



Pós-Graduação em Ciência da Computação

JOYCE ALINE PEREIRA DE OLIVEIRA

**UM MODELO CONCEITUAL PARA GOVERNANÇA DE ECOSISTEMAS
DE SOFTWARE**



Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

Recife

2019

JOYCE ALINE PEREIRA DE OLIVEIRA

**UM MODELO CONCEITUAL PARA GOVERNANÇA DE
ECOSSISTEMAS DE SOFTWARE**

Tese apresentada ao Programa de Pós-Graduação em Ciências da Computação da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de doutor em Ciências da Computação.

Área de concentração: Engenharia de Software

Orientadora: Prof^a. Dra. Carina Frota Alves

Recife

2019

Catálogo na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

O48m Oliveira, Joyce Aline Pereira de
Um modelo conceitual para governança de ecossistemas de software /
Joyce Aline Pereira de Oliveira. – 2019.
197 f.: il., fig., tab.

Orientadora: Carina Frota Alves.
Tese (Doutorado) – Universidade Federal de Pernambuco. CIn, Ciência da
Computação, Recife, 2019.
Inclui referências, apêndices e anexo.

1. Engenharia de software. 2. Governança. I. Alves, Carina Frota
(orientadora). II. Título.

005.1 CDD (23. ed.) UFPE- MEI 2019-057

Joyce Aline Pereira de Oliveira

Um Modelo Conceitual para Governança de Ecossistemas de Software

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação.

Aprovado em: 21/02/2019.

Orientadora: Profa. Dra. Carina Frota Alves

BANCA EXAMINADORA

Prof. Dr. Hermano Perrelli de Moura
Centro de Informática /UFPE

Prof. Dr. Kiev Santos da Gama
Centro de Informática /UFPE

Profa. Dra. Bernadette Farias Lóscio
Centro de Informática / UFPE

Prof. Dr. Cleidson Ronald Botelho de Souza
Instituto de Ciências Exatas e Naturais / UFPA

Prof. Dr. Fernando Marques Figueira Filho
Departamento de Informática e Matemática Aplicada / UFRN

À minha amiga e mãe Alzenira Oliveira.

AGRADECIMENTOS

À Deus que em sua infinita misericórdia, graça e amor tem me sustentado em seus poderosos braços me mostrando em todo tempo que Ele governa todas as coisas.

À minha família, em especial ao meu pai por todo o orgulho e admiração que percebo sempre que olha para mim. À minha mãe por todos os conselhos, conversas e risadas. Deus foi muito generoso ao me dar o privilégio de ser sua filha. Ao meu irmão João Eudes, por todas as vezes nas quais me disse: “Jow, tu és fera!”, mesmo que eu não me veja assim. Ao meu esposo, Wagner por compreender meus estresses, meus anseios e medos. Obrigada por todas as vezes que demonstrou seu amor por mim com gestos tão singelos. À minha princesa Sofia. Enquanto eu concluía esta tese sentia seus primeiros movimentos em minha barriga e agradecia a Deus pela dádiva de me tornar mãe.

Ao pastor Arquimedes que tão sabiamente me orientou a escolher a UFPE para continuar meus estudos. Esta decisão mudou a minha vida e foi o primeiro passo de uma longa jornada que culminou em muitas realizações profissionais e pessoais. Muito obrigada!

À minha orientadora Carina Alves. Obrigada pelas orientações, pela confiança e também pelos ‘puxões de orelha’. Agora compreendo o quanto foram essenciais para meu crescimento e formação enquanto pesquisadora. Muito do que eu aprendi me serão úteis por toda a minha vida profissional e pessoal. Sou extremamente grata por isso.

Ao grupo de pesquisa da UFPE. Muito obrigada George Valença, André Santana e Higor Santos pelas dicas e ensinamentos valiosos. Foi extremamente construtivo trabalhar com vocês.

À direção, coordenação e colegas da UFMT que me apoiaram durante o doutorado considerando minha necessidade de realizar a pesquisa e trabalhar ao mesmo tempo. Agradeço imensamente por todas as vezes que me auxiliaram neste processo seja me atribuindo disciplinas que eu já domino ou me poupando de cargos administrativos que demandariam muito do meu tempo.

Às minhas amigas Carlina, Sheila e Gleice, que já considero família, muito obrigada pelo apoio, por compreenderem minha ausência e por todas as xícaras de café acompanhadas por longas conversas que com certeza me ajudaram a relaxar nos momentos certos.

Aos amigos e colegas que conheci em Mato Grosso e aos que mantive no Nordeste. O companheirismo de vocês tornou esta jornada mais suportável.

À todos que de algum modo contribuíram para a construção desta tese e para a conclusão do meu doutorado, muito obrigada!

RESUMO

Ecossistema de software é definido como um conjunto de atores trabalhando como uma unidade e interagindo em um mercado compartilhado de software e serviços com relacionamentos entre eles. Em um ecossistema de software, organizações trabalham colaborativamente para se manterem lucrativas e sobreviverem às mudanças do mercado. Para que o relacionamento entre estas organizações tenha sucesso é necessário que parceiros participem do ecossistema de software sem violar regras de colaboração ou se apropriar de vantagens que desestabilizem a saúde geral do ecossistema. A aplicação de mecanismos de governança é essencial para a obtenção deste equilíbrio. Entretanto, para implantar a governança é necessário compreender quais elementos precisam ser governados e como estes elementos se relacionam entre si. Este conhecimento é estratégico por possibilitar a descoberta e o tratamento de elementos críticos que prejudicam a saúde do ecossistema de software se forem negligenciados. Diante deste desafio, esta tese visa a construção de um modelo conceitual para apoiar a governança de ecossistemas de software. Um modelo conceitual é a representação de um conjunto de entidades e relacionamentos entre essas entidades, que fazem parte de um determinado domínio de conhecimento e auxiliam no entendimento do domínio em questão. O modelo conceitual proposto foi construído por meio de Design Science Research e representa uma visão holística dos elementos a serem governados em ecossistemas de software e suas relações. Estratégias de pesquisa como revisão sistemática da literatura, avaliação com especialistas e estudos de caso foram adotadas para apoiar a condução do ciclo de Design Science. A revisão sistemática da literatura apoiou a definição de mecanismos, de elementos de governança e de proposições que embasaram a construção do modelo conceitual. A avaliação com especialistas validou o modelo conceitual proposto e auxiliou no seu refinamento. Os estudos de caso realizados possibilitaram a aplicação do modelo conceitual para o contexto dos ecossistemas Eclipse, GNOME e SAP. Estes ecossistemas de software são amplamente estudados por pesquisadores da área. As diferentes características existentes entre estes ecossistemas permitiram a análise do grau de aplicação do modelo conceitual. A pesquisa resultou em um modelo conceitual com 28 elementos que apoia os atores encarregados da governança de diferentes tipos de ecossistemas de software, tais como, aberto, proprietário ou híbrido. Os resultados da revisão sistemática da literatura, o modelo conceitual, o glossário de elementos de governança, assim como o próprio processo de construção do modelo consistem em relevantes contribuições para pesquisa e prática por fornecerem uma visão holística da

governança de ecossistemas de software e por aumentarem o corpo de conhecimento da área. Em particular, os principais benefícios do modelo proposto são: (i) compreender e representar o contexto da governança de ecossistemas de software e (ii) atuar como uma referência para que os profissionais construam seus próprios modelos de governança de acordo com seu contexto e necessidades específicos.

Palavras chave: Ecossistema de software. Governança. Design Science Research.

ABSTRACT

Software ecosystem is defined as a set of actors working as a unit and interacting in a shared market of software and services with relationships between them. In a software ecosystem, organizations work collaboratively to stay profitable and survive market changes. In order for the relationship between these organizations to succeed, it is necessary for partners to participate in the software ecosystem without violating rules of collaboration or to take advantages that destabilize the health of the ecosystem. The application of governance mechanisms is essential to achieve this balance. However, to implement governance it is necessary to understand what elements need to be governed and how these elements relate to each other. This knowledge is strategic because it enables the discovery and treatment of critical elements that undermine the health of the software ecosystem if neglected. Faced with this challenge, this thesis aims at the construction of a conceptual model to support the governance of software ecosystems. A conceptual model is the representation of a set of entities and relationships between these entities, which are part of a given domain of knowledge and help in understanding the domain in question. The proposed conceptual model was constructed through Design Science Research and represents a holistic view of the elements to be governed in software ecosystems and their relationships. Research strategies such as systematic literature review, expert survey and case studies were adopted to support the Design Science cycle. Systematic literature review supported the definition of mechanisms, elements of governance and propositions that supported the construction of the conceptual model. The survey with experts validated the proposed conceptual model and assisted in its refinement. The case studies allowed the instantiation of the conceptual model for the context of Eclipse, GNOME and SAP ecosystems. These software ecosystems are widely studied by researchers in the field. The different characteristics of these ecosystems allowed us to analyze the degree of instantiation of the conceptual model. The research resulted in a conceptual model with 28 elements that supports actors responsible by governance of different types of software ecosystems such as open, proprietary or hybrid. The results of the systematic literature review, the conceptual model, the governance glossary, as well as the modeling process itself consist of relevant contributions to research and practice because they provide a holistic view of software ecosystems governance and increase the body of knowledge of the area. In particular, the main benefits of the proposed model are: (i) to understand and represent the context of software

ecosystems governance and (ii) act as a reference for professionals to construct their own governance models according to their context and needs specific.

Keywords: Software ecosystem. Governance. Design Science Research.

