Capítulo 4

Ecossistemas de Software no Desenvolvimento de Plataformas para Web, Redes Sociais e Multimídia

Rodrigo Santos[1] e Davi Viana[2]

[1] Universidade Federal do Estado do Rio de Janeiro (UNIRIO) – rps@uniriotec.br

[2] Universidade Federal do Maranhão (UFMA) – davi.viana@ufma.br

Abstract

A software ecosystem (SECO) is a set of actors and artifacts that exchange resources and information based on a common technological platform, in which external players are also included. This context has affected decisions on the management and development of those platforms in several domains, especially regarding the architecture, governance and collaboration models. As such, it is important to integrate mechanisms and tools to support the exchange of information, resources and artifacts, as well as to ensure an effective communication and interaction among organizations, developers and users. This chapter presents how the SECO reality changes the development of web, social networks and multimedia platforms.

Resumo

Um ecossistema de software (ECOS) é um conjunto de atores e artefatos, internos e externos a uma organização ou comunidade, que trocam recursos e informações centrados em uma plataforma tecnológica comum. Este contexto tem afetado decisões de gerenciamento e desenvolvimento de tais plataformas, notadamente sobre modelos de arquitetura, de governança e de colaboração, nos mais variados domínios de aplicação. É necessário integrar mecanismos e ferramentas para apoiar a troca de informações, recursos e artefatos, bem como assegurar a comunicação e interação dos desenvolvedores e usuários. O objetivo deste capítulo é apresentar como os ECOSs afetam o desenvolvimento de plataformas para web, redes sociais e multimídia.

4.1. Introdução

O desenvolvimento de plataformas para web, redes sociais e multimídia avançam cada vez mais em pesquisa teórica e aplicada, apresentando desafios que estão além das questões técnicas. A construção desses sistemas como produtos de software tem evoluído para o desenvolvimento de múltiplos produtos, derivados de uma plataforma baseada em uma arquitetura comum e integrados com outros sistemas por meio de redes de atores e artefatos [Manikas 2016]. Conforme apontado na primeira tese de doutorado do Brasil no assunto [Santos 2016], esse conjunto de elementos forma um ecossistema de software (ECOS) e requer a integração de mecanismos e ferramentas para apoiar a troca de informações, recursos e artefatos, bem como para assegurar a comunicação e interação dos desenvolvedores e usuários [Bosch 2012]. Nesse sentido, as redes sociais que se formam no ECOS, fortemente dependentes de web e multimídia, precisam ser analisadas a fim de auxiliar uma melhor concepção, gerenciamento e evolução de um ECOS [Hanssen e Dybå 2012]. Além disso, decisões de gestão e o monitoramento de tais plataformas dependem notadamente dos modelos de arquitetura, de governança e de colaboração, e do seu uso nos mais variados domínios de aplicação.

Pelas razões apresentadas, este capítulo tem grande relevância para a comunidade de Multimídia e Web por propiciar o contato com um tópico atual e de caráter prático na indústria (ECOS). Esta combinação pode auxiliar no tratamento de desafios gerados pela utilização de ECOS, como lidar com questões econômicas e sociais em conjunto com as questões técnicas, conforme [Boehm 2006] [Bosch 2009] [Seichter et al. 2010] [Santos et al. 2012] [Manikas 2016] [Santos 2016]. Dessa forma, o desenvolvimento de plataformas para web, redes sociais e multimídia é afetado pelos ECOS [Santos e Oliveira 2013], e conhecer este assunto é de grande importância para a comunidade de Web e Multimídia. Ademais, desdobramentos de utilização de ECOS podem contribuir para entender, analisar e resolver tais questões pelo fato de que este conceito vem sendo extensamente utilizado no mercado. Nos casos reais da indústria de software, por exemplo, existem ECOSs de dispositivos móveis, como Android e iOS; ECOS de gestão de conteúdo e comunidades na web, como o Moodle; ECOS de redes sociais, como Facebook; e ECOS de desenvolvimento colaborativo, como o Portal do Software Público Brasileiro (SPB).

O objetivo deste capítulo é apresentar como os ECOSs afetam o desenvolvimento de plataformas para web, redes sociais e multimídia. Para isso, os principais conceitos e estratégias de ECOS serão discutidos, seguidos pela discussão de aspectos que influenciam a análise do desenvolvimento de plataformas para web, redes sociais e multimídia. Alguns desafios enfrentados pela academia e pela indústria de software serão apontados nas considerações finais visando explorar alguns pontos importantes de se considerar em tais plataformas, como por exemplo a questão de transparência [Santos et al. 2016]. Para isso, este capítulo está organizado da seguinte forma: a Seção 4.2 apresenta a trajetória, conceitos principais e algumas estratégias para ECOS; na Seção 4.3, são discutidos como a Aprendizagem Organizacional e Gestão do Conhecimento podem influenciar o desenvolvimento de plataformas de ECOS para web, redes sociais e multimídia; por fim, a Seção 4.4 conclui o capítulo com algumas considerações finais pautadas em desafios para ECOS centrados nas plataformas supracitadas, bem como possíveis desdobramentos para trabalhos futuros.

4.2. Conceitos e Estratégias de Ecossistemas de Software

Nesta seção, o objetivo é apresentar alguns conceitos básicos de ECOS, isto é, a sua definição, elementos-chave e atores envolvidos no processo de desenvolvimento, a fim de compreender a interação entre atores e artefatos [Seichter et al. 2010]. A meta é mostrar a importância de cuidar das redes formadas em torno do desenvolvimento de plataformas e seus produtos e serviços em um ECOS, bem com a complexidade de lidar com isso devido a diversos fatores, entre eles o ciclo de vida social das redes criadas durante o desenvolvimento de plataformas para web, redes sociais e multimídia.

4.2.1. Dos Sistemas Monolíticos aos Ecossistemas

Software pode ser definido como um conjunto de instruções que proveem características, funções e desempenho desejados, quando tais instruções são executadas 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 [Messerschmitt e Szyperski 2003] [IEEE, 2004]. Entretanto, o software é um elemento de um sistema lógico, mais do que físico, o que o diferencia significativamente do hardware. Conforme Pressman (2010), software é desenvolvido ou "engenheirado", envolve pessoas e requer uma gestão de projetos efetiva para tratar os custos relacionados a isso. Além disso, o software não se desgasta como o hardware, mas se deteriora, sendo que, em sua maioria, continua sendo construído de maneira customizada e baseado em componentes e bibliotecas existentes.

Historicamente, esses apontamentos remontam a conferência do comitê de ciência da Organização do Tratado do Atlântico Norte em 1968 na qual, de acordo com Brooks (1995), apenas o artigo apresentado por Doug McIlroy tinha relação direta com a engenharia. 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 efetuar tarefas como classificação de itens ou ainda rotinas matemáticas. Essa 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 de trazer vários benefícios como o aumento da produtividade e a redução do esforço de desenvolvimento, os maiores obstáculos são de natureza gerencial, organizacional, econômica e técnica, ou seja, reutilizar demanda um ciclo contínuo de aprendizagem.

Nos últimos anos, com a globalização do desenvolvimento de software e os novos modelos de negócio e de colaboração, as plataformas para redes sociais, web e multimídia começaram a sofrer com as pressões de abertura para agregar novos envolvidos, externos à organização [Santos et al. 2014b]. Isso é chamado de inovação aberta [Chesbrough 2003] e tem motivado a transição da visão tradicional da engenharia de software para os ecossistemas, a fim de investigar as questões sociais e econômicas intrínsecas ao ciclo de vida das plataformas supracitadas. A partir desse discurso, pode-se definir uma trajetória do desenvolvimento de software monolítico para as plataformas dos ecossistemas, ilustrada pelas "4 gerações" apresentadas na Figura 4.1. Existem duas direções: (a) *Da ES tradicional para os ECOSs*, que considera a evolução histórica e o amadurecimento da engenharia de software nas últimas três décadas, em busca de

maximizar os benefícios prometidos ao se reutilizar software; e (b) *Dos ECOSs para a ES sustentável e diversificada*, que considera as diferentes dimensões que precisam ser levadas em consideração no desenvolvimento de plataformas para redes sociais, web e multimídia, visando a sua sustentabilidade[1] e diversidade[2].

