4. desenvolver um sistema para automatizar a definição e modelagem de ECOSs, bem como um mecanismo de extração de subdomínios de ECOSs para análise;
 5. planejar e executar um estudo experimental do tipo experimento controlado com pesquisadores e profissionais de ES, visando avaliar o ambiente de DPE como um todo (análise e projeto de domínio de ECOSs);
 6. refinar o sistema a partir do pontos fracos e ameaças identificados pelo estudo experimental;
 7. integrar o sistema construído a ferramentas de análise de ECOSs, desenvolvidas por outros trabalhos do LENS/Reuse, e.g., um recomendador de tecnologias e de parceiros no ECOS (ALBERT, 2012) e um visualizador do ciclo social dos ECOSs (SANTOS *et al.*, 2013c);
 - *Redigir artigos para eventos e periódicos*, para divulgar os novos resultados para as comunidades relacionadas e angariar novas contribuições por parte dos pesquisadores e profissionais;
 - *Apoiar novos trabalhos do LENS/Reuse gerados a partir desta frente de pesquisa*, para participar do acompanhamento de alunos de iniciação

científica, de projeto final de graduação e de mestrado no desenvolvimento de partes da abordagem proposta, ou ainda de pesquisas que venham a se integrar com ela.

- *Redigir a tese de doutorado*, para concluir o trabalho de pesquisa e defendê-lo para uma banca composta por professores com experiência em ES, Reutilização de Software e ECOSs.

5.3 Resultados Preliminares

Alguns resultados iniciais foram obtidos a partir das atividades de pesquisa realizadas, resumidos a seguir:

- Revisão sobre conceitos de reutilização (Capítulo 2);
- Linha do tempo da Reutilização de Software (Seção 3.2);
- Revisão da literatura sobre ECOSs (Seções 3.3 e 3.4);
- *Roadmap* de pesquisa *ReuseECOS* (Capítulo 4);
- Identificação dos conceitos de ECOSs (Capítulos 3 e 4);
- Mecanismo de negociação e extensão do mecanismo de avaliação na biblioteca Brechó (elementos do senso de negócio);
- Definição dos mecanismos sociais, inicialmente orientados pela API do *Facebook Developers* (elementos do senso de comunidade);
- Proposta da abordagem e do ambiente *ReuseSEEM*
- Publicações: ALBERT *et al.* (2012), BARBOSA *et al.* (2013), BORGES *et al.*(2011); OLIVEIRA *et al.* (2011); SANTOS & WERNER (2010a; 2011a; 2011b; 2010c; 2011d; 2012a; 2012b), SANTOS *et al.* (2010b; 2010c; 2011; 2012a; 2012b; 2012c; 2012d; 2013a; 2013b; 2013c); WERNER & SANTOS (2010; 2012) e WERNER *et al.* (2010; 2012).
 - Outras:
 1. SANTOS, R.P., 2010, “Ecossistemas de Software: Tratando de Economia e Valor em ES”. In: *II Escola Regional de Informática da Regional Norte 1, Palestra*, Manaus, AM, Brasil, Março.
 2. SANTOS, R.P., 2010, “Gerência de Reutilização com a Ferramenta Brechó-VCM”. In: *II Escola Regional de Informática da Regional Norte 1, Minicurso*, Manaus, AM, Brasil, Março.

3. SANTOS, R.P., 2012, “Ecossistemas de Software”. *II Simpósio de Computação da UNIFEI, Palestra*, Itajubá, MG, Brasil, Abril.
4. SANTOS, R.P., 2012, “Ecossistemas de Software: Explorando Diversidade e Sustentabilidade na Indústria de Software”. In: *II Escola Regional de Informática Norte 3, Minicurso*, Porto Velho, RO, Brasil, Novembro.

5.4 Resultados Previstos

São esperados os seguintes resultados para a conclusão das atividades de pesquisa previstas após o exame de qualificação:

- Infraestrutura para Análise de Domínio de ECOSs:
 - Glossário de termos;
 - Modelo conceitual;
 - Subsistema para automatizar a fase de definição de ECOSs;
- Infraestrutura para Projeto de Domínio de ECOSs:
 - Conjunto de elementos e suas relações nos ECOSs, e respectivos modelos;
 - Subsistema para automatizar as fases de modelagem e de análise de ECOSs.
- Estudos experimentais
 - Pesquisa de opinião para avaliar os conceitos de ECOSs;
 - Experimento controlado para avaliar o ambiente de DPE;
- Submissões de artigos para eventos e periódicos (sugestões):
 - *Journal of Systems and Software*;
 - *Information and Software Technology*;
 - *International Workshop on Software Ecosystems*;
 - *International Conference on Software Business*;
 - *International Conference on Management of Emergent Digital EcoSystems*;
 - *International Conference on Digital Ecosystem Technologies*;
 - *Software Product Line Conference*;
 - *International Conference on Software Reuse*;
 - *International Conference on Software Engineering*;

- *International Conference on Global Software Engineering*;
- Workshop de Desenvolvimento Distribuído de Software;
- Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização;
- Simpósio Brasileiro de Engenharia de Software; etc.
- Participação em projetos e acompanhamento de alunos do LENS/Reuse;
- Tese de doutorado;

5.5 Cronograma Proposto

Para a obtenção dos resultados esperados neste trabalho de pesquisa, a Tabela 5.1 mostra a distribuição das seguintes atividades previstas nos meses restantes:

Tabela 5.1 – Cronograma de atividades para esta proposta de tese, onde T = trimestre

ANO	2013			2014				2015
Atividades	T2	T3	T4	T1	T2	T3	T4	T1
A. Refinar os conceitos de ECOSs	■							
B. Planejar e executar pesquisa de opinião para avaliar os conceitos de ECOSs		■						
C. Revisitar conceitos de ECOSs após o estudo		■						
D. Construir glossário de termos		■						
E. Construir modelo de domínio			■					
F. Identificar elementos de ECOSs			■					
G. Estender a representação de RPS e da rede sociotécnica de ECOSs			■					
H. Construir repositório de ativos e de casos de ECOSs como extensão da biblioteca Brechó				■	■			
I. Construir o sistema para o ambiente de DPE					■	■		
J. Planejar e executar experimento controlado para avaliar o sistema construído						■		
K. Revisitar o sistema construído							■	
L. Integrar ferramentas de análise de ECOSs ao sistema construído							■	
M. Redigir artigos	■		■		■		■	■
N. Escrever a tese de doutorado							■	■
O. Defender a tese de doutorado								■