LISTA DE FIGURAS

Figura 1 – Visão geral de um ecossistema de software genérico (Hansen e Dyba, 2012), (Musil et al, 2013).	25
Figura 2 – O CobiT (ISACA, 2007)	34
Figura 3 – Ciclo de vida de serviços do ITIL (Nabiollahi e Sahibuddin, 2008)	35
Figura 4 – Processos do ITIL (Sortica et al, 2004).....	36
Figura 5 – Fases do ciclo de Design Science (Wieringa, 2014).....	42
Figura 6 – Métodos de pesquisa associados às fases de Design Science Research ...	44
Figura 7 – Aplicação do framework de Design Science de Wieringa (2014) para este estudo	45
Figura 8 – Passos da construção do modelo conceitual para governança de ecossistemas de software.	47
Figura 9 – Número de estudos por ano.....	61
Figura 10 – Canais de publicação mais populares.....	62
Figura 11 – Ecossistemas de software investigados pelos estudos incluídos.....	63
Figura 12 – Abordagem de pesquisa adotada pelos estudos incluídos.....	64
Figura 13 – Símbolos notacionais de UML usados no modelo conceitual	91
Figura 14 – Estrutura da construção do modelo conceitual proposto	92
Figura 15 – Modelo conceitual para governança de ecossistemas de software.....	95
Figura 16 – Características dos especialistas.....	98
Figura 17 – Validação das proposições por especialistas.....	101
Figura 18 – Aplicação do modelo conceitual para a perspectiva dos praticantes	113
Figura 19 – Página principal do ecossistema de software do Eclipse	114
Figura 20 – Aplicação do modelo conceitual para o Eclipse	121
Figura 21 – Página principal do ecossistema do GNOME.....	122
Figura 22 – Aplicação do modelo conceitual para o GNOME.....	128
Figura 23 – Página principal do ecossistema SAP	129
Figura 24 – Aplicação do modelo conceitual para o SAP.....	135
Figura 25 – Estratégias para suportar o elemento coopetição	144

LISTA DE TABELAS

Tabela 1 – Associação entre governança e saúde de ecossistemas de software (Baars e Jansen, 2012), Jansen e Cusumano (2012), Lantz e Weijden (2013). Adaptado.	28
Tabela 2 – Critérios da avaliação da qualidade dos artigos.....	58
Tabela 3 – Mecanismos de governança em ecossistemas de software.....	69
Tabela 4 – Definições para saúde de ecossistemas de software encontradas nos estudos	71
Tabela 5 – Métricas associadas à saúde de ecossistemas de software.....	74
Tabela 6 – Tipos de ecossistemas para os quais métricas de saúde foram propostas. 75	
Tabela 7 – Elementos da governança de ecossistemas de software	82
Tabela 8 – Lista de proposições do modelo conceitual	85
Tabela 9 – Definições de governança fornecidas por especialistas.....	98
Tabela 10 – Benefícios de governança do ecossistema de software segundo especialistas	100
Tabela 11 – Testabilidade do modelo conceitual	103
Tabela 12 – Suporte Empírico do Modelo Conceitual	104
Tabela 13 – Poder explicativo do modelo conceitual.....	104
Tabela 14 – Parcimônia do modelo conceitual.....	105
Tabela 15 – Generalidade do modelo conceitual.....	105
Tabela 16 – Utilidade do modelo conceitual	105
Tabela 17 – Características dos Praticantes.....	107
Tabela 18 – Elementos conceituais de cada estudo de caso	136
Tabela 19 – Resultado da aplicação do modelo conceitual para os ecossistemas investigados.....	138

SUMÁRIO

1	INTRODUÇÃO	16
1.1	Contexto da Pesquisa	16
1.2	Questões de Pesquisa	18
1.3	Contribuições da Tese	19
1.4	Síntese do capítulo e estrutura da tese.....	19
2	REFERENCIAL TEÓRICO	21
2.1	Ecosistemas de Software.....	21
2.2	Saúde de ecossistemas de software.....	26
2.3	Fundamentos da Governança.....	29
2.4	Síntese do capítulo	38
3	MÉTODO DE PESQUISA	40
3.1	Design Science Research	40
3.2	Ciclo de Design Science	42
3.2.1	<i>Investigação do problema.....</i>	<i>45</i>
3.2.2	<i>Design da Solução</i>	<i>46</i>
3.2.3	<i>Validação da Solução</i>	<i>48</i>
3.2.4	<i>Implementação da Solução</i>	<i>50</i>
3.2.5	<i>Avaliação da Implementação</i>	<i>53</i>
3.2	Síntese do capítulo	53
4	REVISÃO SISTEMÁTICA DA LITERATURA	55
4.1	Fases da Revisão Sistemática da Literatura.....	56
4.1.1	<i>Definição das questões de pesquisa</i>	<i>56</i>
4.1.2	<i>Processo de busca</i>	<i>57</i>
4.1.3	<i>Avaliação da qualidade dos artigos.....</i>	<i>58</i>
4.1.4	<i>Extração e análise dos dados</i>	<i>59</i>
4.1.5	<i>Crerios de inclusão e exclusão.....</i>	<i>59</i>
4.2	Resultados da Revisão Sistemática da Literatura	60

4.2.1	<i>Visão geral dos estudos</i>	60
4.2.2	<i>QP1.1 - Como a governança de ecossistemas de software é definida?</i>	64
4.2.3	<i>QP1.2 - Quais são os mecanismos propostos pela literatura para governar ecossistemas de software?</i>	67
4.2.4	<i>QP1.3 - Como a saúde é caracterizada pela literatura?</i>	70
4.2.4	<i>QP1.4 - Quais métricas para medir a saúde de ecossistemas de software foram propostas pela literatura?</i>	72
4.3	Ameaças à validade	76
4.4	Comparação com outras revisões da literatura	77
4.5	Síntese do capítulo	80
5	MODELO CONCEITUAL PARA GOVERNANÇA DE ECOSSISTEMAS DE SOFTWARE	81
5.1	Elementos envolvidos na governança de ecossistemas de software	82
5.2	Proposições do Modelo Conceitual	85
5.3	Representação do Modelo Conceitual	90
5.4	Síntese do capítulo	96
6	VALIDAÇÃO DO MODELO CONCEITUAL ATRAVÉS DE SURVEY COM ESPECIALISTAS	97
6.1	Caracterização demográfica	97
6.2	Definições e benefícios da governança de ecossistemas de software	98
6.3	Avaliação das Proposições	100
6.4	Avaliação do Modelo Conceitual de acordo com os Critérios de	103
6.5	Síntese do capítulo	106
7	IMPLEMENTAÇÃO DO MODELO CONCEITUAL	107
7.1	Implementação sob a visão de praticantes	107
7.1.1	<i>Visão geral</i>	107
7.1.2	<i>Resultados das Entrevistas</i>	108
7.1.3	<i>Avaliação da implementação por praticantes</i>	111
7.2	Estudo de caso I- ecossistema de software do Eclipse	114
7.2.1	<i>Visão Geral</i>	114
7.2.2	<i>Aplicação das Proposições do Modelo Conceitual para o Eclipse</i>	115

7.2.3	<i>Modelo Conceitual instanciado para o Eclipse</i>	120
7.3	Estudo de caso II – ecossistema de software GNOME	122
7.3.1	<i>Visão geral</i>	122
7.3.2	<i>Aplicação das proposições do modelo conceitual para o GNOME</i>	123
7.3.3	<i>Modelo Conceitual instanciado para o GNOME</i>	127
7.4	Estudo de caso III – ecossistema de software SAP	129
7.4.1	<i>Visão geral</i>	129
7.4.2	<i>Aplicação das proposições do modelo conceitual para o SAP</i>	130
7.4.3	<i>Modelo Conceitual instanciado para o SAP</i>	134
7.5	Avaliação da implementação por meio dos estudos de caso	136
7.6	Síntese do capítulo	140
8	DISCUSSÃO	142
8.1	Implicações teóricas	142
8.2	Implicações práticas	143
8.3	Trabalhos relacionados	144
8.4	Síntese do capítulo	146
9	CONCLUSÃO	147
9.1	Limitações e ameaças à validade	147
9.2	Trabalhos futuros	150
9.3	Síntese do capítulo	153
	REFERÊNCIAS	154
	APÊNDICE A – SITE COM INSTRUÇÕES PARA PARTICIPANTES DO SURVEY E DAS ENTREVISTAS	166
	APÊNDICE B – QUESTIONÁRIO COM ESPECIALISTAS	170
	APÊNDICE C – ROTEIRO DE ENTREVISTAS COM PRATICANTES	186
	ANEXO A – ESTUDO INCLUÍDOS NA REVISÃO SISTEMÁTICA DA LITERATURA	188

1 INTRODUÇÃO

Este capítulo apresenta o contexto da pesquisa, as motivações que a norteiam, as questões a serem investigadas e os objetivos a serem alcançados.

1.1 Contexto da Pesquisa

Um ecossistema de software (SECO– Software Ecosystem) é definido como um conjunto de atores trabalhando como uma unidade e interagindo em um mercado compartilhado de software e serviços com relacionamentos entre eles (Jansen et al., 2009). Na última década, uma grande quantidade de estudos tem se dedicado a investigar ecossistemas de software sob a perspectiva gerencial, social e tecnológica (Barbosa et al., 2013), (Hansen et al., 2014), (Bosch, 2012). Este interesse pela área é justificado porque ecossistemas de software influenciam a dinâmica de relacionamentos entre fornecedores de soluções de software e seus respectivos clientes e usuários. Dessa forma, ecossistemas de software fornecem um ambiente de negócios estratégico para que diversos atores cresçam e prosperem.

Um ecossistema de software frequentemente se baseia em uma plataforma compartilhada na qual os membros podem construir soluções específicas para criar valor adicional (Jansen et al, 2012). Segundo Boch (2009), plataforma neste contexto consiste em uma arquitetura de linhas de produtos de software com componentes distribuídos em torno da qual soluções são desenvolvidas e utilizadas. Desenvolvedores de software podem ampliar e enriquecer a plataforma enquanto compartilham custos e riscos. Exemplos de plataformas de software bem-sucedidas são o sistema operacional iOS, os aplicativos Google Apps e o navegador Firefox. Estes ecossistemas são gerenciados respectivamente pela Apple, Google e Mozilla. Estas empresas, chamadas de *keystone* ou orquestradoras, promovem o desenvolvimento sustentável do ecossistema, definindo estratégias e orquestrando as atividades dos participantes. O *keystone* é responsável pela gestão, controle e monitoramento de todos os atores que interagem em um ecossistema (Manikas et al., 2013). *Keystones* que são capazes de equilibrar seus próprios interesses, trazendo benefícios conjuntos para outros participantes tendem a criar ecossistemas de software saudáveis (Van Lingen et al, 2013).