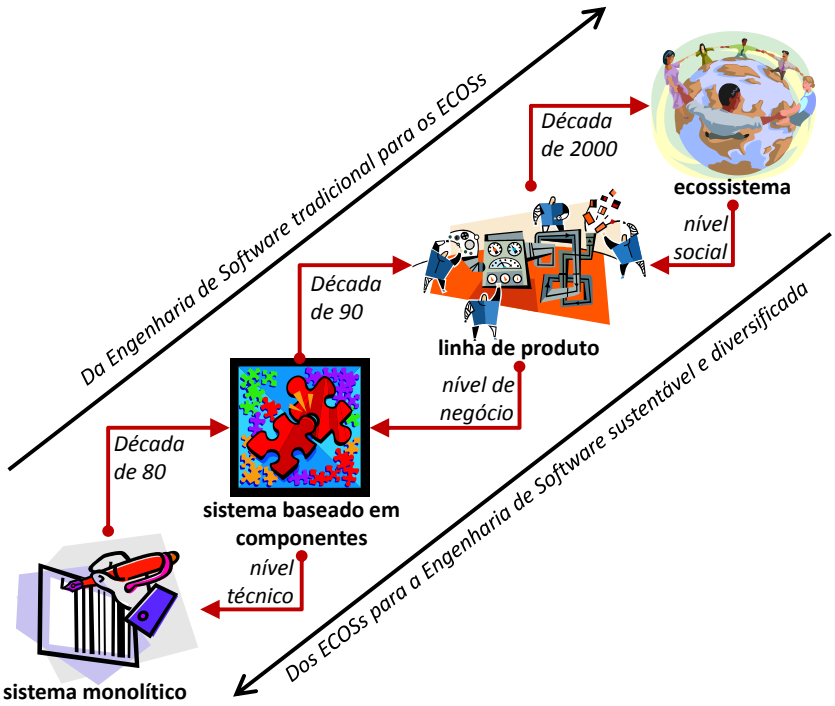

Figura 4.1. Evolução dos sistemas monolíticos para os ecossistemas. Adaptado de Santos (2016)

Em um primeiro momento, a principal reação contra a abordagem "codifica-e-remenda" da década de 1960 foi estabelecer processos em que a codificação era mais bem organizada, baseada na programação estruturada, que era precedida pela modelagem e engenharia de requisitos. Entre as 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, o que contribuiu para a construção dos ***sistemas monolíticos***: sistemas formados por conjuntos de rotinas que têm permissão para interagir livremente entre si. Melhorias foram buscadas em direção à produtividade e à escalabilidade. Por exemplo, organizações que despendiam 60% do seu esforço em testes descobriram que 70% destas atividades representavam retrabalho, podendo ser evitado ao estruturar as fases anteriores do ciclo de vida [Boehm 2006].

Por sua vez, os avanços da computação para melhorar a produtividade da década de 1980, tais como sistemas especialistas, linguagens de alto nível, orientação a objetos, estações de trabalho poderosas e programação visual, propiciaram a reutilização de software. Sistemas operacionais mais poderosos, sistemas de gerenciamento de banco de dados, *frameworks* para interface gráfica, *middleware* distribuído e automação de

[1]Sustentabilidade é a capacidade do ECOS sobreviver às diferentes perdas [Dhungana et al. 2010].

[2]Diversidade é a capacidade de prover diversas oportunidades no ECOS [Dhungana et al. 2010].

escritório permitiram superar a reutilização simples, do tipo "copia-e-cola" [Boehm 2006]. As linguagens de programação orientadas a objeto impulsionaram a prática de modularização. Segundo Biffl et al. (2006), no final dos anos 1980, abordagens de 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***.

Na década de 1990, com a expansão da Internet e da web, um mercado competitivo de software emergiu, de modo que o software passou a ser tratado como elemento de diferenciação no mundo dos negócios. Por exemplo, o caso da Hewlett Packard (HP) foi relatado por Poulin (1996) e mostrou que 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 o estudo de caso foi finalizado. No final da alguns elementos ganharam destaque na indústria: software crítico, gerência de dependências, serviços web, manutenção de sistemas legados, componentes de prateleira (COTS), desenvolmento *open source* e usabilidade. Para as organizações, estava claro o esforço para atingir os benefícios de reutilização, o que levou ao conceito de ***linha de produto de software*** no final dos anos 1990, inspirado pelo sucesso do conceito de linha de produção em organizações como a Toshiba [Boehm 2006]. Nesta abordagem, a organização organiza uma base de ativos reutilizáveis, isto é, artefatos de software, utilizada para desenvolver e compor sistemas.

As organizações que arcaram com os custos de adoção de uma linha de produtos alcançaram sucesso com a reutilização [Bosch e Bosch-Sijtsema 2010]. No entanto, McGregor (2010) afirma que a inovação passou a forçar a abertura das fronteiras da organização, de modo que modelos de negócio, processos, produtos, serviços, recursos, informações e artefatos sejam alvos de uma comunidade em torno de uma plataforma ou tecnologia de software central da organização, podendo gerar vantagem competitiva [Goldman e Gabriel 2005]. Isso significa que a plataforma pode estar disponível para atores externos à organização e, uma vez que ela decide abrir a sua fronteira, pode ocorrer a transição de linhas de produto para ***ecossistemas*** [Bosch 2009]. Para obter sucesso, as organizações precisam descobrir maneiras de explorar os desenvolvedores externos potenciais para torná-los parte de uma comunidade, pela combinação de licenças e recompensas bem definidas, o que requer compreender e adaptar modelos de negócio, além de modelar e analisar as redes socio-técnicas [Campbell e Ahmed 2010].

Bosch (2009) afirma que duas razões motivam uma organização a utilizar a abordagem de ECOS: (1) ela pode perceber que a quantidade de funcionalidades que ela precisa desenvolver para satisfazer as necessidades dos clientes e usuários é muito maior do que ela pode construir, considerando o investimento em P&D e o tempo disponível para um retorno adequado; e (2) 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 a plataforma) com o apoio de atores externos à organização parece ser um mecanismo efetivo. Entretanto, vale a pena destacar que a abordagem de ECOS ainda está em maturação na academia e na indústria: o primeiro registro do uso deste termo na área de engenharia de software remonta o ano de 2003 [Barbosa et al. 2013]. Apesar disso, as motivações continuam vigentes [Messerschmitt e Szyperski 2003] [Jansen et al. 2013]: o software é ubíquo, faz o ambiente interativo, é importante,

é sobre pessoas, pode ser melhor, faz parte de uma indústria em constante mudança, é uma criação social, é sofisticado e complexo, e pode ser controlado.

4.2.2. Conceitos Básicos de ECOS

Na última década, o estudo de ECOS tem se tornado um campo de pesquisa ativo e definido pelo aparecimento de novos modelos de arquitetura de software, de colaboração de desenvolvedores e de negócio abertos [Hanssen e Dyba 2012]. Isso contribui para a criação de comunidades interconectadas em redes complexas de atores e de organizações, interessadas em uma tecnologia de software central (ou plataforma), que oferece oportunidades para geração de valor por diversos atores [Hanssen 2012]. Hanssen e Dyba (2012) afirmam que a abordagem de ECOS representa um salto radical em como a engenharia de software está sendo feita, de maneira que o desenvolvimento está se tornando um processo aberto em um complexo ambiente distribuído. Barbosa et al. (2013) sumarizaram a maior parte dos termos e acrônimos utilizados para se referir a ECOS (Tabela 4.1), uma vez que diferentes grupos de pesquisa e da indústria têm investigado o tema de modo independente.

Tabela 4.1. Termos e acrônimos utilizados para ECOS na literatura. Fonte: [Barbosa et al. 2013]

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

Alguns exemplos de ecossistemas são o ECOS MySQL/PHP, o ECOS Eclipse, o ECOS Microsoft e o ECOS iPhone. Estes exemplos podem ser usados para estabelecer as características típicas de um ecossistema. Por exemplo, um ECOS *pode estar contido* em outro ECOS, 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 *ECOS abertos*, desde que as organizações tenham acesso ao código fonte e às bases de conhecimento, visando maior flexibilidade no desenvolvimento. Considerando as pesquisas básicas feitas por Santos e Werner (2010; 2011), Barbosa e Alves (2011), Hanssen e Dyba (2012) e Manikas e Hansen (2013), observou-se a existência de diversas definições para ECOS na literatura, embora a de Jansen et al. (2009) e a de Bosch (2009) sejam as mais citadas pelos estudos publicados desde 2003:

> *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.* [Jansen et al. 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. O foco está nos aspectos organizacionais do ECOS e nas relações entre as organizações enquanto entidades de software e serviços. [Bosch 2009]

Apesar das numerosas definições existentes e de pelo menos 40 publicações não citar nenhuma delas conforme resultados de uma ampla revisão da literatura no assunto [Manikas e Hansen 2013], dois conceitos são ditos fundamentais em ECOS: (1) um interesse comum em uma *tecnologia de software central*, isto é, sistemas, serviços ou plataforma de software; e (ii) uma *rede de organizações*, isto é, algum tipo de relacionamento, seja simbiótico, ou de evolução conjunta, comercial ou técnica. Nesse contexto, os diferentes relacionamentos que a organização pode ter com tecnologia de software central, os três papéis principais são apontados na Tabela 4.2. Outros papéis compreendem organizações de padronização, revendedores e operadores de sistemas [Jansen et al. 2009]. Cada um dos atores pode se beneficiar de suas participações no ECOS, e o espectro de relacionamentos simbióticos depende de seus papéis e atividades.

De acordo com Manikas e Hansen (2013), pelo menos cinco relacionamentos podem ser identificados em ECOS: dois atores podem (1) ter benefícios mútuos (*mutualismo*); (2) estar em competição direta (*competição/antagonismo*); (3) não ser afetados (*neutralismo*); ou (4) um não ser afetado enquanto o outro é beneficiado (*amensalismo*) ou (5) prejudicado (*parasitismo*) pelo relacionamento. Em resumo, existem três elementos chave em um ECOS [Santos e Werner 2010]: (i) *software*, como plataforma tecnológica comum, tecnologia de software central, soluções de software, plataforma de software ou linha de produto; (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 comunidades *open source*; e (iii) *relacionamentos*, como conexões entre atores com base nos elementos (i) e (ii).