Referências Bibliográficas

- ABECKER, A., BERNARDI, A., HINKELMANN, K., KUHN, O., SINTEK, M., 1998, “Toward a Technology for Organizational Memories”, **IEEE Intelligent Systems and their Applications**, v. 13, n. 3 (May/June), pp. 40-48.
- ALBERT, B.E., 2012, *Um Modelo de Governança de Componentes de Software sob a Perspectiva de Ecossistemas de Software*. Exame de Qualificação de Mestrado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- ALBERT, B.E., SANTOS, R.P., WERNER, C.M.L., 2012, “A Study on Software Components Governance Based on SOA Governance Elements”. In: *Anais do VI Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização de Software*, pp. 120-129, Natal, RN, Brasil, Setembro.
- ALMEIDA, E.S., ALVARO, A., GARCIA, V.C., MASCENA, J.C.C.P., BURÉGIO, V.A.A., NASCIMENTO, L., LUCRÉDIO, D., MEIRA, S.L., 2007, **C.R.U.I.S.E. – Component Reuse in Software Engineering**. Recife, PE, Brasil, RISE/CESAR.
- ALSPAUGH, T.A., ASUNCION, H.U., SCACCHI, W., 2009, “The Role of Software Licenses in Open Architecture Ecosystems”, In: *Proceedings of the First International Workshop on Software Ecosystems, 11th International Conference on Software Reuse*, pp. 4-18, Falls Church, VA, USA, September.
- ALVES, A.M., PÊSSOA, M., 2010, “Brazilian Public Software: Beyond Sharing”. In: *Proceedings of the 2nd ACM/IFIP International Conference on Management of Emergent Digital EcoSystems*, pp. 73-80, Bangkok, Thailand, October.
- ALVES, A.M., PÊSSOA, M., SALVIANO, C.F., 2011, “Towards a Systemic Maturity Model for Public Software Ecosystems”. In: O’CONNOR, R.V., ROUT, T., MCCAFFERY, F. & DORLING, A. (eds.), *Software Process Improvement and Capability Determination, Communications in Computer and Information Science*, Volume 155, pp. 145-156. Springer Berlin Heidelberg.
- ALVES, A.M., STEFANUTO, G., SALVIANO, C., DRUMMOND, P., MEFFE, C., 2009, “Learning Path to an Emergent Ecosystem: The Brazilian Public Software Experience”. In: *Proceedings of the 1st ACM/IFIP International Conference on Management of Emergent Digital EcoSystems*, pp. 454-455, Lyon, France, October.

- ANDRADE, R.V.C.L., 2012, *Understanding Strategies During the Software Ecosystems Lifecycle*. Dissertação de M.Sc., CIn/UFPE, Recife, PE, Brasil.
- ANGEREN, J., KABBEDJIK, J., JANSEN, S., POPP, K.M., 2011, "A Survey of Associate Models used within Large Software Ecosystems", In: *Proceedings of the 3rd International Workshop on Software Ecosystems, 2nd International Conference on Software Business*, pp. 27-39, Brussels, Belgium, June.
- ANVAARI, M., JANSEN, S., 2010, "Evaluating Architectural Openness in Mobile Software Platforms", In: *Proceedings of the 4th European Conference on Software Architecture, 2nd International Workshop on Software Ecosystems*, pp. 85-92, Copenhagen, Denmark, August.
- ARANGO, G.F., 1988, *Domain Engineering for Software Reuse*. Doctoral Dissertation, University of California, Irvine, USA.
- ARANGO, G., 1994, "Domain Analysis Methods". In: SCHAFER, W., PRIETO-DÍAZ, R. & MATSUMOTO, M. (eds.), *Software Reusability*, pp. 17-49. 1.ed., Prentice Hall PTR.
- ARNDT, J.M., DIBBERN, J., 2006, "Co-Innovation in a Service Oriented Strategic Network", In: *Proceedings of the IEEE International Conference on Services Computing*, pp. 285-288, Chicago, IL, USA, September.
- BANKER, R.D., KAUFFMAN, R.J., ZWEIG, D., 1993, "Repository evaluation of Software Reuse", ***IEEE Transactions on Software Engineering***, v. 19, n. 4 (April), pp. 379-389.
- BARBOSA, O.A.L.P., 2012, *Understanding Software Ecosystems: Implications for Practive and Theory*. Dissertação de M.Sc., CIn/UFPE, Recife, PE, Brasil.
- BARBOSA, O., ALVES, C., 2011, "A Systematic Mapping Study on Software Ecosystems", In: *Proceedings of the 3rd International Workshop on Software Ecosystems, 2nd International Conference on Software Business*, pp. 15-26, Brussels, Belgium, June.
- BARBOSA, O., SANTOS, R.P., ALVES, C., WERNER, C., JANSEN, S., 2013, "A Systematic Mapping Study on Software Ecosystems through a Three-dimensional Perspective". In: JANSEN, S., CUSUMANO, M. & BRINKKEMPER, S. (eds.), *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry*, pp. 59-81. Cheltenham, UK, and Northampton, MA, USA, Edward Elgar Publishing.

- BARROCA, L., GIMENES, I.M.S., HUZITA, E.H.M., 2005, “Conceitos Básicos”. In: Gimenes, I.M.S, Huzita, E.H.M. (org.), *Desenvolvimento Baseado em Componentes: Conceitos e Técnicas*, pp. 1-26, Rio de Janeiro, RJ, Brasil, Editora Ciência Moderna.
- BARROS, M., 1995, *Recuperação de Componentes em Bibliotecas de Software*. Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- BASILI, V.R., SHULL, F., LANUBILE, F., 1999, “Building Knowledge through Families of Experiments”, *IEEE Transactions on Software Engineering*, v. 25, n. 4 (July/August), pp. 456-473.
- BASS, L., BUHMAN, C., COMELLA-DORDA, S., LONG, F., ROBERT, J., SEACORD, R., WALLNAU, K., 2000, *Market Assessment of Component-Based Software Engineering*, Technical Report CMU/SEI-2001-TN-007, Software Engineering Institute.
- BECK, K., BEEDLE, M., BENNEKUM, A., COCKBURN, A., CUNNINGHAM, W., FOWLER, M., GRENNING, J., HIGHSMITH, J., HUNT, A., JEFFRIES, R., KERN, J., MARICK, B., MARTIN, R.C., MELLOR, S., SCHWABER, K., SUTHERLAND, J., THOMAS, D., 2001, “Manifesto for Agile Software Development”. Disponível em <<http://www.agilemanifesto.org>>. Acesso em: 20 jan. 2013, 17:00:00.
- BERK, I., JANSEN, S., LUINENBURG, L., S., 2010, “Software Ecosystems: A Software Ecosystem Strategy Assessment Model”, In: *Proceedings of the 4th European Conference on Software Architecture, 2nd International Workshop on Software Ecosystems*, pp. 135-142, Copenhagen, Denmark, August.
- BIFFL, S., AURUM, A., BOEHM, B., ERDOGMUS, H., GRÜNBACHER, P. (eds.), 2006, *Value-Based Software Engineering*. Berlin, Germany, Springer-Verlag.
- BLOIS, A.P.T.B., 2006, *Uma Abordagem de Projeto Arquitetural baseado em Componentes no contexto de Engenharia de Domínio*. Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- BOEHM, B.W., 1981, *Software Engineering Economics*. 1. ed., Upper Saddle River, NJ, USA, Prentice Hall.
- BOEHM, B., 2006, “A View of 20th and 21st Century Software Engineering”, In: *Proceedings of the 28th International Conference on Software Engineering*, pp. 12-29, Shanghai, China, May.