Empresas em um ecossistema são mutuamente dependentes umas das outras para sobreviverem (Iansiti e Levien, 2004). Segundo van Angeren (2014), estes participantes frequentemente engajam em parcerias com o objetivo de compartilhar produtos, serviços ou bens e receber benefícios em troca. Os benefícios incluem visibilidade no mercado, aumento de receita, aquisição de recursos com menor custo, entre outros. Assim, organizações que deveriam ser concorrentes passam a trabalhar de forma colaborativa para se manterem lucrativas e sobreviverem no mercado. Entretanto, a colaboração implica em compartilhamento de informação. Isto inevitavelmente envolve competição, conflito de interesse, aumento de dependência entre parceiros e exposição das organizações (Alves e Pessoa, 2010), (Ceccagnoli et al, 2012).

Parceiros podem acessar dados estratégicos e usá-los para beneficiar-se, prejudicando outras empresas do ecossistema. Para que isto não aconteça, Bachmann (2001) afirma que a relação de dependência mútua entre os atores do ecossistema deve ser baseada em confiança: parceiros respeitam as regras definidas pelo *keystone* da plataforma que, por sua vez, não se aproveita da vulnerabilidade dos membros do ecossistema e nem exerce poder excessivo sobre eles. Para manter este equilíbrio, é necessário criar mecanismos capazes de governar o ecossistema de software de forma que potenciais parceiros se unam a plataforma sem violar regras de colaboração e competição ou se apropriar de vantagens que desestabilizem a saúde do ecossistema (Hurni e Huber, 2014). Saúde neste contexto é caracterizada como a capacidade de um ecossistema de software sobreviver a rupturas do mercado e a se manter resiliente diante de mudanças.

Governança envolve um conjunto de procedimentos estratégicos e processos para controlar, manter ou mudar o ecossistema de software (van Angeren, et al 2016). A governança agrega princípios que orientam a distribuição de direitos e responsabilidades entre as partes interessadas exigindo um cuidadoso equilíbrio entre controle e autonomia dados aos atores. Mecanismos de governança são empregados para definir o nível de controle, direitos de decisões e escopo da propriedade proprietária versus compartilhada em um ecossistema (Tiwana et al, 2010).

Para aplicar mecanismos de governança em um ecossistema de software é necessário conhecer quais elementos precisam ser governados. As relações entre estes elementos e a sua relevância para a sustentabilidade do ecossistema de software indicará

qual o modelo mais apropriado para governar o ecossistema. Existem vários modelos para governar ecossistemas de software. Por exemplo, GNU Linux é um ecossistema *open source* com uma comunidade bem-sucedida de desenvolvedores. iOS da Apple é um exemplo próspero de ecossistema proprietário com mecanismos de controle rígidos. Google construiu um ecossistema ativo em torno de sua comunidade de código aberto Android chamado de "Open handset Alliance ". Por outro lado, Symbian da Nokia é um sistema operacional de código aberto que não conseguiu criar um ecossistema próspero devido à sua incapacidade de atrair parceiros e desenvolver um rico conjunto de aplicativos (West & Wood, 2008). Os exemplos do iOS e Symbian mostram que a escolha de estratégias e de mecanismos de governança são decisões que definem a vida ou a morte das organizações que atuam como keystones.

Com um modelo de governança bem definido, organizações podem ganhar vantagem estratégica sobre outras na medida em que avaliam e melhoram sistematicamente o governo do seu ecossistema de software, levando-o a ter um melhor desempenho e, conseqüentemente, a ser mais saudável (Baars and Jansen, 2012). Um ecossistema de software é considerado saudável quando possui a habilidade de permanecer produtivo e de evoluir ao longo do tempo. Neste contexto, a produtividade está associada a capacidade que o ecossistema possui de converter insumos em resultados.

1.2 Questões de Pesquisa

A seguinte questão de pesquisa orientou este estudo: Como é a dinâmica de governança de ecossistemas de software?

A questão principal foi desdobrada nas seguintes subquestões:

- *QP1.1. Como a governança de ecossistemas de software é definida?*
- *QP1.2. Quais são os mecanismos propostos pela literatura para governar ecossistemas de software?*
- *QP1.3. Como a saúde de ecossistemas é caracterizada pela literatura ?*
- *QP1.4. Quais métricas para medir a saúde de ecossistemas de software foram propostas pela literatura?*
- *QP1.5. Quais são os elementos envolvidos na governança de ecossistemas de software e como estes elementos estão relacionados?*

1.3 Contribuições da Tese

Esta tese visa examinar a governança de ecossistemas de software através da investigação dos seus elementos e de como estes elementos se relacionam entre si. Em particular, as seguintes contribuições são propostas nesta tese de doutorado:

- Investigação de como a governança de ecossistemas é definida, quais são os mecanismos de governança propostos, como saúde de ecossistemas é caracterizada e quais são as métricas de saúde descritas pela literatura,
- Construção de um modelo conceitual de governança de ecossistemas de software baseado nos resultados da revisão sistemática da literatura,
- Avaliação do modelo proposto com especialistas que possuem experiência acadêmica e prática na área de ecossistemas de software,
- Aplicação do modelo proposto em diferentes ecossistemas de software sob a visão de praticantes,
- Aplicação do modelo proposto nos ecossistemas de software Eclipse, SAP e GNOME.

1.4 Síntese do capítulo e estrutura da tese

Este capítulo apresentou o contexto da pesquisa, a questão de pesquisa macro e as subquestões que guiaram a condução do estudo, o objetivo geral e as principais contribuições. O restante da tese está estruturado da seguinte forma:

O Capítulo 2 apresenta a fundamentação teórica da pesquisa. Serão apresentadas as definições inerentes a ecossistemas de software, saúde de ecossistemas de software e governança.

O Capítulo 3 apresenta o método de pesquisa Design Science Research, as etapas de Design Science, além dos procedimentos e instrumentos de coleta e análise de dados.

O Capítulo 4 apresenta os resultados da etapa de investigação do problema de Design Science que foi realizada através de uma revisão sistemática da literatura.

O Capítulo 5 apresenta o modelo conceitual construído para apoiar a governança de ecossistemas de software.

O Capítulo 6 apresenta a validação do modelo conceitual por meio de survey com especialistas.

O Capítulo 7 são descritos os resultados das entrevistas com praticantes e a aplicação do modelo conceitual sob a perspectiva destes e são detalhados os resultados dos estudos de caso realizados nos contextos no Eclipse, no GNOME e no SAP.

O Capítulo 8 são discutidas as implicações teóricas e práticas da pesquisa.

O Capítulo 9 conclui o estudo apresentando as limitações, as ameaças a validade e os trabalhos futuros.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta brevemente as definições teóricas das grandes áreas investigadas neste estudo que são a governança e ecossistemas de software.

2.1 Ecossistemas de Software

Ecossistemas de software têm emergido nos últimos anos como um novo caminho para entender os relacionamentos entre soluções de software, clientes e empresas fornecedoras de produtos de software (Franco-Bedoya, 2014). Há várias definições de ecossistema de software na literatura. Kittlaus e Clough (2009) definem ecossistema de software como “uma rede informal de unidades legalmente independentes que se beneficiam de um produto de software e possuem uma influência positiva sobre o seu sucesso econômico”.

Em um contexto social Manikas e Hansen (2013) preconizam que um “ecossistema de software é a interação de um conjunto de atores sobre uma plataforma tecnológica comum que resulta num certo número de soluções ou serviços de software.” No contexto técnico, Boch (2009) define ecossistema de software como “um conjunto de soluções que permite, suporta e automatiza atividades e transações entre atores e organizações que proveem estas soluções em um ecossistema proprietário ou *open source*”.

Seguindo uma visão sistêmica, Messerschmitt e Szyperski (2005) afirmam que “um ecossistema de software se refere a um conjunto de produtos de software que têm algum dado grau de relacionamento simbiótico”. Em biologia, a palavra simbiose se refere a uma relação mutuamente vantajosa entre dois ou mais organismos vivos de espécies distintas. Esta definição remete a Dhungana et al (2010), os quais afirmam que os conceitos de ecossistema de software são inspirados em ecossistemas naturais. Conforme propõe Dhungana et al (2010), em ambos os contextos há influência de fatores externos, reserva finita de recursos, colaboração, competição e alteração de participantes conforme a dinamicidade do ecossistema.

Numa visão organizacional, Jansen e Cusumano (2012) indicam que ecossistemas de software são “subconjuntos de ecossistemas de negócio” nos quais

padrões (XML, BPM, CORBA), produtos (OpenOffice, SAP Business), hardware (PDAS, HTC Diamond) e plataformas (.NET, Facebook, Android) podem ser instanciados para uma rede comercial genérica. Hansen et al (2014) corroboram esta afirmação ao preconizar que “ecossistema de software pode ser considerado um tipo particular de ecossistema de negócio em que uma plataforma tecnológica intermedia as interações entre os atores.”

Os conceitos anteriormente citados remetem a características técnicas, de negócio e sociais de um ecossistema de software. Todas as definições são válidas, mas enfocam em diferentes aspectos de um ecossistema de software, o que é importante pois todas estas dimensões existem nestes ecossistemas. De fato, dos Santos e Werner (2011) afirmam que um ecossistema de software possui um caráter tridimensional. A dimensão técnica foca na arquitetura da plataforma do ecossistema, no gerenciamento de variabilidade de funcionalidades e na definição de um processo de engenharia de software integrado. Esta dimensão explora a tomada de decisão e a manutenção arquitetural. A dimensão de negócio envolve artefatos, recursos, informações, inovação, planejamento estratégico e todos os insumos necessários para que o ecossistema de software se mantenha firme no mercado. A dimensão social foca no balanceamento e proposição de interações, promoções e atividades de colaboração entre os participantes do ecossistema.