Tabela 4.2. Principais papéis em um ECOS.
Adaptado de Hanssen (2012)

PAPEL	DESCRIÇÃO
Organização Chave ou Dono da Plataforma (*Keystone*)	Uma organização, ou um pequeno grupo, que de alguma forma conduz o desenvolvimento da tecnologia de software central.
Usuários Finais ou Clientes (*End-users*)	Papel chave para a tecnologia de software central, pois representa quem precisa dela para realizar seu negócio, seja de qual tipo for.
Organizações Externas ou Desenvolvedores Externos (*Third-parties*)	Utilizam a tecnologia de software central como base para produzir soluções ou serviços relacionados.

Por fim, Manikas e Hansen (2013) listam 43 ecossistemas que têm sido explorados, dos quais 30 são discutidos cada um em uma publicação, 12 estão presentes em mais de uma publicação e um não foi nomeado, embora seja mencionado 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, Brasero, 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 Worcraft (um). Somente dois ECOSs são fechados (GX Software e SAP), o que pode refletir o desafio de conseguir dados desse tipo de ECOS, ao contrário dos ECOS abertos, que possuem repositórios públicos disponíveis, conforme discutido em estudos como [Santana e Werner 2013].

4.3. Análise de Aspectos que Influenciam o Desenvolvimento de Plataformas para Web, Redes Sociais e Multimídia

Analisar os diversos aspectos que influenciam o nascimento, desenvolvimento, amadurecimento e eventual "morte" de plataformas para web, redes sociais e multimídia mais complexas aparece como uma preocupação em ECOS [Santos e Werner 2010]. Mais especificamente, na medida em que as organizações "abrem" as suas estruturas de negócios para outras contribuírem, mais ecossistemas começam a se formar, o que requer uma análise mais cuidadosa dos diversos aspectos que influenciam o desenvolvimento dessas plataformas. Nesse sentido, um desafio está no gerenciamento da *diversidade de organizações e relações de um ECOS* criado em torno de programadores, fornecedores, parceiros e clientes/usuários [Lima et al. 2014]. Isso pode dificultar a identificação e compreensão dos papéis que cada ator possui no ECOS, bem como os impactos de suas ações e decisões no desempenho do ECOS como um todo. Além disso, é necessário *analisar e tratar o conhecimento sobre a plataforma e os produtos e serviços nela desenvolvidos*. Por fim, é necessário fazer com que *os diversos atores aprendam os conhecimentos para que melhorem sua produtividade*.

Diversas abordagens foram propostas com o objetivo de lidar com esses aspectos de influência durante o ciclo de vida de tais plataformas em ECOS. Essas abordagens podem auxiliar as organizações no tratamento do conhecimento organizacional. Primeiramente, Campbell e Ahmed (2010) afirmam que o conceito de ECOS tem raiz nas teorias do desenvolvimento de plataformas similares (web e multimídia) e das redes sociais e pode ser um caminho para a transição, evolução e inovação na engenharia de software. Para explorar isso, os autores apresentam uma visão de ECOS em três dimensões, a fim de entender as funções assumidas por cada um de seus pilares:

- *arquitetura*: envolve a plataforma em que o ECOS está inserido, assim como questões da arquitetura de software, linha de produto e processos de software;
- *negócio*: envolve o conhecimento sobre o mercado, decisões que os atores devem tomar sobre modelos de negócio, definição do portfólio de produtos do ECOS, e 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 por meio de uma proposição de valor onde todos possam obter ganhos.

Em outra perspectiva, Jansen et al. (2009) discutiu que o ciclo comercial de um ECOS pode ser analisado em quatro fases, uma vez que a dimensão "negócio" envolve uma transação entre atores: (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. Nesse contexto, as dimensões "arquitetura" e "negócio" de Campbell e Ahmed (2010) são normalmente alvos de controle pela organização chave, mas a dimensão "social" é dinâmica. Ou seja, esta dimensão é a "pedra angular" para o sucesso do ECOS ao afetar as fases do seu ciclo de vida, i.e., nascimento, desenvolvimento, amadurecimento e morte/transformação. Avançando nesta discussão, Santos et al. (2013) apresentam uma generalização das três dimensões para tratar transações não comerciais:

- *técnica*: trata o entendimento da abertura das plataformas e do envolvimento de atores externos. É dirigida por três fatores: (a) *engenharia da plataforma*: definição, modelagem e construção do sistema; (b) *arquitetura*: estrutura dos componentes do sistema, relacionamentos e princípios e diretrizes para sua evolução; e (c) *gerência de requisitos*: funcionalidades específicas e comuns;
- *transacional*: trata o entendimento da colaboração ou competitividade, das expectativas dos envolvidos e do impacto na produtividade do seu ambiente. É dirigida por três fatores: (a) *visão do ambiente*: realizar planos e estar aberto a ações situadas; (b) *inovação*: usar a criatividade e atuar conforme um segmento; e (c) *planejamento estratégico*: seguir objetivos e ousar oportunidades;
- *social*: trata 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: (a) *utilidade*: compensações e recompensas esperadas e percebidas pelos membros da rede, financeiras ou não; (b) *promoção*: reconhecimento implícito ou explícito de capacidades e habilidades do colaborador; e (c) *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 4.2), estendido por Santos et al. (2014a) a partir de [Russ 2007] e [Jansen et al. 2009]:

- *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 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 (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 e/ou diversidade de um ECOS.

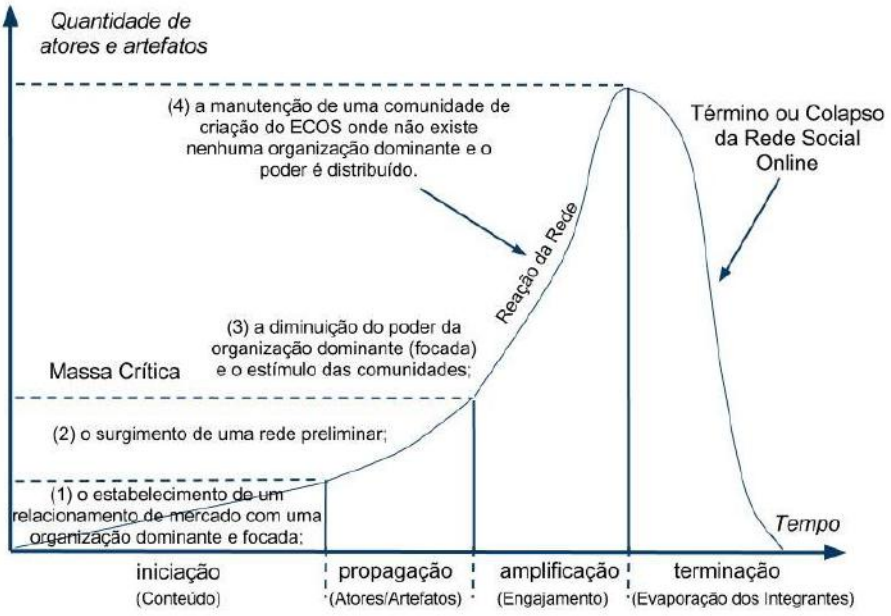

Figura 4.2. Ciclo de vida social de um ECOS. Adaptado de Santos et al. (2014a)

As redes sociais em um ECOS 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. 2013]. Fica claro que as organizações que não acompanham a rapidez da inovação de seus mercados viabilizada pelas redes em um ECOS podem não conseguir atravessar as barreiras cada vez mais altas [Chesbrough 2003]. 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.

Tal ponto de vista motivou Santos et al. (2013) a expandirem o ciclo de vida social para classificar as organizações sob o ponto de vista da adoção de um modelo de inovação em ECOS, quando um novo ator pode ser incorporado à fase "propagação": o **intermediário** (*technology broker*). Os intermediários têm o papel de: (i) facilitar o encontro entre oferta e demanda; (ii) agregar os atores certos a fim de fomentar a inovação; e (iii) apoiar e coordenar o trabalho colaborativo entre os diferentes níveis [Schilling 2008]. Esse papel pode ser analisado a partir da Figura 4.3 e da Figura 4.4, em que a primeira ilustra o processo da inovação fechada em um sistema tradicional (cada organização concebe suas inovações internamente) e a segunda ilustra o processo de inovação aberta (existem intermediários para agir como mediadores ou avaliadores dos diferentes ecossistemas e gerar relatórios ou realizar ações).