- BOEHM, B., HUANG, L.G., 2003, “Value-Based Software Engineering: A Case Study”, *IEEE Computer*, v. 36, n. 3 (March), pp. 33-41.
- BOEHM, B., JAIN, A., 2007, “Developing a Process Framework using Principles of Value-Based Software Engineering: Research Sections”, *Software Process: Improvement and Practice*, v. 12, n. 5 (September), pp. 377-385.
- BOEHM, B., SULLIVAN, K.J., 2000, “Software Economics: A Roadmap”, In: *Proceedings of the 22nd International Conference on Software Engineering, The Future of Software Engineering*, pp. 319-344, Limerick, Ireland, June.
- BORGES, H.S., SANTO, R.E., SANTOS, R.P., COSTA, H.A.X., WERNER, C.M.L., 2011, “Portal EduES 2.0: Uma Ferramenta para Apoiar a Gerência de Reutilização no Domínio de Educação em ES”, In: *Anais do VII Workshop Anual do MPS, Sessão de Ferramentas*, pp. 156-164, Campinas, SP, Brasil.
- BOSCH, J., 2000, *Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach*. Addison-Wesley Professional.
- BOSCH, J., 2009, “From Software Product Lines to Software Ecosystem”, In: *Proceedings of 13th International Software Product Line Conference*, pp. 1-10, San Francisco, CA, USA, August.
- BOSCH, J., 2010, “Architecture Challenges for Software Ecosystems”, In: *Proceedings of the 4th European Conference on Software Architecture, 2nd International Workshop on Software Ecosystems*, pp. 93-95, Copenhagen, Denmark, August.
- BOSCH, J., 2012, “Software Ecosystems Taking Software Development Beyond the Boundaries of the Organization”, *Journal on Systems and Software*, v. 85, n. 7 (July), pp.1453-1454.
- BOSCH, J., BOSCH-SIJTSEMA, P., 2010, “From Integration to Composition: On the Impact of Software Product Lines, Global Development and Ecosystems”, *Journal of Systems and Software*, v. 83, n. 1 (January), pp. 67-76.
- BOUCHARAS, V., JANSEN, S., BRINKKEMPER, S., 2009, “Formalizing Software Ecosystem Modeling”, In: *Proceedings of the 1st International Workshop on Open Component Ecosystems, ACM SIGSOFT Symposium on the Foundations of Software Engineering*, pp. 41-50, Amsterdam, The Netherlands, August.
- BRAGA, R., 2000, *Busca e Recuperação de Componentes em Ambientes de Reutilização de Software*. Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

- BRECHÓ, 2013, “Projeto Brechó: Sistema de Informação para WEB com uma Base de Dados de Componentes, Fornecedores e Consumidores”. Disponível em <<http://reuse.cos.ufrj.br>>. Acesso em: 20 jan. 2013, 17:00:00.
- BREITMAN, K., 2005, **Web Semântica – A Internet do Futuro**. 1.ed., LTC.
- BRERETON, P., BUDGEN, D., 2000, “Component-based Systems: A Classification of Issues”, **IEEE Computer**, v. 33, n. 11 (November), pp. 54-62.
- BROOKS, F.P., 1995, **The Mythical Man-Month: Essays on Software Engineering**. 2. ed., Addison-Wesley Professional.
- BROWN, A.W., 2000, “Large Scale Component Based Development”. 1.ed., Prentice Hall PTR.
- BURMANN, F.N., MAURO, P.A., 2011, *O Papel dos Technology Brokers para a Facilitação da Inovação Aberta: Uma Análise de Tipos, Características e Funcionalidades Modelo*. Projeto Final de B.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- BUSCHMANN, F., MEUNIER, R., ROHNERT, H., SOMMERLAD, P., STAL, M., 1996, **Pattern-Oriented Software Architecture Volume 1: A System of Patterns**. Wiley.
- CAI, X., LYU, M.R., WONG, K., KO, R., 2000, “Component-Based Software Engineering: Technologies, Development Frameworks, and Quality Assurance Schemes”, In: *Proceedings of the 7th Asia-Pacific Software Engineering Conference*, p. 372, Singapore, Singapore, December.
- CAMPBELL, P.R.J., AHMED, F., 2010, “A Three-dimensional View of Software Ecosystems”, In: *Proceedings of the 4th European Conference on Software Architecture, 2nd International Workshop on Software Ecosystems*, pp. 81-84, Copenhagen, Denmark, August.
- CAMPBELL, P.R.J., AHMED, F., BOSCH, J. & JANSEN, S., 2010, “2nd International Workshop on Software Ecosystems (EcoSys)”, In: *Proceedings of the Fourth European Conference on Software Architecture, 2nd International Workshop on Software Ecosystems*, pp. 63-64, Copenhagen, Denmark, August.
- CAPURUÇO, R.A.C., CAPRETZ, L.F., 2010, “Integrating Recommender Information in Social Ecosystems Decisions”, In: *Proceedings of the 4th European Conference on Software Architecture, 2nd International Workshop on Software Ecosystems*, pp. 143-150, Copenhagen, Denmark, August.

- CATALDO, M., HERBSLEB, J.D., 2010, “Architecting in Software Ecosystems: Interface Translucence as an Enabler for Scalable Collaboration”, In: *Proceedings of the 4th European Conference on Software Architecture, 2nd International Workshop on Software Ecosystems*, pp. 65-72, Copenhagen, Denmark, August.
- CHESBROUGH, H.W., 2003, ***Open Innovation: The New Imperative for Creating and Profiting from Technology***. Harvard Business School Publishing Corporation.
- CLEMENTS, P., NORTHROP, L., 2001, ***Software Product Lines: Practices and Patterns***. 3. ed., Addison-Wesley Professional.
- CORCHO, O., FERNÁNDEZ-LÓPEZ, M., GÓMEZ-PÉREZ, A., 2003, “Methodologies, Tools and Languages for Building Ontologies: Where is their Meeting Point?”, ***Data & Knowledge Engineering***, v. 46, n. 1 (July), pp. 41-64.
- COSTA, G.C.B.C., 2013, *Uma Abordagem para Linha de Produtos de Software Científico Baseada em Ontologia e Workflow*. Dissertação de M.Sc., UFJF, Juiz de Fora, MG, Brasil.
- CUKIERMAN, H.L., TEIXEIRA, C., PRIKLADNICKI, R., 2007, “Um Olhar Sociotécnico sobre a ES”, ***Revista de Informática Teórica e Aplicada***, v. 14, n. 2, pp. 207-227.
- CZARNECKI, K., EISENECKER, U., 2000, ***Generative Programming: Methods, Tools, and Applications***. 1.ed., Addison-Wesley Professional.
- DEST, 2007, “Inaugural IEEE International Digital Ecosystems and Technologies Conference”. Disponível em <<http://www.ieee-dest.curtin.edu.au/2007/index.php>>. Acesso em: 20 jan. 2013, 17:00:00.
- DHUNGANA, D., GROHER, I., SCHLUDERMANN, E., BIFFL, S. (2010) “Software Ecosystems vs. Natural Ecosystems: Learning from the Ingenious Mind of Nature”, In: *Proceedings of the 4th European Conference on Software Architecture, 2nd International Workshop on Software Ecosystems*, pp. 96-102, Copenhagen, Denmark, August.
- ENSMENGER, N., ASPRAY, W., 2002, “Software as Labor Process”. In: HASHAGEN, U., KEIL-SLAWIK, R., NORBERG, A. (eds.), *History of Computing: Software Issues*, pp. 139-165, Berlin, Germany, Springer-Verlag.
- FALBO, R.A., GUIZZARDI, G., DUARTE, K.C., 2002, “An Ontological Approach to Domain Engineering”, In: *Proceedings of the 14th International Conference on*