O termo ecossistema refere-se ao funcionamento do sistema como um todo. No entanto, cada indivíduo desempenha um papel importante na estabilidade global e sustentabilidade de um ecossistema (Dhungana et. Al, 2010). A composição de um ecossistema de software define como ele funciona, como ele efetivamente reage as mudanças, quais tipos de atores existem, com que frequência esses atores surgem, qual papel eles assumem e quão ampla é a sua atuação (Jansen et al, 2009). Manikas e Hansen (2013) definiram um conjunto dos atores mais comuns em um ecossistema de software:

- *Keystone*, orquestrador ou dono da plataforma: companhia, departamento de uma empresa, ator ou conjunto de atores, comunidade ou entidade independente que é responsável pelo bom funcionamento do ecossistema de software. Esta unidade é responsável por gerir a plataforma por meio do estabelecimento de regras, processos e procedimentos de negócio. Também se responsabiliza pela definição e monitoramento de padrões de

qualidade e pela orquestração do relacionamento entre os atores do ecossistema.

- *Niche player* ou time desenvolvedor: é o ator que contribui tipicamente desenvolvendo ou adicionando componentes à plataforma e produzindo funcionalidades que os clientes precisam. Este ator complementa o trabalho do *keystone*, fornecendo valor para o ecossistema. Dependendo do modelo de gerenciamento do ecossistema os participantes do nicho podem influenciar a tomada de decisão na gestão do ecossistema de software.
- Ator externo, desenvolvedor (equipe) externo ou comunidade: é o ator (empresa, pessoa, entidade) que proporciona valor indireto ao ecossistema. Este ator é externo ao gerenciamento do ecossistema de software e, geralmente, possui atuação limitada. Dependendo da natureza do ecossistema, o ator externo pode desenvolver funcionalidades em paralelo com a plataforma, identificar erros, promover o ecossistema de software e seus produtos ou propor melhorias. Este tipo de ator compreende o papel do participante ou seguidor em ecossistemas *open source*.
- Vendedor, fornecedor independente de software, revendedor ou revendedor com valor agregado: é principalmente a empresa ou unidade de negócio que faz lucro com a venda dos produtos do ecossistema de software para clientes, usuários finais ou outros fornecedores. Os produtos podem ser integrações completas, componentes, venda ou locação de licenças ou contratos de suporte. Um *keystone* é altamente dependente de seus fornecedores (Van Den Berk et al, 2010). Se um fornecedor muda sua estratégia ou vai à falência, isso terá grande efeito sobre todo o ecossistema de software (Van Den Berk et al, 2010).
- Cliente ou usuário: é a pessoa, empresa ou entidade que compra ou obtém um produto completo ou parcial do ecossistema de software através de um fornecedor, de um vendedor ou diretamente do ecossistema. Segundo Van Den Berk (et al, 2010), os clientes são a força mais importante de um

ecossistema de software. Se eles não estiverem satisfeitos eles poderão, e irão mudar para um ecossistema concorrente (Van Den Berk et al, 2010).

Adicionalmente aos papéis de *niche player* e de *keystone*, Jansen et al (2009) cita o papel do dominador, *broker* e *bridge*. O dominador é um ator que progressivamente assimila ou elimina outros *keystones*, outros dominadores ou outros *niche players* se tornando o único fornecedor de inovação dentro do ecossistema de software. Fortes dominadores em alguns casos são regulados por fatores externos que tornam possível a sobrevivência do ecossistema de software, entretanto podem levar o ecossistema a destruição. O *broker* é um ator que serve como uma interface entre dois outros atores. *Bridge* é um participante ativo em dois ecossistemas de softwares simultaneamente e que possui influência sobre ambos fornecendo recursos como dinheiro, pessoas ou conhecimento.

Segundo Hansen e Dyba (2009) e Musil et al (2013), um ecossistema de software genérico possui cinco aspectos que são (Figura 1): Organização de referência central, Característica de rede, plataforma, valores compartilhados e autoregulação. Cada aspecto é descrito em sequência conforme definições dadas por Hansen et al (Ano) e Juergen Musil et al (ano):

- a. *Organização central de referência:* Cada ecossistema de software é governado por uma organização que é estratégica na supervisão e orientação da evolução do ecossistema. Esta organização possui recursos necessários para atuar no crescimento do ecossistema e pode exercer alguma forma de controle sobre os atores (Ex: Appel, Facebook, Apache Foundation).
- b. *Caráter em rede:* Os ecossistemas de software tendem a gerar estruturas de rede entre seus atores e organizações. De fato, uma expectativa central ao se aderir a um ecossistema é prosperar a partir dos benefícios que surgem dessas interdependências.
- c. *Plataforma:* A plataforma é o driver central em torno do qual o ecossistema de software evolui e pode ser caracterizada como uma fundação, uma tecnologia ou conjunto de componentes utilizados por várias empresas distintas.

- d. *Valores compartilhados*: Os valores compartilhados podem ser extrínsecos (ex: produto de software, plataforma, serviços) ou intrínsecos (ex: motivações individuais).
- e. *Autorregulação*: A autorregulação é conseguida através de feedback e interações regulares entre os atores.

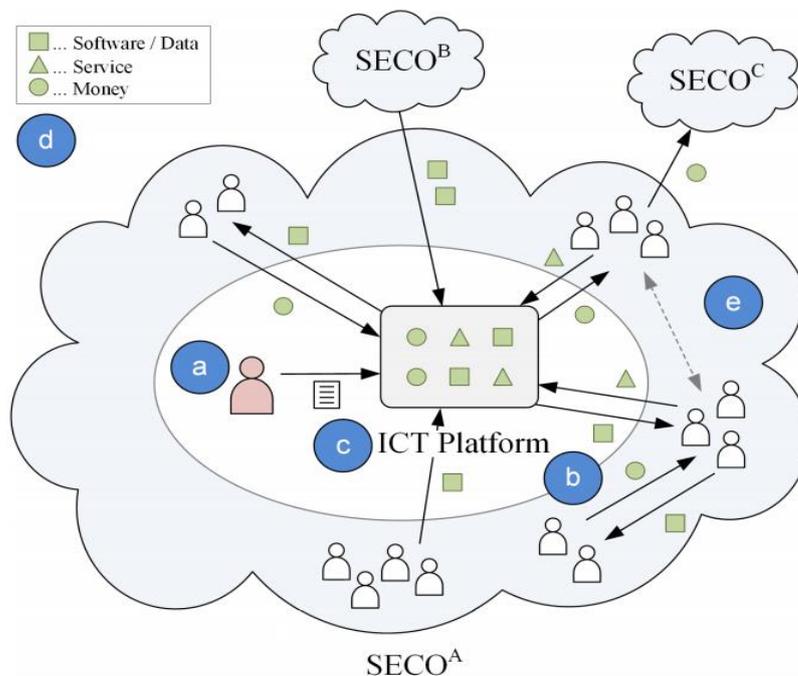


Figura 1: visão geral de um ecossistema de software genérico (Hansen e Dyba, 2012), (Musil et al, 2013).

Segundo Manikas e Hansen (2013), um ecossistema pode ser proprietário ou aberto. Em um ecossistema proprietário, organizações visam sobreviver e prosperar Jansen e Cusumano (2013). Nesse escopo os atores aspiram a obtenção de benefícios financeiro e há uma grande empresa que atua como a orquestradora do ecossistema (Ex: Microsoft, Apple). O código fonte e outros artefatos que geram valor são protegidos e novos atores passam por algum tipo de avaliação antes de sua adesão ao ecossistema ser permitida (Manikas e Hansen, 2013).

Em ecossistemas *open source*, barreiras de entrada costumam ser mais flexíveis e os participantes variam, podendo ser fundações (ex. Eclipse foundation), empresas comerciais (ex. RedHat) ou desenvolvedores independentes (ex. membros do Rail ecosystem) (Jansen e Cusumano, 2013). Neste contexto, todos trabalham

colaborativamente, os atores podem participar de retornos não monetários (ex. fama, conhecimento, ideologia, etc.) e o foco concentra-se na resolução de problemas técnicos e sociais (Manikas e Hansen, 2013).

Ainda há o conceito de ecossistema híbrido, aquele que mescla características de um ecossistema proprietário e de um aberto (Ex: Eclipse). Independentemente do tipo (aberto, proprietário ou *open source*) um ecossistema de software precisa ser governando para manter-se sustentável e para que possua propensão ao crescimento.

2.2 Saúde de ecossistemas de software

Em ecossistemas de software um problema a ser governado é o paradoxo da mudança (Wareham et al, 2014), (Tilson et al, 2010). Isto refere-se ao fato de que o ecossistema precisa ser estável e evoluível ao mesmo tempo. Sem evolução, o ecossistema de software não pode ajustar-se adequadamente às mudanças nas necessidades dos clientes, à dinâmica do mercado e à evolução da tecnologia (Wareham, et al 2014). Sem estabilidade, os atores têm pouca garantia de que seus investimentos em recursos financeiros e humanos podem render retornos a longo prazo (Wareham, et al 2014). Um dos principais objetivos da governança de ecossistema de software é balancear este conflito (Tiwana et al, 2010). Para alcançar este equilíbrio deve-se compreender como funciona a distribuição de poder entre os atores, quais são as formas e regras de entrada, como parcerias são estabelecidas, quão acessível o ecossistema deve ser para manter o equilíbrio do uso, desenvolvimento e comercialização dos produtos e serviços, entre outros aspectos (Schrieck, 2016). Todos estes fatores afetam a saúde do ecossistema de software.

Segundo Den Hartigh et al (2013) o termo saúde refere-se a capacidade de o ecossistema prosperar ao longo do tempo mantendo-se resiliente diante das dificuldades. Segundo Santos et al (2014), saúde é definida como a habilidade de um ecossistema permanecer estável e ao mesmo tempo expansível, atendendo todas as demandas diárias. Jansen e Cusumano (2013) afirmam que saúde consiste na capacidade de o ecossistema crescer e se manter sustentável. Neste sentido, quanto mais saudável o ecossistema de software for, maior será a sua propensão ao crescimento. Carvalho et al (2018) reforça que saúde consiste em definir o grau em que um ecossistema de software oferece oportunidades para os atores que dependem dele.