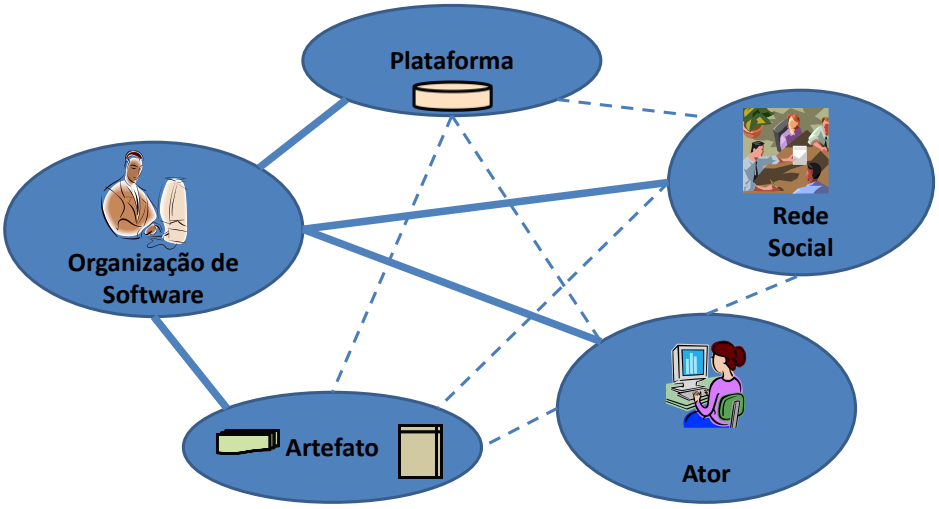

Figura 4.3. Fontes de inovação fechada em um sistema tradicional, onde as linhas pontilhadas são relações que não estão estabelecidas claramente, dado que a organização de software centraliza as relações. Adaptado de Santos et al. (2014a)

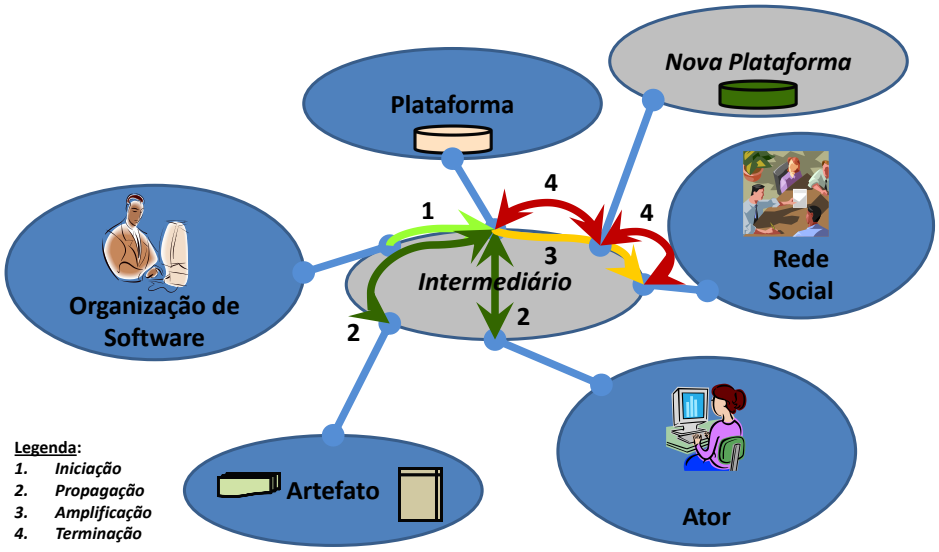

Figura 4.4. Fontes de inovação aberta em um ECOS, incorporando aspectos do ciclo de vida social. Adaptado de Santos et al. (2014a)

Por fim, Manikas e Hansen (2013) classificam os ECOS quanto a sua missão em *proprietários* e/ou *abertos*. 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 para participar do ecossistema. No ECOS aberto tradicional, os atores não necessariamente participam do ecossistema para obter rendimentos financeiros por suas atividades, sendo mais fácil colaborar, pois tipicamente não requer qualquer atestado de novos atores. Os autores ainda observam que ecossistemas abertos tratam mais as questões de natureza técnica e social, ao passo que aqueles proprietários incluem questões estratégicas e de negócio. Uma das razões é que ecossistemas abertos permitem o uso de técnicas de mineração e de processamento sobre código, *logs* de *commits* etc., embora careça de um modelo de negócio para participação de atores. Por outro lado, ECOSs proprietários conseguem prontamente informação sobre as estratégias do ecossistema e seu posicionamento no mercado, mas o acesso ao código fonte e *commits* de desenvolvedores é mais difícil.

Entre outros aspectos que influenciam o desenvolvimento de plataformas para redes sociais, web e multimídia, parte das abordagens está voltada para o uso direto de práticas de engenharia de software, e outra 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, devendo-se tratar gerenciamento, regras de negócio e restrições, integração e existência de múltiplas funcionalidades [Cataldo e Herbsleb 2010]. Segurança e confiabilidade consistem em instrumentos para equilibrar modularidade e flexibilidade na arquitetura da plataforma a fim de assegurar 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 e 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.

A análise e projeto arquitetural passam a incluir princípios como identificar metas de negócio e descrever requisitos significativos, táticas e avaliação arquitetural. O processo de elicitação de requisitos se torna um desafio relevante em ECOS, pois os novos atores se juntam aos tradicionalmente envolvidos no desenvolvimento, alguns deles numerosos e distantes do gerenciamento central do ECOS [Fricker 2009]. Cadeias de valor e cadeias de resultados passam a apoiar a engenharia de requisitos [Yu e Deng 2011], além da modelagem, análise e visualização de redes sociais, técnicas e socio-técnicas para identificar redes de influência e de interoperabilidade [Santos et al. 2013]. Bosch (2009) acrescenta mais duas questões: uso de processos ágeis e a composição de produtos e serviços em ecossistemas. Nesse contexto, a gestão do conhecimento e a aprendizagem no ECOS merecem ser analisados e são discutidos nas subseções a seguir.

4.3.1. Teoria de Criação do Conhecimento e Processo de Aprendizagem

Considerando o desenvolvimento de plataformas para redes sociais, web e multimídia, a gestão do conhecimento e a aprendizagem dos atores são cruciais, sobretudo em ECOS. Nesse sentido, o conhecimento sempre se origina nas pessoas sendo criado através da interação entre o Conhecimento Tácito e Explícito [Nonaka e Takeuchi 1995]. Essa interação deu origem ao modelo SECI (do inglês, *Socialization, Externalization,*

Combination e Internalization). Cada processo deste modelo representa uma conversão de conhecimento. No processo de socialização, o Conhecimento Tácito é compartilhado diretamente com outra pessoa. Esse conhecimento socializado pode não se tornar Explícito. No processo de externalização, o Conhecimento Tácito é convertido em Conhecimento Explícito. Neste caso, a organização possui a possibilidade de compartilhar o conhecimento com toda a organização. Já na etapa de combinação ocorre a combinação de componentes isolados do Conhecimento Explícito para a geração de um novo Conhecimento Explícito. A Figura 4.5 apresenta componentes do modelo SECI e o processo de aprendizagem criado através do modelo.

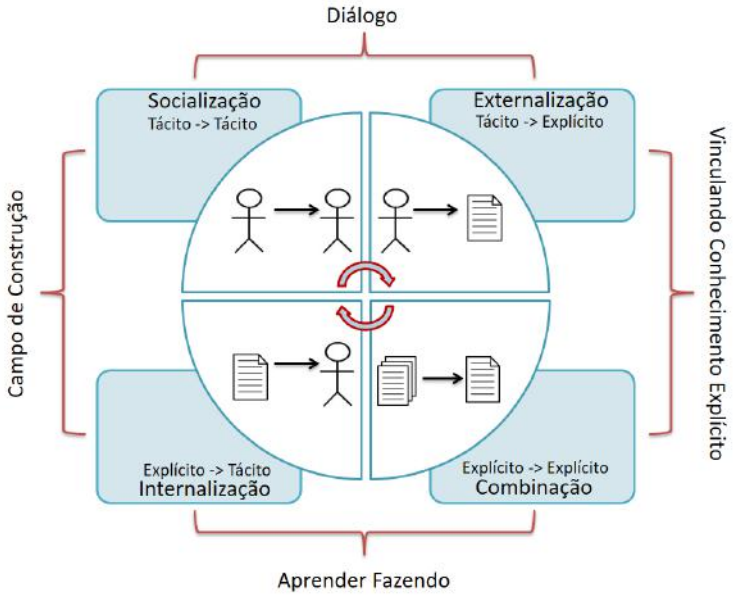

Figura 4.5. Modelo SECI e Processo de Aprendizagem. Adaptado de Nonaka e Takeuchi (1995)

Um processo de aprendizagem pode ser gerado a partir da realização dos quatro processos do modelo SECI. Ao executar os quatro processos, tem-se a espiral do conhecimento [Schneider 2009]. Inicialmente, é realizada uma socialização onde ocorre a troca de Conhecimento Tácito através de diálogo. Em seguida, esse conhecimento é externalizado por meio da escrita, além de ser combinado com outros conhecimentos através de vínculos entre os componentes do Conhecimento Explícito. O conhecimento externalizado é aplicado em alguma atividade de trabalho específica, neste momento ocorre o "aprender fazendo", pois a pessoa irá utilizar o Conhecimento Explícito de forma prática. Por fim, verifica-se que há um campo de construção do conhecimento, onde a base de Conhecimentos Tácitos é enriquecida. Neste momento, o processo de aprendizagem se inicia novamente [Nonaka e Takeuchi 1995]. Para que as organizações utilizem o conhecimento de forma eficaz, é necessário gerenciar tanto o Conhecimento Tácito quanto o Explícito [Hansen et al. 2000] [Choi e Lee 2003]. A utilização desses dois conhecimentos é importante para acumular o conhecimento corporativo e explorar novos potenciais [Choi e Lee 2003].