- Software Engineering and Knowledge Engineering*, pp. 351-358, Ischia, Italy, July.
- FAVARO, J., PFLEEGER, S.L., 2011, "Software as a Business", **IEEE Software**, v. 28, n.4 (July/August), pp. 22-25.
- FERNANDES, P.C.C., 2009, *UbiFEX: Uma Abordagem para Modelagem de Características de Linha de Produtos de Software Sensíveis ao Contexto*. Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- FERREIRA FILHO, J.B., BARAIS, O., BAUDRY, B., VIANA, W., ANDRADE, R. M.C., 2012, "An Approach for Semantic Enrichment of Software Product Lines". In: *Proceedings of the 16th International Software Product Line Conference*, v. 2, pp. 188-195, Salvador, BA, Brazil, Setembro.
- FLECK, L., 1986, ***La Génesis y el Desarrollo de un Hecho Científico: Introducción a la Teoría del Estilo de Pensamiento y del Colectivo de Pensamiento***. Madrid, Spain, Alianza.
- FORRESTER, 2013, "Forrester Research, Inc.". Disponível em <<http://www.forrester.com/home>>. Acesso em: 20 jan. 2013, 17:00:00.
- FRAKES, W.B., KANG, K.C., 2005, "Software Reuse Research: Status and Future", **IEEE Transactions on Software Engineering**, v. 31, n. 7 (July), pp. 529-536.
- FRANÇA, A.C.C., SILVA, F.Q.B., 2009, "Developing Motivational Programs for Software Engineers through an Experimental Method", In: *Proceedings of the XXIII Brazilian Symposium on Software Engineering*, pp. 145-154, Fortaleza, CE, Brazil, October.
- FRANZOSI, E. M., GARCIA, A., RODRIGUES, S. A., BLASCHEK, J. R., SOUZA, J. M., 2009, "Uma Proposta de Arquitetura Referencial SOA para Desenvolvimento de Sistemas para o Governo". In: *Anais do I Workshop de Computação Aplicada em Governo Eletrônico, XXIX Congresso da Sociedade Brasileira de Computação*, pp. 1417-1430, Bento Gonçalves, RS, Brasil, Julho.
- FREEMAN, P. *Reusable Software Engineering Concepts and Research Directions*, Software Reusability (Tutorial). Washington, D.C., IEEE Computer Society Press, pp. 10-23.
- FRICKER, S., 2009, "Specifications and Analysis of Requirements Negotiation Strategy in Software Ecosystems", In: *Proceedings of the First International Workshop on Software Ecosystems, 11th International Conference on Software Reuse*, pp. 19-33, Falls Church, VA, USA, September.

- GARTNER, 2013, “*Technology Research Gartner, Inc.*”. Disponível em <<http://www.gartner.com/technology/home.jsp>>. Acesso em: 20 jan. 2013, 17:00:00.
- GIMENES, I.M., TRAVASSOS, G.H., 2002, “O Enfoque de Linha de Produto para Desenvolvimento de Software”. In: *Anais do XXII Congresso da Sociedade Brasileira de Computação, XXI Jornada de Atualização em Informática*, Florianópolis, SC, Brasil, Julho.
- GOLDMAN, R., GABRIEL, R.P., 2005, ***Innovation Happens Elsewhere***. Morgan Kaufmann.
- GRUBER, T.R., 1993, A Translation Approach to Portable Ontology Specifications, ***Knowledge Acquisition***, v. 5, n. 2 (June), pp. 199-220.
- GUERRA, A.C., COLOMBO, R.M.T., 2009, ***Tecnologia da Informação: Qualidade de Produto de Software***. Brasília, DF, Brasil, PBQP Software.
- HANSEN, G.K., 2012, “A Longitudinal Case Study of an Emerging Software Ecosystem: Implications for Practice and Theory”, ***Journal of Systems and Software***, v. 85, n. 7 (July), p. 1455-1466.
- HANSEN, G.K., DYBA, T., 2012, “Theoretical Foundations of Software Ecosystems”, In: *Proceedings of the 4th International Workshop on Software Ecosystems, 3rd International Conference on Software Business*, pp. 6-17, Boston, MA, USA, June.
- HARTIGH, E., TOL, M., VISSCHER, W., 2006, “The Health Measurement of a Business Ecosystem”. In: *Proceedings of the Sixth Annual Meeting of the European Chaos and Complexity in Organisations Network*, pp. 1-39, Bergen aan Zee, The Netherlands, October.
- HILLEGERSBERG, J., TRAAS, V., DRAGT, R., 2001, “A Longitudinal Study of the e-Market for Software Components”, In: *Proceedings of the 14th Bled Electronic Commerce Conference*, pp. 481-494, Bled, Slovenia, June.
- HUNINK, I., VAN ERK, R., JANSEN, S., BRINKKEMPER, S., 2010, “Industry Taxonomy Engineering: The Case of the European Software Ecosystem”, In: *Proceedings of the 4th European Conference on Software Architecture, 2nd International Workshop on Software Ecosystems*, pp. 111-118, Copenhagen, Denmark, August.
- HYRYNSALMI, S., MÄKILÄ, T., JÄRVI, A., SUOMINEN, A., SEPPÄNEN, M., KNUUTILA, T., 2012, “App Store, Marketplace, Play! An Analysis of Multi-

- Homing in Mobile Software Ecosystems”, In: *Proceedings of the 4th International Workshop on Software Ecosystems, 3rd International Conference on Software Business*, pp. 59-72, Boston, MA, USA, June.
- IANSITI, M., LEVIEN, R., 2004a, “Strategy as Ecology”, *Harvard Business Review*, v. 82, n. 3 (March), pp. 68-78.
- IANSITI, M., LEVIEN, R., 2004b, *The Keystone Advantage: What the New Dynamics of Business Ecosystems Mean for Strategy, Innovation, and Sustainability*. Harvard Business Review Press.
- IEEE, 2004, *Std 1517 - IEEE Standard for Information Technology - Software Life Cycle Processes - Reuse Processes*, Institute of Electrical and Electronics Engineers.
- ISO/IEC, 1995, *ISO/IEC 12207: Information Technology – Software Life Cycle Processes*. Geneve.
- ITABORAHY, A.L.C., 2007, *Gerenciamento de Projetos de Software Baseado no Valor para o Negócio*. Dissertação de M.Sc., Universidade Católica de Brasília, Brasília, DF, Brasil.
- IYER, B, 2012, “EcoSysNetworks: A Method for Visualizing Software Ecosystems”, In: *Proceedings of the 4th International Workshop on Software Ecosystems, 3rd International Conference on Software Business*, pp. 1-5, Boston, MA, USA, June.
- JANSEN, S.R.L., 2007, *Customer Configuration Updating in a Software Supply Network*. PhD Thesis, Utrecht University, Utrecht, The Netherlands.
- JANSEN, S., BRINKKEMPER, S., FINKELSTEIN, A., 2009a, “Business Network Management as a Survival Strategy: A Tale of Two Software Ecosystems”, In: *Proceedings of the First International Workshop on Software Ecosystems, 11th International Conference on Software Reuse*, pp. 34-48, Falls Church, VA, USA, September.
- JANSEN, S., CUSUMANO, M., 2012, “Defining Software Ecosystems: A Survey of Software Ecosystems and Ecosystem”, In: *Proceedings of the 4th International Workshop on Software Ecosystems, 3rd International Conference on Software Business*, pp. 41-58, Boston, MA, USA, June.
- JANSEN, S., CUSUMANO, M., BRINKKEMPER, S. (eds.), 2013, *Software Ecosystems: Analyzing and Managing Business Networks in the Software*