Vários trabalhos relacionam saúde com governança em ecossistemas de software (Baars e Jansen, 2012), Alami et al (2015), Wkuk et al (2014). Estes autores consideram que a governança possibilita o alcance de melhor performance, gerando assim grande impacto sobre a saúde de ecossistemas de software.

Em seu trabalho seminal, Iansiti e Levien (2004) definiram três métricas críticas de saúde para os ecossistemas empresariais e biológicos: produtividade, robustez e criação de nicho. Observa-se que seus estudos sobre a saúde do ecossistema influenciaram fortemente a maioria dos autores no domínio de ecossistemas de software que investigam este tópico. Dada a grande relevância do trabalho de Iansiti e colegas para o corpo de conhecimento no campo da saúde de ecossistema de software, adotou-se neste estudo sua classificação. Na sequência, cada elemento é definido:

- **Produtividade:** A habilidade em que um ecossistema converte entradas em saídas;
- **Robustez:** A habilidade de um ecossistema de sobreviver a interrupções, como mudanças tecnológicas e de mercado;
- **Criação de nicho:** A habilidade de um ecossistema de aumentar significativamente a diversidade, criando recursos e nichos valiosos.

Baars e Jansen (2012) definiram um framework contendo um conjunto de atividades de governança e as associou com cada métricas mencionada acima (Tabela 1) O estudo foi posteriormente complementado por Jansen e Cusumano (2012) que avaliaram o framework por meio de estudos de caso sob a perspectiva da governança e da saúde de ecossistemas de software. O mesmo framework foi utilizado por Lantz e Weijden (2013) também por meio de um estudo de caso para avaliar o grau em que a governança afeta a saúde em um ecossistema de software.

Tabela 1: Associação entre governança e saúde de ecossistemas de software (Baars e Jansen, 2012), Jansen e Cusumano (2012), Lantz e Weijden (2013). Adaptado.

Métrica	Ação de governança	Significado da ação
Produtividade	Organizar dia dos desenvolvedores	Promover a interação entre desenvolvedores do ecossistema de software por meio de eventos.
	Realizar marketing colaborativo	Unir esforços entre os parceiros de organizações distintas para aumentar a visibilidade do ecossistema de software.
	Criar programas de parcerias e novos canais de vendas	Inovar para aumentar a receita do ecossistema de software e a criação de valor.
Robustez	Criar modelo de parceria	Consiste em definir regras, processos e incentivos relacionados à adesão de parceiros.
	Realizar marketing	Divulgar os serviços, produtos e ações do ecossistema de software para aumentar a participação.
	Criar programa de desenvolvimento de parcerias	Definir estratégias para fortalecer parceiros “fracos” que possam afetar a saúde do ecossistema e atrair outros participantes.
	Formar alianças	Criar novos relacionamentos e fomentar esta prática para aumentar a agregação de participantes no ecossistema.
	Estabilizar APIs	Simplificar a configuração de software para facilitar a compatibilidade entre aplicações.
	Definir barreiras de entrada	Definir critérios para dificultar a adesão de parceiros que podem enfraquecer o ecossistema de software.
	Propagar conhecimento operacional	Promover a comunicação e o compartilhamento do conhecimento entre os atores.
Criação de nicho	Expandir a aplicabilidade	Ampliar a diversidade por meio da criação de valor.
	Tornar a estratégia explícita	Compartilhar estratégias e roadmaps e em criar confiança entre os atores.
	Criar APIs	Desenvolver APIs padronizadas que permitam a compatibilidade entre aplicações e a interação entre desenvolvedores externos.
	Realizar co-desenvolvimento	Desenvolver colaborativamente para aumentar inovação, reduzir custos e diminuir o tempo de implementação.
	Desenvolver novos modelos de negócio	Introduzir novas oportunidades de negócio para ampliar a criação de nicho.

A preocupação destes autores em relacionar a governança com a saúde mostra que a conexão entre estes temas de fato existe e não pode ser ignorada ao se investigar a governança no contexto de ecossistemas de software. A classificação da saúde em produtividade, a criação de nicho e a robustez só é passível de monitoramento mediante a definição de métricas e indicadores que operacionalizem estes três elementos. Estas

observações justificam a inserção das questões QP1.3 (*Como a saúde de ecossistemas é caracterizada pela literatura*) e QP1.4 (*Quais métricas para medir a saúde de ecossistemas de software foram propostas pela literatura?*)

2.3 Fundamentos da Governança

Governança é um conceito associado à necessidade de proteção dos investimentos e à sustentabilidade das organizações (Hoogervost, 2009). O tema governança surgiu em meados da década de trinta e recebeu grande atenção no início de 2000 mediante a necessidade percebida pelas empresas de controlar fraudes, de solucionar conflitos inerentes à propriedade dispersa e de alinhar a divergência entre os interesses dos sócios, executivos versus o melhor interesse da empresa (IBCG, 2013).

É necessário ressaltar que governança difere de gestão. A gestão é responsável pelo planejamento, desenvolvimento, execução e monitoramento das atividades em consonância com a direção definida pelo órgão de governança a fim de atingir os objetivos corporativos (ISACA, 2012). Já a governança garante que as necessidades, condições e opções das partes interessadas sejam avaliadas a fim de determinar objetivos corporativos acordados e equilibrados, definindo a direção através de priorizações e tomada de decisão, monitorando o desempenho e a conformidade com a direção e os objetivos estabelecidos (ISACA, 2012).

Segundo Bird (2001), enquanto executivos e gerentes administram, desenvolvem, implementam e monitoram as estratégias de negócios no dia-a-dia, os conselhos e outras estruturas de governança lidam com a política, cultura e direção geral da organização. Executivos gerem organizações em virtude da autoridade delegada a eles por aqueles que governam (Bird, 2001).

A governança atua no sentido de responsabilizar, fiscalizar e controlar a gestão de uma corporação (Hoogervorst, 2009). O conjunto de relações entre a gestão de uma empresa, seu conselho de administração, acionistas e outras partes interessadas caracteriza a governança corporativa (OECD, 2004).

Segundo o IBGC (2013), a governança corporativa surgiu para superar o "conflito de agência" clássico. Nesta situação, o proprietário (acionista) delega à um agente especializado (administrador) o poder de decisão sobre a empresa nos termos da lei. Nesta

situação podem surgir divergências quanto ao entendimento de cada um dos grupos daquilo que consideram ser o melhor para a empresa e do que as práticas de governança buscam realizar. Neste sentido, a governança corporativa visa criar um conjunto eficiente de mecanismos, tanto de incentivos quanto de monitoramento, a fim de assegurar que o comportamento dos administradores esteja sempre alinhado com os objetivos da empresa. Segundo Hoogervost (2009) a busca pela excelência da governança corporativa se justifica pelo fato de que empresas só podem sobreviver e crescer se continuarem a executar o seu propósito com sucesso, isto é, entregar produtos e serviços para o cliente de forma eficaz.

A governança corporativa baseia-se nos seguintes princípios (IBGC, 2013), (Hoogervost, 2009), (OECD, 2004) *apud* Santana (2015):

- **Alinhamento estratégico:** Cursos de ação e objetivos em todos os níveis organizacionais devem estar alinhados com a estratégia da organização (missão, visão, planejamento estratégico e seus desdobramentos).
- **Efetividade:** Todas as ações organizacionais devem buscar a efetividade, traduzida como eficácia e eficiência ao longo do tempo no alcance dos objetivos estabelecidos.
- **Equidade:** Todas as diferentes partes interessadas devem ser consideradas no estabelecimento de critérios e decisões relativas às ações organizacionais.
- **Legitimidade:** Princípio jurídico fundamental do Estado Democrático de Direito que amplia a incidência do controle para além da aplicação isolada do critério da legalidade. Não basta verificar se a lei foi cumprida, mas se o interesse público, o bem comum, foi alcançado.
- **Responsabilidade:** Deve haver sempre uma pessoa que possa ser cobrada e responsabilizada por decisões e resultados de desempenho das ações de governança.
- **Prestação de contas:** Dá às partes interessadas a possibilidade de cobrar ou dar reconhecimento pelos resultados à pessoa certa.
- **Transparência:** Deve haver clareza de entendimento e acesso para as partes interessadas sobre os critérios, decisões e resultados de ações de

governança (resguardada a proteção a dados estratégicos ou sigilosos de natureza legal, cuja veiculação possa prejudicar a organização).

Em suma, a governança corporativa engloba um conjunto de princípios através dos quais os investidores externos se protegem contra a expropriação por parte dos gestores internos (La porta et al, 2000). Segundo La porta (2000), expropriação pode ser entendida como i) desvio ou roubo de lucros, ii) favorecimento ilegal de vendas para compradores específicos, iii) alocação de cargos gerenciais para pessoas não qualificadas em troca de benefícios pessoais, iv) utilização de artifícios que coloquem os interesses pessoais acima dos interesses da empresa prejudicando a sua sustentabilidade. Estes pontos estão também relacionados com o “problema da agência” (Jensen and Meckling, 1976).

De acordo com La Porta et al (2000), quando gestores recebem a incumbência de governar corporações, tipicamente obtêm direitos e poderes protegidos por lei. Entretanto, em muitos casos a expropriação ocorre por excesso de poder. Gestores se beneficiam do poder dado pelos investidores externos para mudar o corpo de diretores, interromper projetos, gerenciar o capital corruptamente e manipular ativos em próprio proveito. Portanto a alocação de poder também é um problema a ser tratado a luz dos princípios de governança.

Em ecossistemas de software, os princípios da governança corporativa também estão presentes. Atores de um ecossistema de software possuem missões, visões e planejamentos estratégicos distintos, entretanto, para trabalharem colaborativamente é necessário que haja um *alinhamento estratégico* entre eles, do contrário haverá conflito e possível ruptura no relacionamento. Todos os atores também precisam trabalhar na busca da eficácia e da eficiência no alcance dos seus objetivos. Se o objetivo é criar valor, deve haver *efetividade* no alcance deste propósito.

Todos os atores devem ser considerados no processo de tomada de decisão e todos devem ter a oportunidade de crescer dentro do ecossistema de software. Se a *equidade* não for considerada, o ator que tem maior autonomia tenderá a se beneficiar mais da sua posição, anulando as chances de crescimento do ator que possui maior dependência.