As plataformas para redes sociais, web e multimídia, bem como os produtos e serviços construídos a partir delas podem fazer uso deste processo, uma vez que é necessário transmitir o conhecimento gerado durante o desenvolvimento e fazer com

que novos colaboradores passem a cooperar com o ECOS criado. Por exemplo, Lopes et al. (2009) apresentam uma proposta de plataforma baseadas em *Service Oriented Architecture* (SOA), que permite que as funcionalidades sejam disponibilizadas e utilizadas como um conjunto de serviços descritos através de suas interfaces. O conceito de externalização do conhecimento pode auxiliar na disponibilização de conhecimentos sobre como utilizar a plataforma. De modo semelhante, o conceito de internalização pode apoiar na aprendizagem de utilização da plataforma.

4.3.2. Aprendizagem pelo Circuito Simples e Circuito Duplo de Argyris e Schön

Nesta abordagem, a aprendizagem é vista como um processo cíclico. Para Argyris e Schön (1978), a aprendizagem ocorre através da mudança de comportamento ao serem verificados os resultados de estratégias de ação e realização de correções. Essas estratégias de ação são atividades regidas por variáveis definidas a partir de objetivos e restrições. Quando as ações ou atividades são executadas, determinadas consequências ou resultados são identificados [Schneider 2009]. Existem duas formas de verificar a ocorrência da aprendizagem, são elas: circuito simples e circuito duplo. No circuito simples (do inglês, *single-loop learning*), a aprendizagem ocorre através da comparação entre as consequências das estratégias de ação executadas com as consequências desejáveis de uma estratégia de ação definida. Se houver um desvio entre as consequências planejadas e as reais, busca-se uma nova estratégia de ação dentro do conjunto de variáveis [Argyris e Schön 1978]. É importante frisar que esse circuito promove aprendizagem incremental [Souza-Silva e Davel 2007] e que pode ocorrer na mente das pessoas, ou através de uma simulação de duas alternativas [Schneider 2009].

Schneider (2009) apresenta um exemplo da aprendizagem de circuito simples aplicado ao desenvolvimento de sistemas. Quando um programador está trabalhando com uma linguagem que ele não está tão familiarizado e ocorre um erro de compilação (consequências reais), o programador irá alterar seu código fonte (estratégia de ação) para alcançar o seu resultado satisfatório (consequências desejáveis). Sendo que o resultado satisfatório é definido a partir do objetivo de obter um software com dado conjunto de requisitos (variáveis). Desta forma, entende-se que o programador aprendeu a não ocasionar mais o erro de compilação. Além disso, as consequências devem fornecer, sempre que possível, um *feedback*. Este *feedback* é um importante componente para que a aprendizagem ocorra de acordo com essa teoria, pois é através dele que é possível realizar melhorias no comportamento pessoal, como acontece em ECOS. A Figura 4.6 apresenta o processo da aprendizagem em circuito simples.

Figura 4.6. Aprendizagem de Circuito Simples (*single-loop learning*). Adaptado de Schneider (2009)

Em algum momento, pode ser necessária a mudança de alguma variável, objetivo e/ou restrições devido a questionamentos surgidos a partir do *feedback*. Quando isso ocorre, tem-se a possibilidade do ciclo duplo de aprendizagem (do inglês, *double-loop learning*). O processo é semelhante ao circuito simples, além das mudanças nas estratégias de ações (circuito simples), também é necessário verificar alterações nas variáveis (circuito duplo), conforme discute Argyris e Schön (1978). Neste caso, o conhecimento é adquirido neste nível mais alto, isto é, durante o circuito duplo. Levando em consideração o desenvolvimento de software, tem-se que o programador implementa um módulo. Se houver um comportamento diferente do esperado (consequências desejáveis), o programador pode alterar o módulo (circuito simples) ou pode começar a questionar as consequências desejáveis de modo que se analisem as possíveis alterações nos requisitos para satisfazer as consequências desejáveis (circuito duplo). Neste momento do circuito duplo, pode ocorrer à aprendizagem a partir do resultado atingido com a implementação do módulo [Schneider 2009]. A Figura 4.7 representa graficamente a ocorrência do ciclo duplo.

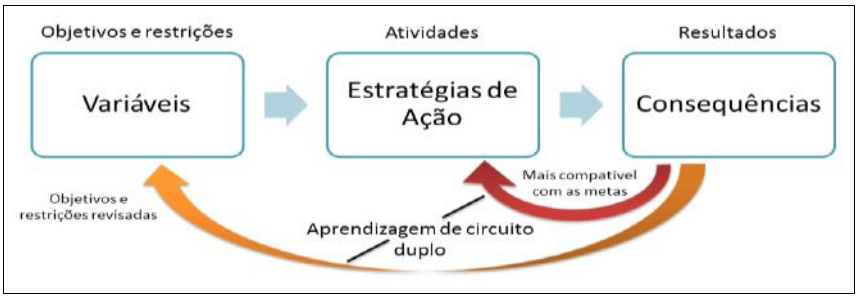

Figura 4.7. Aprendizagem de Circuito Duplo (*double-loop learning*). Adaptado de Schneider (2009)

4.3.3. Comunidades de Prática

Comunidades de prática são grupos de pessoas que compartilham os mesmos interesses e mesmos problemas em um tópico. Essas comunidades possuem como propósito principal a criação de conhecimento, possuindo conexões fracas, autogerenciamento e informalidades [Voss e Schafer 2003]. Segundo Miranda (2004), existem diversas definições para comunidades de prática. Contudo, todas apresentam que as comunidades de prática são estruturas sociais que têm facilidade em lidar com o conhecimento.

As pessoas que participam de uma comunidade de prática não trabalham necessariamente juntas todos os dias, mas elas se conhecem e se encontram devido ao valor encontrado nas suas interações [Wenger et al. 2002]. Essas interações incluem troca de informações, insights e conselhos, além de auxiliar na resolução de problemas, discutir situações e explorar ideias. As comunidades podem criar conhecimento explícito por meio de ferramentas, padrões, manuais e outros documentos, ou podem apenas desenvolver um entendimento tácito sobre o que está se compartilhando [Wenger et al. 2002]. Isso também acontece no contexto de ECOS.

A comunidade de prática é constituída de duas grandes partes: (a) *participação na comunidade*, interagindo e envolvendo-se nas iniciativas sociais; e (b) *concretização*, em que ocorre o processo de criar artefatos a partir dos resultados pela comunidade de prática, como documentos e através da interação face a face [Mestad et al. 2007].

Existem alguns indicadores que descrevem quando uma comunidade de prática foi criada [Mestad et al. 2007]. Esses indicadores são definidos por Wenger et al. (2002) e são características que podem ser necessárias para o sucesso de implementação deste tipo de comunidade. A Tabela 4.3 apresenta alguns exemplos desses indicadores.

Tabela 4.3. Indicadores de Wenger que descrevem a existência de uma comunidade de prática. Adaptado de Wenger et al. (2002)

1. Relacionamentos mutualmente sustentados.
2. Formas comuns de engajar-se para fazer as coisas juntos.
3. Fluxo rápido de informações e propagação de inovação.
4. Ausência de considerações introdutórias, as conversações e interações nas comunidades devem ser apenas a continuação de um processo em andamento.
5. Definição rápida de um problema a ser discutido.
6. Saber o que os outros sabem, o que eles podem fazer, e como eles podem contribuir para uma iniciativa.
7. Capacidade de avaliar a adequação das ações e produtos.
8. Ferramentas específicas, representações e outros artefatos.
9. Conhecimento local, histórias compartilhadas.
10. Jargões e atalhos para comunicação.
11. Certos estilos reconhecidos por todos os membros.
12. Discurso compartilhado refletindo certa perspectiva.

A aprendizagem nas comunidades de prática é estimulada através dos relacionamentos estabelecidos, como em ECOS. No contexto do desenvolvimento de plataformas para redes sociais, web e multimídia, as comunidades de prática são utilizadas para apoiar o acesso a especialistas e recursos de informação. Além disso, podem ser aplicadas para contribuir com o compartilhamento de conhecimento sobre tecnologias e, assim, contribuir para a aprendizagem e melhorar a produtividade durante o desenvolvimento de plataformas web e redes sociais. Esta teoria de comunidades de prática também serve como base para o desenvolvimento de sistemas de *e-discourses* [Voss e Schafer 2003]. *E-discourse* é um processo de comunicação argumentativa orientada a objetivos que é apoiado por uma ferramenta. Através dessa comunicação, as pessoas podem contribuir com ideias e perspectivas para soluções de problemas.

4.3.4. Teoria das Redes de Aprendizagem

A forma com que a aprendizagem é organizada no desenvolvimento de software é descrita pela teoria das redes de aprendizagem [van der Krogt 1998]. As pessoas aprendem em todas as organizações, seja uma empresa com estrutura hierárquica ou caótica, como acontece no contexto de ECOS. A rede de aprendizagem consiste em várias atividades de aprendizagem organizadas pelos membros da empresa, possuindo três grandes componentes [Škerlavaj e Dimovski 2006]:

- **Processo de Aprendizagem:** desenvolvimento de políticas e programas de aprendizagem e forma de execução dos programas;
- **Estruturas de aprendizagem:** estrutura do conteúdo para aprendizagem, estrutura organizacional e o clima da empresa para aprendizagem;
- **Atores:** colaboradores em todos os níveis.