- Industry.** Cheltenham, UK, and Northampton, MA, USA, Edward Elgar Publishing.
- JANSEN, S., FINKELSTEIN, A., BRINKKEMPER, S., 2009b, “A Sense of Community: A Research Agenda for Software Ecosystems”, In: *Proceedings of the 31st International Conference on Software Engineering, New and Emerging Research Track*, pp. 187-190, Vancouver, BC, Canada, May.
- JASPER, R., USCHOLD, M., 1999, “A Framework for Understanding and Classifying Ontology Applications”. In: *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods*, pp. 11.1-11.12, Stockholm, Sweden, August.
- KANG, K.C., COHEN, S.G., HESS, J.A., NOVAK, W.E., PETERSON, A.S., 1990, *Feature-Oriented Domain Analysis (FODA) – Feasibility Study*, Technical Report CMU/SEI-90-TR-21, Software Engineering Institute.
- KITTLAUS, H.B., CLOUGH, P.N., 2009, ***Software Product Management and Pricing: Key Success Factors for Software Organizations***. Springer Publishing Company.
- KREBS, C.J., 2008, ***Ecology: The Experimental Analysis of Distribution and Abundance***. 6. ed., Benjamin Cummings.
- KRUEGER, C.W., 1992, “Software Reuse”, *ACM Computing Surveys*, v. 24, n. 2 (June), pp. 131-183.
- LATOUR, B., 2000, ***Ciência em Ação: Como Seguir Cientistas e Engenheiros Sociedade Afora***. São Paulo, SP, Brasil, Editora UNESP.
- LEE, J., MUTHIG, D., 2006, “Feature-oriented Variability Management in Product Line Engineering”, *Communications of the ACM*, v. 49, n. 12 (December), pp. 55-59.
- LIMA, D., SILVA, R., GARCIA, A., RODRIGUES, S., CHAVES, M., SANTOS, R., SOUZA, J., “Sistema para Transição de Governos no Brasil”, In: *IX Simpósio Brasileiro de Sistemas de Informação*, João Pessoa, PB, Brasil, Maio. (submetido)
- LOPES, V.P., 2010, *Repositório de Conhecimento de um Ambiente de Apoio a Experimentação em ES*. Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

- LOPES, L.G., MURTA, L., WERNER, C., 2006, “Odyssey-CCS: A Change Control System Tailored to Software Reuse”, In: *Proceedings of the 9th International Conference on Software Reuse*, pp. 170-183, Torino, Italy, June.
- LUCRÉDIO, D., PRADO, A.F., ALMEIDA, E.S., 2004, “A Survey on Software Components Search and Retrieval”. In: *Proceedings of the 30th EUROMICRO Conference*, pp. 152-159, Rennes, France, August-September.
- LUNGU, M., LANZA, M., GIRBA, T., ROBBES, R., 2010, “The Small Project Observatory: Visualizing Software Ecosystems”, ***Science of Computer Programming***, v. 75, n. 4 (April), pp. 264-275.
- LUQI, GUO, J., 1999, “Toward Automated Retrieval for a Software Component Repository”, In: *Proceedings of the Conference and Workshop on Engineering of Computer-Based Systems*, pp. 99-105, Nashville, TN, USA, March.
- MAGDALENO, A.M., 2010, *Apoio à Decisão para o Balanceamento de Colaboração e Disciplina nos Processos de Desenvolvimento de Software*. Exame de Qualificação de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- MAFRA, S.N., TRAVASSOS, G.H., 2006, “Estudos Primários e Secundários Apoiando a Busca por Evidência em ES”, Relatório Técnico ES687/06, PESC/COPPE/UFRJ, Março de 2006.
- MANIKAS, K., HANSEN, K.M., 2013, “Software Ecosystems – A Systematic Literature Review”, ***Journal of Systems and Software***, v. 86, n. 5 (May), pp. 1294-1306.
- MARINHO, A., MURTA, L., WERNER, C., 2009a, “Extending a Software Component Repository to Provide Services”, In: *Proceedings of the 11th International Conference on Software Reuse*, Falls Church, VA, USA, September. *Lecture Notes in Computer Science*, v. 5791, Springer, pp. 258-268.
- MARINHO, A.S., MURTA, L.G.P., WERNER, C.M.L., 2009b, “Infraestrutura de Serviços na Biblioteca de Componentes Brechó”, In: *Anais da Sessão de Ferramentas do III Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização de Software*, pp. 18-25, Natal, RN, Brasil, Setembro.
- MARINHO, A.S., SANTOS, R.P., MURTA, L.G.P., WERNER, C.M.L., 2009c, “Infraestrutura de Tarificação de Componentes e Serviços na Biblioteca Brechó”. In: *Anais da Sessão de Ferramentas do III Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização de Software*, pp. 2-9, Natal, RN, Brasil, Setembro.

- MCGREGOR, J. D., 2010, "A Method for Analyzing Software Product Line Ecosystems", In: *Proceedings of the 4th European Conference on Software Architecture, 2nd International Workshop on Software Ecosystems*, pp. 73-80, Copenhagen, Denmark, August.
- MCGREGOR, J. D., 2012, "Ecosystem Modeling and Analysis", In: *Proceedings of 16th International Software Product Line Conference*, p. 268, Salvador, BA, Brazil, August.
- MEIRA, A.F., SILVA, F.Q.B., 2009, "Habillities and Behavioural Profiles of SQA Professionals Related to Process Maturity Levels", In: *Proceedings of the XXIII Brazilian Symposium on Software Engineering*, pp. 136-144, Fortaleza, CE, Brazil, October.
- MENDES, A., 2002, ***Arquitetura de Software: Desenvolvimento Orientado para Arquitetura***. Campus.
- MENS, T., GOEMINNE, M., 2011, "Analysing the Evolution of Social Aspects of Open Source Software Ecosystems", In: *Proceedings of the 3rd International Workshop on Software Ecosystems, 2nd International Conference on Software Business*, pp. 77-88, Brussels, Belgium, June.
- MESSERSCHMITT, D.G., SZYPERSKI, C., 2003, ***Software Ecosystem: Understanding an Indispensable Technology and Industry***. 1. ed., The MIT Press.
- MILER JÚNIOR, N., 2000, *A Engenharia de Aplicações no Contexto da Reutilização baseada em Modelos de Domínio*. Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- MOLDER, J., LIER, B., JANSEN, S., 2011, "Clopenness of Systems: The Interwoven Nature of Ecosystems", In: *Proceedings of the 3rd International Workshop on Software Ecosystems, 2nd International Conference on Software Business*, pp. 52-64, Brussels, Belgium, June.
- MONTONI, M.A., 2003, *Aquisição de Conhecimento: Uma aplicação no Processo de Desenvolvimento de Software*. Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- MOORE, J.F., 1993, "Predators and Prey: A New Ecology of Competition", ***Harvard Business Review***, v. 71, n. 3 (May/June), pp. 75-86.
- MOORE, J.F., 1996, ***The Death of Competition: Leadership & Strategy in the Age of Business Ecosystems***. Harper Business.