Todos no ecossistema de software devem estar apoiados em bases legais para o alcance do bem comum. É necessário, portanto, que sejam criados processos de auditoria para verificar se as normas do ecossistema de software estão sendo cumpridas e se o bem comum (por exemplo, a criação de valor) está sendo obtido. Logo, a *legitimidade* também é um princípio a ser considerado.

Se um ator realiza atividades de colaboração, e estas atividades impactam a sustentabilidade do ecossistema de software, ele deve ser responsabilizado formalmente por suas ações. A formalização da *responsabilidade* possibilita que o ator seja recompensado ou punido de acordo com sua conduta, com base nas normas do ecossistema de software.

O ator precisa prestar contas das ações que estão sob sua responsabilidade. A prestação de contas informa o desempenho do ator e reporta a situação do ecossistema de software. Os relatórios gerados na *prestação de contas* são insumos relevantes para a distribuição de recompensas e para a análise da saúde do ecossistema de software.

A prestação de contas deve ser realizada com *transparência*, ou seja, com clareza, consistência e veracidade para que seja reportado o estado real da saúde do ecossistema de software e para que as pessoas certas sejam recompensadas. Se a prestação de contas é omissa, o ecossistema de software não será corretamente governando e a sua sustentabilidade será comprometida.

A expropriação e o “problema de agência” também existem em ecossistema de software. Um ator pode tentar se apropriar indevidamente do valor criado por outro ator e pode tentar manipular ativos obtidos pelo ecossistema em benefício próprio. Um keystone também utilizar seu poder excessivamente para controlar os outros atores da plataforma.

Embora princípios da governança corporativa sejam identificados na governança de ecossistemas de software, a equivalência entre estes princípios, até o estudado, não é formalizada na literatura. A maturidade da área de governança corporativa e o alinhamento dos princípios e dos problemas tratados neste domínio com a governança de ecossistema de software indica que a governança corporativa pode ser utilizada como base para a construção do modelo conceitual que foi elaborado nesta tese. Princípios da

governança corporativa têm sido utilizados como referência para estruturar frameworks que englobam aspectos distintos da gestão de uma organização, por exemplo: governança de Tecnologia da Informação (TI) (Weill e Ross, 2004), governança de Arquitetura Orientada a Serviços (Schepers et al, 2008) e governança de Gestão de Processos de Negócio (Korhonen, 2007). Todos esses tipos de governança são comumente referenciados como subconjuntos da governança corporativa (Kingsford, 2003).

Governança de Tecnologia da Informação é o alinhamento estratégico da TI com o negócio de tal forma que o máximo valor comercial é alcançado através do eficaz controle, desenvolvimento, manutenção, prestação de contas, gerenciamento de desempenho e gerenciamento de risco de TI. Nabiollahi e Sahibuddin (2008) afirmam que a governança de TI é o processo pelo qual decisões são tomadas em torno de investimentos de Tecnologia da Informação e de como os resultados destas decisões são monitorados e mensurados por todas as partes.

Segundo Earl (1994), governança de Tecnologia da Informação visa o alinhamento entre os objetivos de TI e os objetivos da empresa, o uso responsável de recursos de TI e a gestão adequada de riscos. De acordo com Weeb et al (2006), estruturas, frameworks de controle e processos são termos frequentemente utilizados na literatura de governança de TI. Estes termos representam conceitos e ferramentas importantes para a aplicação, implementação e desenvolvimento de governança de TI.

Dois dos principais frameworks de governança de Tecnologia da Informação são o CobiT (Control Objectives for Information and Related Technology) e o ITIL (Information Technology Infrastructure Library). O CobiT foi criado pelo ISACA (*Information Systems Audit and Control Association*) através do Instituto de Governança de TI (*IT Governance Institute*). O CobiT (Figura 2) mantém um conjunto atualizado de padrões internacionais de boas práticas referentes ao uso corporativo da TI para os gerentes e auditores de tecnologia (Sortica et al, 2004). O CobiT aborda a governança de processos, atividades de TI e é organizado em um ciclo de vida composto por quatro domínios: i) planejar e organizar, ii) adquirir e implementar, iii) entregar e apoiar e iv) monitorar e avaliar. Cada domínio possui processos específicos. Para cada processo, recursos afetados são identificados. Estes recursos são pessoas, informação, aplicações e infraestrutura. No CobiT, requisitos de negócio são avaliados conforme os seguintes

critérios: efetividade, eficiência, confidencialidade, integridade, disponibilidade, conformidade e confiabilidade.

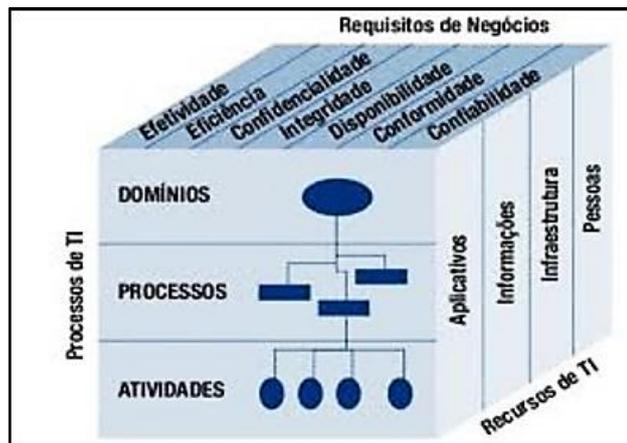


Figura 2: O CobiT (ISACA, 2007)

O ITIL, por sua vez, está na terceira versão e é composto basicamente por dois componentes: o Guia Complementar do ITIL e o ITIL Core. O Guia Complementar do ITIL consiste em um conjunto de publicações que fornece orientações de governança considerando distinções existentes referentes a setores industriais, tipos de organizações, modelos de operação e arquiteturas tecnológicas. O ITIL Core consiste em três volumes que contêm as melhores práticas aplicáveis para todos os tipos de organizações que proveem serviços de TI. A estrutura do ITIL Core é representada por um ciclo de vida iterativo e multidimensional (Figura 3) composto pelas seguintes fases: estratégia de serviço, desenho de serviço, transição de serviço, operações de serviço e melhoria contínua de serviço (Nabiollahi e Sahibuddin, 2008). Um serviço no contexto do ITIL consiste em uma atividade prestada pela organização relacionada à Tecnologia de Informação. Normalmente o serviço é disponibilizado em um portfólio e solicitado pelos departamentos organizacionais ao setor responsável pela governança e gestão da Tecnologia de Informação.

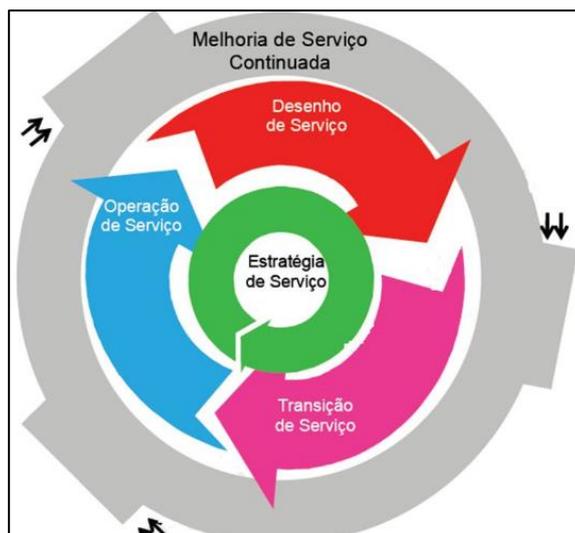


Figura 3: Ciclo de vida de serviços do ITIL (Nabiollahi e Sahibuddin, 2008)

A fase de estratégia de serviço orienta como projetar, desenvolver e implantar políticas, guias e processos de gerenciamento de serviço de TI considerando custos e riscos inerentes à Tecnologia de Informação. Esta fase indica quais serviços podem ser oferecidos, como criar valor para os clientes, como realizar investimentos estratégicos e como a qualidade de serviço pode ser definida. A fase de desenho de serviço orienta o projeto e o desenvolvimento de serviços e a definição de um processo de gerenciamento de serviço. Esta fase inclui não apenas a criação de novos serviços de TI, mas também a melhoria de serviços e a busca da conformidade com padrões para aumentar o valor dos serviços para os clientes. A fase de transição de serviço provê orientações para o desenvolvimento e a melhoria de capacidades para a operação de novos serviços de TI ou para a mudança de serviços que já estão sendo oferecidos considerando os riscos inerentes a esta mudança. A fase de operação orienta a execução de serviços de TI na prática, considerando critérios de eficiência e efetividade na entrega e suporte de serviços. A melhoria contínua é uma fase que não é representada no final do ciclo, mas sim acima dele, indicando que perpassa todas as fases e que deve ser realizada durante todo o processo de serviços. Esta fase indica como manter o valor para os clientes através do melhor design, transição e operação de serviços combinando princípios, práticas e métodos de gerenciamento da qualidade e de gestão de mudanças.

O princípio básico do ITIL é o objeto de seu gerenciamento: a infraestrutura de TI. O ITIL descreve os processos que são necessários para dar suporte à utilização e ao gerenciamento da infraestrutura de TI (Figura 4). Outro princípio fundamental do ITIL é o fornecimento de qualidade de serviço aos clientes com custos justificáveis, isto é, relacionar os custos dos serviços de tecnologia e como estes trazem valor estratégico ao negócio (Sortica et al, 2004). O interesse nesta área deve-se ao fato de que, através de metodologias (processos) padronizadas de Gerenciamento do Ambiente de TI, é possível obter uma relação adequada entre custos e níveis de serviços prestados (Sortica et al, 2004). A Figura 4 apresenta os processos do ITIL subdivididos em: Gerenciamento de Aplicações, Gerenciamento de Serviços e Gerenciamento de Infraestrutura de Tecnologia de Comunicações e de Informação (TCI).