Os atores da aprendizagem interagem uns com os outros para organizar as atividades do processo de aprendizagem. Com o passar do tempo, certos padrões estabelecidos inicialmente são desenvolvidos. Esses padrões formam as estruturas de aprendizagem [Poell et al. 2000]. A forma da rede de aprendizagem depende da dinâmica dos atores envolvidos e das características de trabalho da organização [van der Krogt 1998] [Poell et al. 2000]. Existem quatro tipos teóricos de redes de aprendizagem: liberal, vertical, horizontal e externa. No tipo *liberal*, normalmente a estrutura da aprendizagem é orientada a indivíduos, ou seja, cada colaborador pode ter sua forma de aprender. No tipo *vertical*, os sistemas de aprendizagem são auxiliados por planos de política e programas de aprendizagem elaborados. Por outro lado, no tipo *horizontal*, a aprendizagem é orientada a organização como um todo. Por fim, no tipo *externa*, a aprendizagem é estruturada de acordo com os profissionais externos (consultores). Para analisar os tipos de rede de aprendizagem pode ser utilizada a análise de redes sociais [Wasserman e Faust 1994]. A Tabela 4.4 apresenta um maior detalhamento dos quatro tipos teóricos de redes de aprendizagem.

Tabela 4.4. Quatro tipos teóricos de redes de aprendizagem. Adaptados de Škerlavaj e Dimovski (2006) e van der Krogt (1998)

	Tipos de Redes de Aprendizagem			
	Liberal	Vertical	Horizontal	Externa
Processo de Aprendizagem	Atividades simples	Linearmente planejado	Orientados a organização	Coordenado externamente
Desenvolvimento de políticas de aprendizagem	Implícito	Planejado	Aprendizagem	Inspiradora
Desenvolvimento de programas de aprendizagem	Coletar	Projetar	Desenvolver	Inovar
Execução dos programas de aprendizagem	Auto direcionada	Guiada	Aconselhada	Consultiva
Estrutura de Aprendizagem				
Estrutura do conteúdo para aprendizagem	Não estruturado (orientado a indivíduos)	Estruturado (orientado a atividades ou funções)	Aberta ou temática (orientada a organização ou problema)	Sistemática (orientadas a profissionais)
Estrutura Organizacional	Fracamente acoplada (contratos)	Centralizada (Formalizada)	Horizontal (igualitária)	Direcionada externamente (profissional)
Clima Organizacional para a aprendizagem	Liberal	Regulativa (regras e leis)	Integrativa (Aprendizagem e trabalho)	Inspiradora
Atores	Individuais	Oficiais, especialistas	Grupos	Atores externos

4.3.5. Ciclo de Aprendizagem de Kolb

Por fim, a teoria da aprendizagem experimental é descrita por Kolb (1984) como sendo o processo por onde o conhecimento é aprendido através da transformação da experiência, algo importante no contexto de ECOS. Neste ciclo, a "experiência concreta" dos indivíduos é base para a "observação reflexiva" que suporta reflexões na mente desses indivíduos. A observação reflexiva é necessária ao processo de criação de conclusões mais genéricas ou abstratas, gerando assim a "conceituação abstrata". Essas conclusões ou abstrações são aplicadas em mais situações, gerando assim a

"experimentação ativa". Durante a experimentação ativa, novas implicações podem ser testadas, criando novas experiências.

Contudo, Kolb descreve que nem sempre o indivíduo consegue passar por todas as fases do ciclo. Neste método, o ciclo de Kolb busca dar um sentido às experiências dos indivíduos, grupos e organizações por meio da experiência, reflexão, conclusão e experimentação. De forma paralela, os componentes do ciclo de Kolb auxiliaram no desenvolvimento dos quatro estilos de aprendizagem: divergência, assimilação, convergência e acomodação. Cada estilo representa uma combinação de dois componentes do ciclo de aprendizagem. A Figura 4.8 apresenta o ciclo de aprendizagem de Kolb juntamente com os estilos de aprendizagem. Esses estilos são encontrados nas comunidades de ECOS.

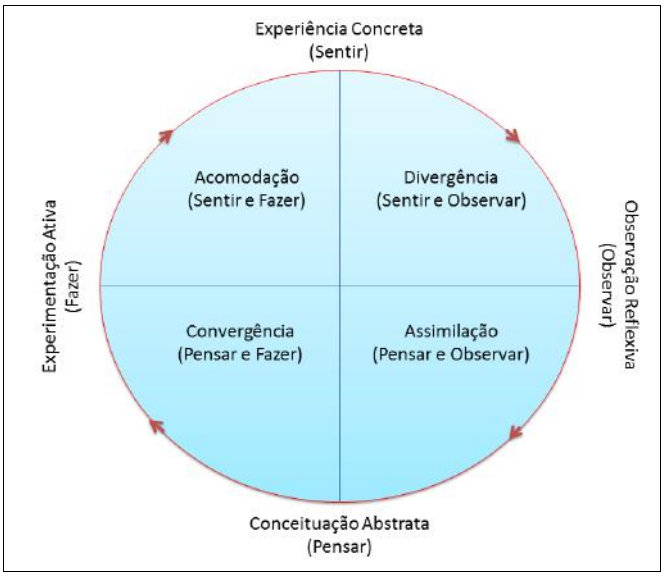

Figura 4.8. Ciclo de aprendizagem e estilos de aprendizagem. Adaptado de Kolb et al. (2001)

No estilo de aprendizagem *divergência*, as pessoas são hábeis em observar as coisas de diferentes perspectivas. Elas preferem observar a fazer, com o propósito de obter informações e usar a imaginação para resolver os problemas. Neste estilo, as pessoas preferem trabalhar em grupos e são emotivas. No estilo *assimilação*, existe uma abordagem concisa e lógica. Ideias e conceitos são mais relevantes que as pessoas. Estas pessoas requerem uma explicação boa e clara ao invés de prática. Pessoas que possuem este estilo preferem ler e explorar modelos teóricos. Já no estilo *convergência*, as pessoas podem resolver problemas e usarão sua aprendizagem para encontrar soluções para as questões práticas. As pessoas que possuem este estilo são mais abertas a experimentar novas ideias, simular e realizar atividades práticas. Por fim, no estilo *acomodação*, as pessoas preferem inicialmente praticar e tirar suas conclusões com base na experiência obtida. Também existe a preferência por trabalhar em equipes para completar atividades [Kolb et al. 2001]. Uma pessoa não possui somente um estilo de aprendizagem, pode ter tendências claras para um ou mais estilos de aprendizagem.

4.4. Considerações Finais

Dado o papel relevante do conceito "ecossistema" na engenharia de software do século XXI, este capítulo apresentou como ECOSs afetam o desenvolvimento de plataformas para web, redes sociais e multimídia. Neste contexto, é importante ressaltar que a inovação é constante em um ECOS e que diferentes tecnologias podem se unir para fortalecer as redes de produção de software que se formam em torno de uma plataforma e dos seus envolvidos. A integração de tecnologias e serviços através da web e mobile já é uma realidade, mas novas possibilidades vêm sendo exploradas. 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 em ECOS pode ser ainda mais estimulada pela adoção de software social (e.g., Wikis, blogs etc.), que permite aos atores interagir e compartilhar conhecimento em uma dimensão social, realçando o potencial humano de criação, em vez de privilegiar a transmissão um para muitos. Nesse contexto, diversos aspectos de influência precisam ser analisados ao longo do desenvolvimento das plataformas supracitadas, conforme investigado e discutido na Seção 4.3.

Alguns desafios e fatores de sucesso foram apontados por Bosch (2009) e Jansen et al. (2009), resumidos na Tabela 4.5: (a) caracterizar e modelar ECOS, 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; (b) estabelecer os relacionamentos entre as redes de produção de software; (c) gerenciar a qualidade em ECOS; (d) planejar portfólios e linhas de produtos em ECOS; (e) prover métodos, técnicas e ferramentas para desenvolver arquiteturas de sistemas que atendam à extensibilidade, à portabilidade e à variabilidade; e (f) tratar implicações gerais em engenharia de software, incluindo mecanismos de comunicação, agilidade na concepção e desenvolvimento de soluções e composição de produtos e serviços.

Tabela 4.5. Desafios para ECOS.
Adaptado de Bosch (2009) e Jansen et al. (2009)

NÍVEIS	DESAFIOS	DESCRIÇÃO
Organizacional	#1: *Portfólio e plano da linha de produção*	Gerenciar configurações e reutilização para funcionalidades da plataforma.
	#2: *Gestão do conhecimento*	Aplicar mineração para extrair e visualizar *feedback* dos envolvidos.
	#3: *Arquitetura estensível, portável e variável*	Explorar interfaces, requisitos e comunicação sobre plataformas.
	#4: *Integração de sistemas na organização*	Desenvolver *frameworks* para gerência do produto de software e seus ciclos.
Rede de Produção	#1: *Estabelecimento das relações*	Identificar papéis e relacionamentos.
	#2: *Tempo de liberações*	Gerenciar versões e seus impactos.
	#3: *Gestão de qualidade*	Gerar diretrizes para certificação.
Ecossistema	#1: *Caractetização e modelagem de ECOS*	Identificar tamanho, vizinhança, padrões e papéis no ECOS.
	#2: *Políticas e estratégias entre ECOSs*	Determinar instrumentos para permitir orquestração em cada ECOS.
	#3: *Estratégia para vencer e lucrar na rede*	Integrar modelos de negócio ao ECOS.