- MOORE, J., BAILIN, S., 1991, "Domain Analysis: Framework for Reuse". In: PRIETO-DÍAZ, R., ARANGO, G. (eds), *Domain Analysis and Software System Modeling*, pp. 179-203, Los Alamitos, CA, USA, IEEE Computer Society Press.
- NAUR, P., RANDELL, B. (eds.), 1969, *Software Engineering: Report on a Conference Sponsored by the NATO Science Committee*, Garmish, Germany, 7th to 11th October 1968. Brussels, Scientific Affairs Division.
- NEIGHBORS, J.M., 1980, *Software Construction Using Components*. Doctoral Dissertation, University of California, Irvine, USA.
- NORTHROP, L., 2004, *Software Product Line Adoption Roadmap*, Technical Report CMU/SEI-2004-TR-022, Software Engineering Institute.
- ODYSSEY, 2013, "Projeto Odyssey: Infraestrutura de Suporte à Reutilização Baseada em Modelos de Domínio". Disponível em <<http://reuse.cos.ufrj.br>>. Acesso em: 20 jan. 2013, 17:00:00.
- O'LEARY, D.E., 1998, "Using AI in Knowledge Management: Knowledge Bases and Ontologies", In: **IEEE Intelligent Systems**, v. 13, n. 3 (May), pp. 34-39.
- OLIVEIRA, R.F., 2006, *Formalização e Verificação de Consistência na Representação de Variabilidades*. Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- OLIVEIRA, F.M.A., SANTOS, R.P., COSTA, A.V., 2011, "Plataforma para a Rede de Políticas Públicas". In: *Anais do XIV Encontro Nacional da Associação Nacional de Pós-Graduação e Pesquisa em Planejamento Urbano e Regional, Sessão Livre 11 - Política Pública, Rede e Mutação do Território*, pp. 262-263, Rio de Janeiro, RJ, Brasil.
- OMG, 2008, "Software Process Engineering Meta-Model – SPEM 2.0". Disponível em <<http://www.omg.org/spec/SPEM/2.0/>>. Acesso em: 20 jan. 2013, 17:00:00.
- OVERHAGE, S., THOMAS, P., 2004, "A Business Perspective on Component Trading: Criteria, Immaturities, and Critical Success Factors", In: *Proceedings of the 30th Euromicro Conference*, pp. 108-117, Rennes, France, September.
- PETTERSSON, O., SVENSSON, M., GIL, D., ANDERSSON, J., MILRAD, M., 2010, "On the Role of Software Process Modeling in Software Ecosystem Design", In: *Proceedings of the 4th European Conference on Software Architecture, 2nd International Workshop on Software Ecosystems*, pp. 103-110, Copenhagen, Denmark, August.

- PINTO, A., ALMEIDA, A.C.M., MORAIS, E., 2009, “Collaborative Software Development Process for Geographically Distributed Teams”, In: *Anais do III Workshop de Desenvolvimento Distribuído de Software, XXIII Simpósio Brasileiro de ES*, pp. 89-97, Fortaleza, CE, Brasil, Outubro.
- POPP, K.M., 2011, “Hybrid Revenue of Software Companies and their Relationship to Hybrid Business Models”, In: *Proceedings of the 3rd International Workshop on Software Ecosystems, 2nd International Conference on Software Business*, pp. 77-88, Brussels, Belgium, June.
- POPP, K.M., 2012, “Leveraging Open Source Licenses and Open Source Communities in Hybrid Commercial Open Source Business Models”, In: *Proceedings of the 4th International Workshop on Software Ecosystems, 3rd International Conference on Software Business*, pp. 33-40, Boston, MA, USA, June.
- POULIN, J.S., 1996, ***Measuring Software Reuse: Principles, Practices, and Economic Models***. Addison Wesley Professional.
- PRADO, L.J., 2002, ***Guia Balanced Scorecard***. 1.ed., e-book, LJP E-ZINE – A Revista Eletrônica de Gestão.
- PRESSMAN, R.S., 2010, ***Software Engineering – A Practitioner’s Approach***. 7. ed. New York, NY, USA, McFraw-Hill.
- PRIETO-DÍAZ, R., ARANGO, G., 1991, ***Domain Analysis and Software Systems Modeling***. Los Alamitos, CA, USA, IEEE Computer Society Press.
- RUSS, C., 2007, “Online Crowds – Extraordinary Mass Behavior on the Internet”. In: *Proceedings of the International Conference on New Media Technology*, pp. 65-76, Graz, Austria, September.
- SALMINEN & MIKKONEN, 2012, “Mashups – Software Ecosystems for the Web Era”, In: *Proceedings of the 4th International Workshop on Software Ecosystems, 3rd International Conference on Software Business*, pp. 18-32, Boston, MA, USA, June.
- SAMETINGER, J., 1997, ***Software Engineering with Reusable Components***. 1. ed. Berlin, Germany, Springer-Verlag.
- SANTANA, F.W., WERNER, C.M.L., 2013, “Towards the Analysis of Software Projects Dependencies: An Exploratory Visual Study of Software Ecosystems”. In: *Proceedings of the 5th International Workshop on Software Ecosystems, 4th International Conference on Software Business*, pp. 7-18, Potsdam, Germany, June.

- SANTO, R.E., 2012, “Serviços de Apoio ao Planejamento de Revisões Sistemáticas da Literatura”. Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- SANTO, R.E., SANTOS, R.P., WERNER, C.M.L., TRAVASSOS, G.H., 2009, “Portal EduES Brasil: Um Ambiente para Apoiar a Pesquisa em Educação em ES no Brasil”. In: *Anais do II Fórum de Educação em ES, XXIII Simpósio Brasileiro de ES*, pp. 33-40, Fortaleza, CE, Brasil.
- SANTOS, R.P., 2008, “Elementos para uma Abordagem Sociotécnica do Desenvolvimento de Software com *Extreme Programming*”, **Scientia**, v. 19, n. 2 (Julho-Dezembro), pp. 102-116.
- SANTOS, R.P., 2010, *Brechó-VCM: Uma Abordagem Baseada em Valor para Mercados de Componentes*. Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- SANTOS, R.P., WERNER, C.M.L., 2009, “Uma Abordagem Baseada em Valor para um Mercado de Componentes Apoiado pela Evolução de Repositórios”. In: *Anais do III Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização de Software*, pp. 152-165, Natal, RN, Brasil.
- SANTOS, R.P., WERNER, C.M.L., 2010a, “Revisiting the Concept of Components in Software Engineering from a Software Ecosystem Perspective”, In: *Proceedings of the 4th European Conference on Software Architecture, 2nd International Workshop on Software Ecosystems*, pp. 135-142, Copenhagen, Denmark, August.
- SANTOS, R.P., WERNER, C.M.L., 2010b, “Analyzing the Concept of Components in the Brechó-VCM Approach through a Sociotechnical and Software Reuse Management Perspective”. In: *Proceedings of the Fourth Brazilian Symposium on Software Components, Architectures and Reuse*, pp. 21-30, Salvador, BA, Brazil.
- SANTOS, R.P., WERNER, C.M.L., 2011a, “Gerência e Desenvolvimento de Sistemas de Informação na Realidade dos Ecossistemas de Software”, In: *Anais do VII Simpósio Brasileiro de Sistemas de Informação, Minicursos*, pp. 20-25, Salvador, BA, Brasil, Maio.
- SANTOS, R.P., WERNER, C.M.L., 2011b, “A Proposal for Software Ecosystems Engineering”, In: *Proceedings of the 3rd International Workshop on Software Ecosystems, 2nd International Conference on Software Business*, pp. 40-51, Brussels, Belgium, June.