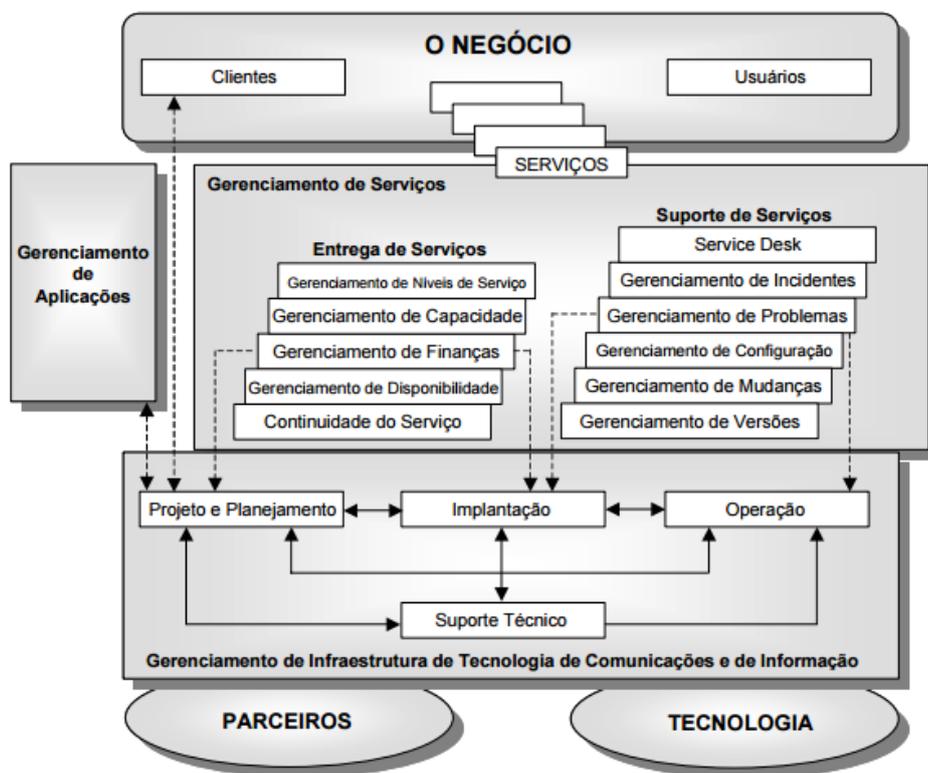


Figura 4: Processos do ITIL (Sortica et al, 2004)

Segundo Josutis (2007), a Arquitetura Orientada à Serviços (*Service Oriented Architecture - SOA*) consiste em um paradigma para projetar serviços que estão sob o domínio de proprietários distintos de modo que sejam reutilizáveis e interoperáveis, ou seja, capazes de se comunicarem de forma transparente entre si independente da tecnologia utilizada por cada um. A definição de serviço em SOA difere do significado

de serviço adotado pelo ITIL. Um serviço no contexto de SOA consiste em uma funcionalidade escrita em linguagem de programação que representa uma atividade de negócio. A descrição do serviço determinada pela entidade provedora, ou seja, pela organização que o fornece, estabelece as políticas e limitações para sua utilização.

Enquanto a governança de TI visa o governo dos serviços e dos ativos da Tecnologia de Informação, a governança de SOA visa a implementação de mecanismos de controle para planejar, implantar e dar continuidade à SOA em uma organização. Niemann et al (2008) ressaltam que a governança em SOA possui quatro fundamentos principais: políticas, processos, métricas e organização. As políticas orientam a correta execução das atividades para governar SOA. Os processos reforçam as políticas. As métricas monitoram o cumprimento das políticas e indicam quais reforços são necessários para que estas sejam seguidas. A organização é responsável por estabelecer uma cultura que suporta o processo de governança de SOA.

A governança de SOA é fundamentada nos conceitos de Governança de TI, podendo ser considerada como um subconjunto desta, mas com foco específico no projeto, desenvolvimento, fornecimento e prestação de contas de serviços de SOA (Hojaji et al, 2010b). Não há um modelo de referência padronizado e amplamente adotado em governança SOA como há na governança de TI (ITIL e CobiT), mas autores têm propostos frameworks para orientar a adoção da governança neste âmbito (Hojaji et al, 2010a), (Brow et al, 2006), (Varadan et al, 2008), (Niemman et al, 2008).

Uma organização pode implementar todas as governanças (corporativa, TI, SOA) e mantê-las alinhadas para que ao governar dimensões específicas possam auxiliar em conjunto no alcance dos objetivos estratégicos. Assim, a governança corporativa orienta o estabelecimento de políticas e o controle de gestores, a governança de TI guia o planejamento e o fornecimento de serviços da Tecnologia da Informação, a governança de SOA norteia a boa composição de funcionalidades reutilizáveis.

A governança de TI tem uma ampla interseção com a governança de ecossistemas de software (Manikas et al, 2015). Tanto a primeira quanto a segunda se preocupam com processos, papéis, procedimentos, regras e políticas associados ao desenvolvimento de produtos e de serviços de software e com o alcance eficiente de serviços estratégicos neste

domínio. Entretanto, enquanto a governança de TI se preocupa em manter o alinhamento entre o negócio e a Tecnologia da Informação de uma organização, a governança de ecossistema de software visa o alinhamento entre distintas organizações. Neste sentido, a governança de ecossistemas de software difere das governanças descritas anteriormente por englobar diferentes organizações que apesar de trabalharem colaborativamente possuem suas próprias políticas e processos.

De acordo com Albert et al (2013), o desejo de qualquer organização é entender o mercado no qual atua, as tecnologias envolvidas nesse mercado, as ferramentas que implementam essas tecnologias, os ativos produzidos relacionados a suas atividades de negócios, fornecedores, concorrentes, clientes e como esses elementos se relacionam uns com os outros. Para fazer melhor uso de seus recursos e para obter maior rentabilidade é necessário organizar estas informações. Sem uma visão suficiente da governança, essas organizações possuem dificuldade para fornecer políticas, padrões e tecnologias que apoiem o ecossistema de software (Jansen et al, 2009), (Santos e Werner, 2012).

2.4 Síntese do capítulo

Este capítulo apresentou os conceitos relacionados ao tema investigado visando proporcionar ao leitor conhecimento para contextualizar a pesquisa e compreender os resultados obtidos. Na Seção 2.1 foram apresentados os conceitos relacionados à ecossistemas de software, os tipos de ecossistemas de software definidos pela literatura (híbrido, proprietário e *open source*) e o conjunto de atores mais comuns existentes em um ecossistema de software. Conclui-se que um ecossistema de software consiste em um conjunto de atores que trabalham em colaboração em torno de uma plataforma para obter benefícios mútuos. Estes atores atuam em diversos papéis sob um caráter tridimensional que envolve aspectos sociais, de negócio e técnicos. A Seção 2.2 mostrou os conceitos associados à saúde de ecossistemas de software e sua relação com governança. Foi identificado que a governança possui uma grande influência sobre a saúde de ecossistemas de software, que consiste na habilidade de um ecossistema ser resiliente em relação a mudanças e ter propensão ao crescimento. A saúde de ecossistemas é classificada pela literatura em termos de criação de nicho, produtividade e robustez. A Seção 2.3 contextualizou os fundamentos da governança de forma geral, abordando os princípios da governança corporativa que é amplamente utilizada como base para

definição de guias e estratégias por outras áreas como Tecnologia da Informação e Arquitetura Orientada a Serviços.

3 MÉTODO DE PESQUISA

Este capítulo apresenta uma visão geral do método utilizado na condução da pesquisa, Design Science Research, do ciclo de investigação e dos instrumentos de coleta e de análise de dados do estudo.

3.1 Design Science Research

A condução desta pesquisa de doutorado foi guiada por Design Science Research. Este método orienta o ciclo de desenho, avaliação e refinamento de um artefato permitindo que o objetivo proposto seja alcançado. A justificativa para adotar Design Science Research é baseada na capacidade deste método de pesquisa auxiliar os pesquisadores a construir um artefato e melhorá-lo através de um processo contínuo de refinamento e avaliação. Design Science Research consiste em um método de investigação que visa a criação de artefatos inovadores para resolver problemas ou propor melhorias (Iivari e Venable, 2009). De acordo com Järvinen (2007), Design Science possui as seguintes características:

- Produz artefatos técnicos,
- Produz conhecimento (conceitos, construções, modelos e métodos),
- Foca na construção e avaliação de artefatos, sendo estas as principais atividades realizadas em Design Science,
- Resolve problemas de construção (produzindo inovações) e problemas de melhoria (melhoria de artefatos existentes),
- Produtos gerados através de Design Science são avaliados segundo critérios de valor e utilidade.

Design Science Research se diferencia do desenvolvimento de projetos rotineiros pela geração de soluções com base em rígido rigor metodológico. Este método se baseia em uma abordagem de projeto explicitamente organizada, racional e totalmente sistemática não apenas referente ao aproveitamento do conhecimento científico inerente à criação de artefatos, mas ao próprio planejamento e execução do projeto como uma atividade científica (Cross, 2001).

Design Science Research permite a criação e avaliação de artefatos propostos para resolver problemas organizacionais (Hevner, 2007) e pode ser descrito como uma abordagem que visa o desenvolvimento de estratégias para alcançar propósitos humanos (March e Smith 1995). Wieringa (2014) considera que Design Science visa também o aperfeiçoamento de coisas existentes para atender a objetivos humanos de uma forma melhor. Wieringa (2014) ainda afirma que Design Science enfatiza a conexão entre conhecimento e prática, mostrando que é possível produzir conhecimento científico através da concepção de coisas úteis. Holmström et al (2009) preconiza que Design Science Research é a pesquisa que busca (i) explorar novas alternativas de solução para resolver problemas, (ii) explicar este processo exploratório, e (iii) melhorar o processo de resolução de problemas.