Por fim, pode-se destacar dois pontos de vista, como mostra a Figura 4.9 [Barbosa et al. 2013]. Do ponto de vista do *gerenciamento*, tem-se: (i) como lidar com

ECOS do tipo *open source*, i.e., plataformas abertas e as redes constituídas; (ii) como implementar governança de ECOS, i.e., gerir estratégias de sobrevivência e agregação da comunidade em uma plataforma diante de competidores e colaboradores; e (iii) como realizar a análise de ECOS, i.e., identificar, coletar, relacionar e extrair informação a partir de redes constituídas em uma plataforma. Por outro lado, do ponto de vista da *engenharia*, tem-se: (i) como abrir a plataforma, i.e., permitir que atores externos à organização chave contribuam para a evolução da plataforma (*transparência*); (ii) como manter a qualidade da plataforma, i.e., verificar se a plataforma continua atendendo às necessidades da comunidade, bem como dos atores e relacionamentos que a compõem, considerando suas funcionalidades e componentes; e (iii) como modelar a arquitetura da plataforma, i.e., planejar o processo de construção e evolução da plataforma considerando questões técnicas, modelos transacionais e redes sociais.

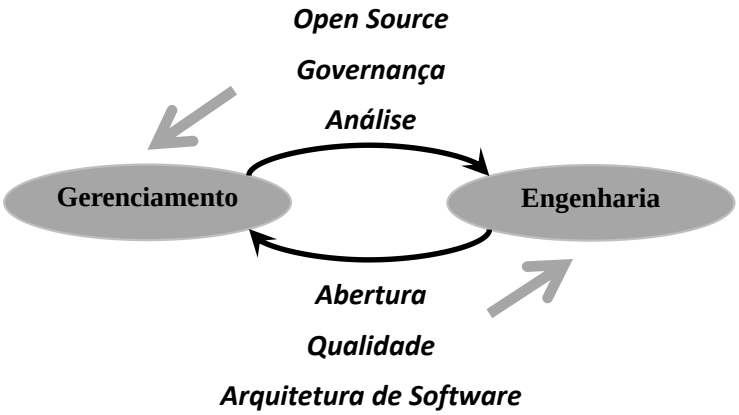

Figura 4.9. Desafios em ECOS.
Adaptado de Barbosa et al. (2013)

Agradecimentos

O primeiro autor gostaria de agradecer ao CNPq (Proc. No. PDJ 150539/20016-9) pelo apoio financeiro para realização deste trabalho.

Referências

Argyris, C., Schön, D.A. (1978) "Organizational Learning: A Theory of Action Perspective". Jossey-Bass.

Barbosa, O.A.L.P., Alves, C.F. (2011) "A Systematic Mapping Study on Software Ecosystems". In: Proceedings of the 3rd International Workshop on Software Ecosystems (IWSECO) – 2nd International Conference on Software Business (ICSOB), Brussels, Belgium, pp. 15-26, June.

Barbosa, O.A.L.P., Santos, R.P., Alves, C.F., Werner, C.M.L., Jansen, S. (2013) "A Systematic Mapping Study on Software Ecosystems from a Three-Dimensional Perspective". In: Jansen, S., Brinkkemper, S., Cusumano, M.A. (Org.), Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry, pp. 59-81, Edward Elgar Publishing.

Biffl, S., Aurum, A., Boehm, B., Erdogmus, H., Grünbacher, P. (2006) "Value-Based Software Engineering". Springer-Verlag.

Boehm, B. (2006) "A View of 20th and 21st Century Software Engineering". In: Proceedings of the 28th International Conference on Software Engineering (ICSE), Shanghai, China, pp. 12-29, May.

Bosch, J. (2009) "From Software Product Lines to Software Ecosystem". In: Proceedings of the 13th International Software Product Line Conference (SPLC), San Francisco, USA, pp. 1-10, August.

Bosch, J. (2010) "Architecture Challenges for Software Ecosystems". In: Proceedings of the 4th European Conference on Software Architecture (ECSA) – 2nd International Workshop on Software Ecosystems (IWSECO), Copenhagen, Denmark, pp. 93-95, August.

Bosch, J. (2012) "Software Ecosystems: Implications for Strategy, Business Model and Architecture". In: Proceedings of the 34th International Conference on Software Engineering (ICSE), Tutorials, Zurich, Switzerland, June.

Bosch, J., Bosch-Sijtsema, P. (2010) "From Integration to Composition: On the Impact of Software Product Lines, Global Development and Ecosystems". The Journal of Systems and Software 83(1):67-76.

Brooks, F.P. (1995) "The Mythical Man-Month: Essays on Software Engineering". Addison-Wesley Professional, 2nd Ed.

Campbell, P.R.J., Ahmed, F. (2010) "A Three-dimensional View of Software Ecosystems". In: Proceedings of the 4th European Conference on Software Architecture (ECSA) – 2nd International Workshop on Software Ecosystems (IWSECO), Copenhagen, Denmark, pp. 81-84, 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 (ECSA) – 2nd International Workshop on Software Ecosystems (IWSECO), Copenhagen, Denmark, pp. 65-72, August.

Chesbrough, H.W. (2003) "Open Innovation: The New Imperative for Creating and Profiting from Technology". Harvard Business School Publishing Corporation.

Choi, B., Lee, H. (2003) "An Empirical Investigation of KM Styles and their Effect on Corporate Performance". Information & Management 40(5):403-417.

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 (ECSA) – 2nd International Workshop on Software Ecosystems (IWSECO), Copenhagen, Denmark, pp. 96-102, August.

Fricker, S. (2009) "Specifications and Analysis of Requirements Negotiation Strategy in Software Ecosystems". In: Proceedings of the First International Workshop on Software Ecosystems (IWSECO) – 11th International Conference on Software Reuse (ICSR), Falls Church, USA, pp. 19-33, September.

Goldman, R., Gabriel, R.P. (2005) "Innovation Happens Elsewhere". Morgan Kaufmann.

Hansen, M., Nohria, N., Tierney, T. (2000) "What's your Strategy for Managing Knowledge". The Knowledge Management Yearbook, v. 2001, pp. 55-69.

Hanssen, G.K. (2012) "A Longitudinal Case Study of an Emerging Software Ecosystem: Implications for Practice and Theory". The Journal of Systems and Software 85(7):1455-1466.

Hanssen, G.K., Dybå, T. (2012) "Theoretical Foundations of Software Ecosystems". In: Proceedings of the 4th International Workshop on Software Ecosystems (IWSECO) – 3rd International Conference on Software Business (ICSOB), Cambridge, USA, pp. 6-17, June.

IEEE (2004) "Std 1517 - IEEE Standard for Information Technology - Software Life Cycle Processes - Reuse Processes". Institute of Electrical and Electronics Engineers.

Jansen, S., Cusumano, M., Brinkkemper, S. (2013) "Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry". Edward Elgar Publishing.

Jansen, S., Finkelstein, A., Brinkkemper, S. (2009) "A Sense of Community: A Research Agenda for Software Ecosystems". In: Proceedings of the 31st International Conference on Software Engineering (ICSE), New and Emerging Research Track, Vancouver, Canada, pp. 187-190, May.

Kolb, D.A. (1984) "Experiential Learning: Experience as the Source of Learning and Development". Prentice Hall.

Kolb, D.A., Boyatzis, R.E., Mainemelis, C. (2001) "Experiential Learning Theory: Previous Research and New Directions". Perspectives on Thinking, Learning, and Cognitive Styles 1(2001):227-247.

Lima, T.M.P., Barbosa, G.S., Santos, R.P., Werner, C.M.L. (2014) "Uma Abordagem Socio-técnica para Apoiar Ecossistemas de Software". iSys: Revista Brasileira de Sistemas de Informação 7(3):19-37.

Lopes, F., Delicato, F., Batista, T., Pires, P. (2009) "Uma Plataforma Baseada em Serviços Web para Integração de Middleware de Contexto". In: Proceedings of the XV Brazilian Symposium on Multimedia and the Web (WEBMEDIA), Fortaleza, Brasil, Article No. 13, October.

Manikas, K. (2016) "Revisiting Software Ecosystems Research: A Longitudinal Literature Study". The Journal of Systems and Software 117(2016):84-103.

Manikas, K., Hansen, K.M. (2013) "Software Ecosystems – A Systematic Literature Review". The Journal of Systems and Software 86(5):1294-1306.

McGregor, J.D. (2010) "A Method for Analyzing Software Product Line Ecosystems", In: Proceedings of the 4th European Conference on Software Architecture (ECSA) – 2nd International Workshop on Software Ecosystems (IWSECO), Copenhagen, Denmark, pp. 73-80, August.