- SANTOS, R.P., WERNER, C.M.L., 2011c, “Brechó-EcoSys: From a Component Library to a Software Ecosystems Platform”. In: *12th International Conference on Software Reuse, Tool Demonstrations*, pp. 1-4, Pohang, South Korea, June.
- SANTOS, R.P., WERNER, C.M.L., 2011d, “Treating Business Dimension in Software Ecosystems”, In: *Proceedings of the 3rd ACM/IFIP International Conference on Management of Emergent Digital EcoSystems*, pp. 197-201, San Francisco, CA, USA, November.
- SANTOS, R.P., WERNER, C.M.L., 2012a, “Treating Social Dimension in Software Ecosystems through ReuseECOS Approach”, In: *Proceedings of the IEEE DEST-CEE 2012 – 6th International Conference on Digital Ecosystem Technologies – Complex Environment Engineering*, pp. 1-6, Campione d’Italia, Italy, June.
- SANTOS, R.P., WERNER, C.M.L., 2012b, “ReuseECOS: An Approach to Support Global Software Development through Software Ecosystems”, In: *Proceedings of the 7th IEEE International Conference on Global Software Engineering Workshops, VI Workshop on Distributed Software Development*, pp. 60-65, Porto Alegre, Brazil, August.
- SANTOS, R.P., COSTA, H.A.X., WERNER, C.M.L., AMÂNCIO, A.F., PARREIRA JÚNIOR, P.A., RESENDE, A.M.P., SILVEIRA, F.F., 2009a, “An Approach Based on Maintainability Criteria for Building Aspect-Oriented Software Design Model”, *Revista de Informática Teórica e Aplicada*, v. XVI, n. 2, pp. 157-171.
- SANTOS, R.P., SILVA, M.A., WERNER, C.M.L., 2009b, “Mecanismos de Classificação e de Precificação para Explorar Informações sobre o Valor de Componentes na Biblioteca Brechó”. In: *Anais da Sessão de Ferramentas do III Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização de Software*, pp. 10-17, Natal, RN, Brasil, Setembro.
- SANTOS, R.P., WERNER, C.M.L., SILVA, M.A., 2010a, “Brechó-VCM: A Value-Based Approach for Component Markets”, *International Transactions on Systems Science and Applications*, v. 6, n. 2/3 (August), pp. 179-199.
- SANTOS, R.P., MARINHO, A.S., SILVA, M.A., WERNER, C.M.L., MURTA, L.G.P., 2010b, “Brechó 2.0: Uma Ferramenta para Apoiar a Gerência de Reutilização” In: *Anais do VI Workshop Anual do MPS, Sessão de Ferramentas*, pp. 174-182, Campinas, SP, Brasil.

- SANTOS, R.P., SANTOS, P.S.M., SANTO, R.E., 2010c, “Projeto e Desenvolvimento de Sistemas de Informação para Gestão de Conteúdo na Web utilizando Java EE e JBoss Seam”. In: *Anais do VI Simpósio Brasileiro de Sistemas de Informação, Minicursos*, pp. 4.1-4.6, Marabá, PA, Brasil, Maio.
- SANTOS, R.P., WERNER, C.M.L., COSTA, H.A.X., VASCONCELOS, S., 2011, “Supporting Software Engineering Education through a Learning Objects and Experience Reports Repository”. In: *Proceedings of the 23rd International Conference on Software Engineering and Knowledge Engineering*, pp. 272-275, Miami, FL, USA, July.
- SANTOS, R.P., ESTEVES, M.G.P., FREITAS, G., SOUZA, J.M., 2012a, “Utilizando Redes Sociais Para Apoiar os Ecossistemas de Software”. In: *Anais do XXXII Congresso da Sociedade Brasileira de Computação, I Brazilian Workshop on Social Network Analysis and Mining*, pp. 1-4, Curitiba, PR, Brasil.
- SANTOS, R.P., WERNER, C.M.L., COSTA, H.A.X., ABÍLIO, R.S., BORGES, H.S., 2012b, “Managing Reusable Learning Objects and Experience Reports in EduSE Portal”. In: *Proceedings of the 13th IEEE International Conference on Information Reuse and Integration*, pp. 631-638, Las Vegas, NV, USA.
- SANTOS, R.P., WERNER, C.M.L., BARBOSA, O.A.L.P., ALVES, C.F., 2012c, “Software Ecosystems: Trends and Impacts on Software Engineering”, In: *Anais do XXVI Simpósio Brasileiro de ES, Trilha Especial “Grandes Desafios da ES/Sistema”*, pp. 206-210, Natal, RN, Brasil, Setembro.
- SANTOS, R.P., MAGDALENO, A.M., RODRIGUES, C.S.C., WERNER, C.M.L., 2012d, “Tecnologia da Informação para Rede de Pesquisa”. In: EGLER, T.T.C.; TAVARES, H.M. (org.), *Política Pública, Rede Social e Território*, pp. 259-288, 1. ed., Rio de Janeiro, RJ, Brasil, Letra Capital.
- SANTOS, R.P., WERNER, C.M.L., ALVES, C.F., PINTO, M.J., CUKIERMAN, H.L., OLIVEIRA, F.M., EGLER, T.T.C., 2013a, “Ecossistemas de Software: Um Novo Espaço para a Construção de Redes e Territórios envolvendo Governo, Sociedade e a Web”. In: EGLER, T.T.C. et al. (org.), *Política Pública, Rede Social e Território II*, 1. ed., Rio de Janeiro, RJ, Brasil. (submetido)
- SANTOS, R.P., ESTEVES, M.G.P., FREITAS, G. S., SOUZA, J.M., 2013b, “Using Social Networks to Support Software Ecosystems Comprehension and Evolution”, *Social Networking*, v. 1. (submetido)

- SANTOS, R.P, TOSTES, L.M.M., WERNER, C.M.L., 2013c, “A Brechó-EcoSys extension to support negotiation in the software ecosystems context”, In: *Proceedings of the 14th IEEE International Conference on Information Reuse and Integration*, pp. 578-585, San Francisco, CA, USA, August.
- SBCUP, 2012, “IV Simpósio Brasileiro de Computação Ubíqua e Pervasiva”. Disponível em < <http://www.imago.ufpr.br/csbc2012/sbcup.php>>. Acesso em: 20 jan. 2013, 17:00:00.
- SCHILLING, M.A., 2008, ***Strategic Management of Technological Innovation***. 2. ed., McGraw Hill.
- SCHNEIDER, D., SOUZA, J.M., MORAES, K., 2011, “Multidões: A Nova Onda do CSCW?”, In: *Anais do VIII Simpósio Brasileiro de Sistemas Colaborativos*, pp. 24-31, Paraty, RJ, Brasil, Outubro.
- SEACORD, R.C., 1999, “Software Engineering Component Repositories”, In: *Proceedings of the International Workshop on Component-Based Software Engineering, 21st International Conference on Software Engineering*, pp. 1-7, Los Angeles, CA, USA, May.
- SEICHTER, D., DHUNGANA, D., PLEUSS, A., HAUPTMANN, B., 2010, “Knowledge Management in Software Ecosystems: Software Artifacts as First-class Citizens”, In: *Proceedings of the 4th European Conference on Software Architecture, 2nd International Workshop on Software Ecosystems*, pp. 119-126, Copenhagen, Denmark, August.
- SHAPER, D., 1974, “Scientific Theories and their Domains”. In: SUPPE, F. (ed), *The Structure of Scientific Theories*, pp. 518-565, Urbana, IL, USA, University of Illinois Press.
- SHAW, M., GARLAN, D., 1996, ***Software Architecture: Perspectives on an Emerging Discipline***. Prentice Hall.
- SILVA, F.Q.B., CÉSAR, A.C.F., 2009, “An Experimental Research on the Relationships between Preferences for Technical Activities and Behavioural Profile in Software Development”, In: *Proceedings of the XXIII Brazilian Symposium on Software Engineering*, pp. 126-135, Fortaleza, CE, Brazil, October.
- SMITH, T.M., SMITH, R.L., 2008, ***Elements of Ecology***. 7. ed., Benjamin Cummings.
- SOFTEX, 2007, *Perspectivas de Desenvolvimento e Uso de Componentes na Indústria Brasileira de Software e Serviços*, Relatório SOFTEX-MCT-DPCT/Unicamp.