Messerschmitt, D.G., Szyperski, C. (2003) "Software Ecosystem: Understanding an Indispensable Technology and Industry". The MIT Press, 1st Ed.

Mestad, A., Myrdal, R., Dingsoyr, T., Myrdal, R. (2007) "Building a Learning Organization: Three Phases of Communities of Practice in a Software Consulting Company". In: Proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS), Big Island, USA, p. 189a, January.

Miranda, R. (2004) "Um Ambiente de Apoio a Comunidades de Prática no Contexto da Estação TABA". Dissertação de Mestrado, COPPE/UFRJ, Rio de Janeiro, Brasil.

Nonaka, I., Takeuchi, H. (1995) "The Knowledge-Creating Company". Oxford University Press, 17th Ed.

Poell, R.F., Chivers, G.E., van der Krogt, F.J., Wildemeersch, D. (2000) "Learning-Network Theory: Organizing the Dynamic Relationships between Learning and Work". Management Learning 31(1):25-49.

Poulin, J.S. (1996) "Measuring Software Reuse: Principles, Practices, and Economic Models". Addison Wesley Professional.

Pressman, R.S. (2010) "Software Engineering – A Practitioner's Approach". McFraw-Hill, 7th Ed.

Russ, C. (2007) "Online Crowds – Extraordinary Mass Behavior on the Internet". In: Proceedings of the International Conference on New Media Technology, Graz, Austria, pp. 65-76, September.

Santana, F., Werner, C. (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 (IWSECO) – 4th International Conference on Software Business (ICSOB), Potsdam, Germany, pp. 9-16, June.

Santos, R.P. (2016) "Managing and Monitoring Software Ecosystem do Support Demand and Solution Analysis". PhD Thesis, COPPE/UFRJ, Rio de Janeiro, Brasil.

Santos, R.P., Cappelli, C., Maciel, C., Leite, J.C.S.P. (2016) "Transparência em Ecossistemas de Software". In: Anais do VII Congresso Brasileiro de Software: Teoria e Prática (CBSoft) – X Workshop em Desenvolvimento Distribuído de Software, Ecossistemas de Software e Sistemas-de-Sistemas (WDES), Maringá, Brasil, pp. , 75-79, Setembro.

Santos, R.P., Esteves, M.G.P., Freitas, G., Souza, J.M. (2014a) "Using Social Networks to Support Software Ecosystems Comprehension and Evolution". Social Networking 3(2):108-118.

Santos, R.P., Oliveira, J. (2013) "Análise e Aplicações de Redes Sociais em Ecossistemas de Software". In: Anais do IX Simpósio Brasileiro de Sistemas de Informação (SBSI), Minicursos, João Pessoa, Brasil, v. 2, pp. 19-24, Maio.

Santos, R.P., Valença, G.A., Viana, D., Estácio, B.J.S., Fontão, A.L., Marczak, S., Werner, C.M.L., Alves, C.F., Conte, T.U., Prikladnicki, R. (2014b) "Qualidade em Ecossistemas de Software: Desafios e Oportunidades de Pesquisa". In: Anais do V

Congresso Brasileiro de Software: Teoria e Prática (CBSoft) – VIII Workshop em Desenvolvimento Distribuído de Software, Ecossistemas de Software e Sistemas-de-Sistemas (WDES), Maceió, Brasil, v. 2, pp. 41-44, Setembro.

Santos, R.P., Werner, C.M.L. (2010) "Revisiting the Concept of Components in Software Engineering from a Software Ecosystem Perspective". In: Proceedings of the 4th European Conference on Software Architecture Workshops (ECSA) – 2nd International Workshop on Software Ecosystems (IWSECO), Copenhagen, Denmark, pp. 135-142, August.

Santos, R.P., Werner, C.M.L. (2011) "A Proposal for Software Ecosystems Engineering". In: Proceedings of the 3rd International Workshop on Software Ecosystems (IWSECO) – 2nd International Conference on Software Business (ICSOB), Brussels, Belgium, pp. 40-51, June.

Santos, R.P., Werner, C.M.L., Alves, C.F., Pinto, M.J., Cukierman, H.L., Oliveira, F.M., Egler, T.T.C. (2013) "Ecossistemas de Software: Um Novo Espaço para a Construção de Redes e Territórios envolvendo Governo, Sociedade e a Web". In: Werner, C.M.L., Oliveira, F.J.G., Ribeiro, P.T. (Org.), Políticas Públicas: Interações e Urbanidades, pp. 337-366, Letra Capital.

Santos, R.P., Werner, C.M.L., Barbosa, O.A.L.P., Alves, C.F. (2012) "Software Ecosystems: Trends and Impacts on Software Engineering". In: Proceeding of the XXVI Brazilian Symposium on Software Engineering (SBES), Natal, Brazil, pp. 206-210, September.

Schilling, M.A. (2008) "Strategic Management of Technological Innovation". McGraw Hill, 2nd Ed.

Schneider, K. (2009) "Experience and Knowledge Management in Software Engineering". Springer.

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 (ECSA) – 2nd International Workshop on Software Ecosystems (IWSECO), Copenhagen, Denmark, pp. 119-126, August.

Škerlavaj, M., Dimovski, V. (2006) "Social Network Approach to Organizational Learning". Journal of Applied Business Research 22(2):89-98.

Souza-Silva, J., Davel, E. (2007) "Da Ação à Colaboração Reflexiva em Comunidades de Prática". Revista de Administração de Empresas 47(2007):1-13.

van der Krogt, F.J. (1998) "Learning Network Theory: The Tension Between Learning Systems and Work Systems in Organizations". Human Resource Development Quarterly 9(2):157-177.

Voss, A., Schafer, A. (2003) "Discourse Knowledge Management in Communities of Practice". In: Proceedings of the 14th International Workshop on Database and Expert Systems Applications (DEXA), Prague, Czech Republic, pp. 782-786, September.

Wasserman, S., Faust, K. (1994) “Social Network Analysis: Methods and Applications”. Cambridge University Press.

Wenger, E., Mcdermott, R., Snyder, W.M. (2002) “Cultivating Communities of Practice”. Harvard Business Review Press.

Werner, C.M.L. (1992) “Reutilização de Software no Desenvolvimento de Software Científico”. Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro, Brasil.

Yu, E., Deng, S. (2011) “Understanding Software Ecosystems: A Strategic Modeling Approach”. In: Proceedings of the 3rd International Workshop on Software Ecosystems (IWSECO) – 2rd International Conference on Software Business (ICSOB), Brussels, Belgium, pp. 65-76, June.

Biografia Resumida dos Autores

Rodrigo Santos é doutor e mestre em Engenharia de Sistemas e Computação pela COPPE – Universidade Federal do Rio de Janeiro (UFRJ) no tema Ecossistemas de Software, com período sanduíche na University College London (UCL). É professor do Departamento de Informática Aplicada da Universidade Federal do Estado do Rio de Janeiro (UNIRIO) e pesquisador na COPPE/UFRJ nos temas Ecossistemas de Software, Engenharia de Requisitos e Aquisição. Atua como professor em disciplinas de graduação e de pós-graduação em Computação desde 2007 e em pesquisas e desenvolvimento em Processo de Software e Gestão e Desenvolvimento de Ecossistemas no governo pela Fundação Coppetec. É avaliador de cursos superiores de Computação pelo Ministério da Educação (MEC). Organizou o CBSoft/WDES 2015 e está organizando o SBQS/WASHES 2016. Ministrou palestras, minicursos e tutoriais em eventos como CLEI (2016), IHC (2016), SBQS (2016, 2015, 2009), CBSOFT (2016, 2013, 2012), SBSI (2013, 2011, 2010), CIbSE (2012), SBIE (2010), ICTAC (2010) e ERIs (Campos/RJ - 2015, 2010, 2008; Itacoatiara/AM - 2013; Porto Velho/RO - 2012; Itajubá/MG - 2012; Manaus/AM - 2010, 2010; Teresina/PI - 2008; Lavras/MG - 2008, 2007). CV Lattes: http://lattes.cnpq.br/8613736894676086.

Davi Viana é doutor e mestre em Informática pelo Programa de Pós-Graduação em Informática da Universidade Federal do Amazonas (UFAM). Graduado em Ciência da Computação pela UFAM. Possui curso Técnico em Informática pela Fundação Nokia de Ensino. Atualmente é professor do Curso de Engenharia da Computação da Universidade Federal do Maranhão (UFMA). É ainda membro colaborador no Programa de Pós-Graduação em Ciência da Computação (PPGCC) da UFMA e colaborador no Laboratório de Sistemas Distribuídos Inteligentes (LSDi) da UFMA. Ministrou cursos sobre Melhoria de Processos de Software. Além disso, foi membro do comitê de organização do IHC 2013 e co-organizador geral do SBQS 2015. Está organizando o SBQS/WASHES 2016. Tem interesse de pesquisa nas áreas de qualidade de software, melhoria processo de software (MPS), implementação de programas de MPS com ênfase na adoção de modelos de maturidade, e engenharia de software experimental (ESE). CV Lattes: http://lattes.cnpq.br/9297257833779277.