- SOFTEX, 2012, *Guia Geral MPS de Software*, Sociedade SOFTEX, Agosto de 2012.
- SPÍNOLA, R., DIAS-NETO, A., TRAVASSOS, G., 2010, “Developing Software Technologies through Experimentation: Experiences from the Battlefield”. In: *Proceedings of the XIII Ibero-American Conference on Software Engineering*, pp. 1-10, Cuenca, Ecuador, April.
- STEFANUTO, G., ALVES, A.M., SPIESS, M., CASTRO, P.F.D., 2011, “Quality in Software Digital Ecosystems: The User Perceptions”, In: *Proceedings of the 3rd ACM/IFIP International Conference on Management of Emergent Digital EcoSystems*, pp. 85-88, San Francisco, CA, USA, November.
- SZYPERSKI, C., GRUNTZ, D., MURER, S., 2002, ***Component Software – Beyond Object-Oriented Programming***. 2. ed. New York, NY, USA, Addison-Wesley and ACM Press.
- TEIXEIRA, E.N., 2011, *ODYSSEYPROCESS-FEX: Uma Abordagem para Modelagem de Variabilidades de Linha de Processos de Software*. Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- TEIXEIRA, E.N., WERNER, C.M.L., VASCONCELOS, A.P., 2009, “An Approach to Support a Flexible Feature Modeling”. In: *Anais do III Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização de Software*, pp. 81-94, Natal, RN, Brasil.
- THOUGHWORKS, 2013, “*ThoughtWorks Technology Radar*”. Disponível em <<http://www.thoughtworks.com/radar>>. Acesso em: 20 jan. 2013, 17:00:00.
- TRAAS, V., HILLEGERSBERG, J., 2000, “The Software Component Market on the Internet Current Status and Conditions for Growth”, ***ACM SIGSOFT Software Engineering Notes***, v. 35, n. 1 (January), pp. 114-117.
- TRIST, E.L., 1977, “A Concept of Organizational Ecology”, ***Australian Journal of Management***, v. 2, n. 2 (October), pp. 161-175.
- TRIST, E.L., 1981, “*The Evolution of Socio-technical Systems*”, Occasional Paper No. 2 (June). Associate of the Ontario Quality of Working Life Centre, Toronto, Canada, 67p.
- USCHOLD, M., 1998, “Knowledge Level Modelling: Concepts and Terminology”, ***Knowledge Engineering Review***, v. 13, n. 1 (March), pp. 5-29.
- VAHIA, C.M., MAGDALENO, A.M., WERNER, C.M.L., 2011, “EvolTrack-SocialNetwork: Uma Ferramenta de Apoio à Visualização de Redes Sociais”. In:

- Anais do II Congresso Brasileiro de Software: Teoria e Prática, Sessão de Ferramentas*, pp. 7-13, São Paulo, SP, Brasil, Setembro.
- VENKATRAMAN, N., 1994, "IT-Enabled Business Transformation: From Automation to Business Scope Redefinition", *MIT Sloan Management Review*, v. 32, n. 2 (January), p. 73.
- WERNER, C.M.L., 1992, *Reutilização de Software no Desenvolvimento de Software Científico*. Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- WERNER, C.M.L., 2009, "Building Software Ecosystems from a Reuse Perspective", In: *Proceedings of the First International Workshop on Software Ecosystems, 11th International Conference on Software Reuse*, p. 3, Falls Church, VA, USA, September.
- WERNER, C.M.L., BRAGA, R.M.M., 2000, "Desenvolvimento baseado em Componentes". In: *Anais do XIV Simpósio Brasileiro de ES, Minicursos*, pp. 297-329, João Pessoa, PB, Brasil, Outubro.
- WERNER, C.M.L., BRAGA, R.M.M., 2005, "A Engenharia de Domínio e o Desenvolvimento Baseado em Componentes". In: Gimenes, I.M.S, Huzita, E.H.M. (org.), *Desenvolvimento Baseado em Componentes: Conceitos e Técnicas*, pp. 57-103, Rio de Janeiro, RJ, Brasil, Editora Ciência Moderna.
- WERNER, C., BRAGA, R., MATTOSO, M., MURTA, L., COSTA, M., PINHEIRO, R., OLIVEIRA, A., 2000, "Infraestrutura Odyssey: Estágio Atual", In: *Anais da VI Sessão de Ferramentas do XIV Simpósio Brasileiro de ES*, pp. 366-369, João Pessoa, PB, Brasil, Outubro.
- WERNER, C.M.L., CORRÊA, C.K.F., SANTOS, R.P., SCHOTS, M., MURTA, L.G.P., LYRA, F., 2010, "Adoption of Configuration Management in the Industry: Strategies and Lessons Learned", *CLEI Electronic Journal*, v. 13, n. 2 (August), paper 8, pp. 1-11.
- WERNER, C., MURTA, L., LOPES, M., DANTAS, A., LOPES, L. G., FERNANDES, P., PRUDÊNCIO, J. G., MARINHO, A., RAPOSO, R., 2007, "Brechó: Catálogo de Componentes e Serviços de Software", In: *Anais da XIV Sessão de Ferramentas do XXI Simpósio Brasileiro de ES*, pp. 24-30, João Pessoa, PB, Brasil, Outubro.
- WERNER, C., MURTA, L., MARINHO, A., SANTOS, R., SILVA, M., 2009, "Towards a Component and Service Marketplace with Brechó Library", In:

- Proceedings of the IADIS International Conference WWW/Internet 2009*, pp. 567-574, Rome, Italy, November.
- WERNER, C.M.L., SANTOS, R.P., 2009, “Gerência de Reutilização de Software”, In: *VIII Simpósio Brasileiro de Qualidade de Software, Minicursos*, Ouro Preto, MG, Brasil, Junho.
- WERNER, C.M.L., SANTOS, R.P., 2010, “Component-Based Software Engineering: Software Component Repositories”, In: *ICTAC School on Software Engineering, Tutorials*, Natal, RN, Brasil, Agosto.
- WERNER, C.M.L., SANTOS, R.P., 2012, “Ecossistemas de Software: Estágio Atual, Direções de Pesquisa e a Prática na Indústria de Software”, In: *Proceedings of the XV Ibero-American Conference on Software Engineering*, Buenos Aires, Argentina.
- WERNER, C.M.L., SANTOS, R.P., ALVES, C.F., 2012, “Ecossistemas de Software: Estágio Atual, Direções de Pesquisa e a Prática na Indústria de Software”, In: *Anais do III Congresso Brasileiro de Software: Teoria e Prática, Tutoriais*, pp. 12-13, Natal, RN, Brasil, Setembro.
- WIKIPEDIA, 2013, “Armazém de dados”. Disponível em <http://pt.wikipedia.org/wiki/Armazém_de_dados>. Acesso em: 20 jan. 2013, 17:00:00.
- XAVIER, J.R., 2001, *Criação e Instanciação de Arquiteturas de Software Específicas de Domínio no Contexto de uma Infraestrutura de Reutilização*. Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- YU, E. DENG, S., 2011, “Understanding Software Ecosystems: A Strategic Modeling Approach”, In: *Proceedings of the 3rd International Workshop on Software Ecosystems, 2nd International Conference on Software Business*, pp. 65-76, Brussels, Belgium, June.