Conforme afirma Hevner (2010), o objetivo de Design Science Research é a construção de artefatos relevantes em termos de valor e de utilidade tanto em nível prático quanto teórico. O artefato gerado durante a execução do ciclo de Design Science é um constructo, um modelo, um método ou uma aplicação de um objeto (Gregor e Jones 2007), (March and Smith 1995), (Kuechler e Vaishnavi 2008), (Walls et al. 1992). Constructos são definidos como conceitos, vocabulários ou símbolos que caracterizam uma determinada área (Kuechler e Vaishnavi, 2008). Segundo Ostrowski et al (2012), modelos fornecem uma abstração da realidade e representam a conexão entre o problema e a solução, permitindo visualizar os efeitos das decisões projetadas e as mudanças passíveis de serem realizadas no mundo real. Um método é definido como um conjunto de passos necessários para realizar uma tarefa (Gregor e Jones 2007). Na aplicação, o objeto a ser instanciado pode ser um protótipo, uma ferramenta ou o próprio modelo previamente construído. Ao utilizar Design Science Research, o pesquisador deve avaliar qual tipo de artefato deve ser gerado e o contexto no qual ele será utilizado (March e Smith 1995).

Segundo Hevner et al. (2007), o artefato resultante de Design Science Research deve possuir duas características essenciais: relevância e inovação. É necessário que o artefato resolva um problema importante de forma inovadora, eficiente e eficaz (Geerts, 2011). Venable et al (2012) acrescenta uma terceira característica: rigor. O processo envolvido na construção e avaliação do artefato deve ser rigoroso (Venable et al, 2012).

Conforme afirma Hevner (2007), é o rigor da construção do artefato que distingue Design Science Research da prática de concepção de produtos na Tecnologia de Informação.

3.2 Ciclo de Design Science

Visando garantir o rigor da pesquisa, é necessário adotar um processo de Design Science Research que deverá orientar desde a construção até a avaliação do artefato (Hevner, 2007). Há vários processos de Design Science Research propostos na literatura, tais como: (Wieringa, 2014), (Holmström et al, 2009), (Venable et al, 2012), (Geerts, 2011). Estes processos também são chamados de ciclos de Design Science. O ciclo de design Science é iterativo-incremental, ou seja, cada fase pode ser revisitada várias vezes ao longo da execução e evolução da pesquisa. Os ciclos de design science propostos pelos autores anteriormente mencionados diferenciam entre si com relação a quantidade de fases e com relação aos procedimentos a serem realizados em cada fase.

Nesta pesquisa foi adotado o ciclo de Design Science Research proposto por Wieringa (2014) por ser uma proposta já instanciada para o contexto de pesquisas em engenharia de software e sistemas de informação. O ciclo proposto por Wieringa (2014) é baseado nas seguintes etapas: investigação do problema, design da solução, validação da solução, implementação da solução e avaliação da implementação (Figura 5). Após a execução do primeiro ciclo, novos ciclos de refinamento são realizados.

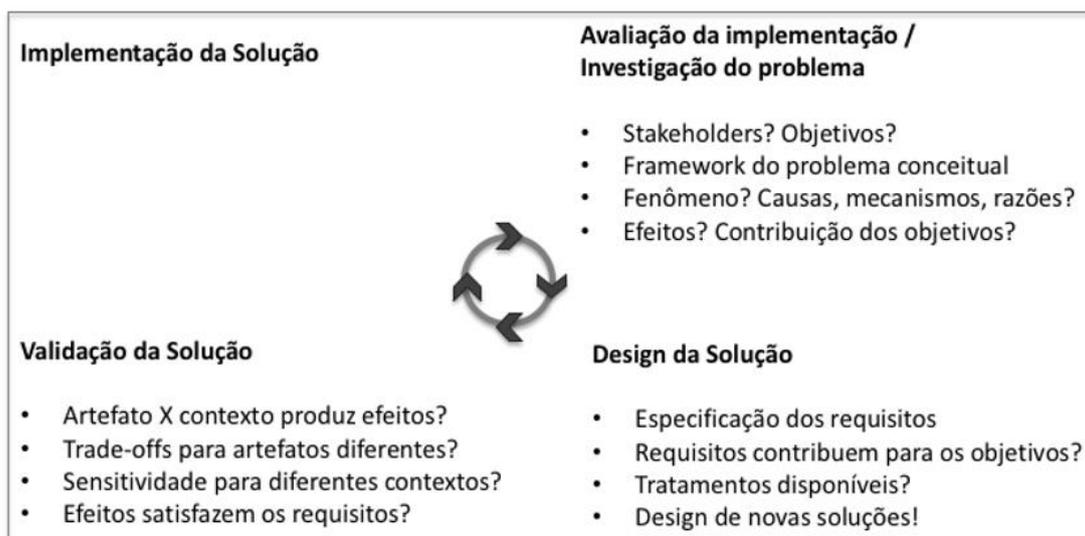


Figura 5: Fases do ciclo de Design Science (Wieringa, 2014).

Para cada fase foram definidos os métodos de pesquisa (Figura 6), assim como os instrumentos de coleta e de análise de dados. Cada método é descrito dentro das próximas seções que correspondem às fases do ciclo de vida de Design Science.

Na fase de *investigação do problema* foi realizada uma Revisão Sistemática da Literatura com base nas orientações de Kitchenham e Charters (2007). A revisão, que agregou buscas automáticas e manuais foi realizada nas bases eletrônicas e conferências mais conhecidas pela comunidade acadêmica da área. O Capítulo 4 apresenta os resultados desta etapa.

Na fase de *design da solução*, apresentada no Capítulo 5, foi construído o artefato selecionado para responder à questão de pesquisa QP1.4. O artefato selecionado foi um modelo conceitual para apoiar a governança de ecossistemas de software. Modelos conceituais são úteis para representar e explicar graficamente constructos de um domínio específico e seus relacionamentos de forma intuitiva. A construção de proposições para embasar cada relacionamento entre elementos do artefato facilitou a validação realizada posteriormente. A construção do artefato seguiu as orientações de Noy e McGuinness (2001) e Sjoberg et al. (2018) e foi baseada nos resultados obtidos a partir da Revisão Sistemática da Literatura.

A fase de *validação da solução* descrita no Capítulo 6 consistiu em verificar e refinar o artefato proposto por meio de um survey com especialistas. Os resultados foram obtidos através da aplicação de um questionário sobre as proposições elaboradas e sobre o grau de compreensão do modelo conceitual. As orientações de Kitchenham e Pfleeger (2008) guiaram a condução do survey e os critérios de Sjoberg et al (2008) foram utilizados para avaliar o artefato.

A fase de *implementação da solução* (Capítulos 7) foi conduzida por meio de entrevistas semiestruturadas com praticantes que atuam em ecossistemas de software. O roteiro de entrevistas incluído no Apêndice C, indica quais questões foram incluídas. Percebe-se que as questões mencionam elementos e relacionamentos do modelo conceitual dando ênfase a sua aplicação prática. Portanto, há relação entre as perguntas e o modelo conceitual, mas buscou-se fazer essa ligação de forma sutil, visto que há termos que são teóricos (exemplo: modelo de associação) e que poderiam ser conhecidos pelos

praticantes por outros sinônimos. As afirmações dos praticantes forneceram insumos para instanciar o artefato proposto. Esta fase foi complementada com estudos de caso (Merriam, 2002), realizados nos ecossistemas SAP, GNOME e Eclipse por meio de pesquisa documental e os dados foram tratados utilizando-se a técnica de análise de conteúdo (Flick, 2009). Os estudos de caso possibilitaram a aplicação do artefato proposto para os contextos investigados. A fase de *avaliação da implementação* contendo os resultados da fase anterior que são discutidos através de análise realizada pela autora desta tese também é descrita no Capítulo 7.

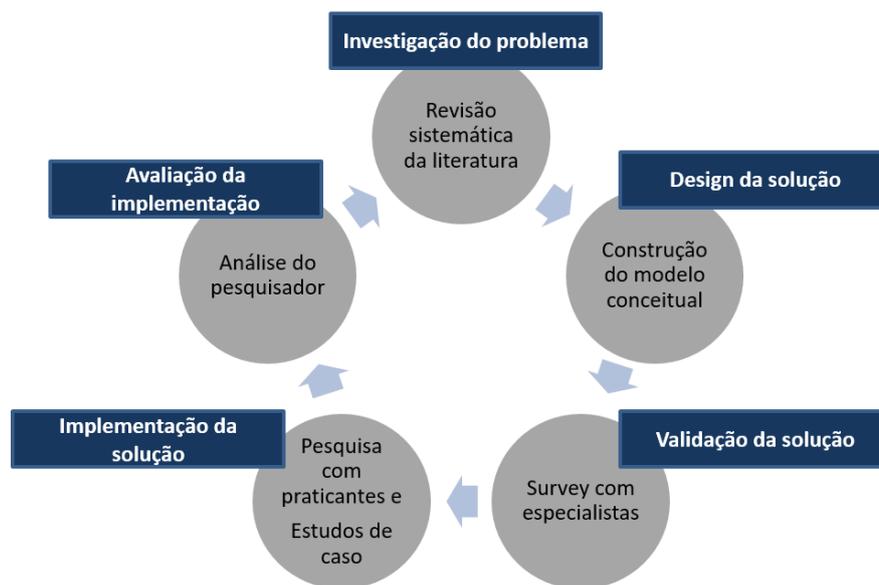


Figura 6: Métodos de pesquisa associados às fases de Design Science Research

Wieringa (2014) afirma que até chegar a fase de avaliação da implementação de forma definitiva, Design Science resume-se em avaliar e refinar continuamente o modelo/artefato/método até que ele atenda seu propósito. Entretanto, para realizar as atividades do ciclo de Design Science é necessário conhecer o contexto social dos *stakeholders* e os objetivos do projeto. Também é necessário que o pesquisador domine o contexto do conhecimento, ou seja, as teorias e conceitos que embasam o tema de pesquisa a ser estudado.

A Figura 7 mostra que o contexto social desta pesquisa envolve os participantes de um ecossistema de software (*keystone*, fornecedor, desenvolvedor e cliente). Os métodos utilizados para avaliar e refinar o artefato foram survey com especialistas e estudos de caso. O artefato gerado consistiu em um modelo conceitual para apoiar a

governança de ecossistemas de software. O processo de construção, refinamento e avaliação do modelo conceitual de governança para ecossistemas de software foi orientado pelas questões de pesquisa definidas no Capítulo 1.

Os fundamentos teóricos que embasam o contexto de conhecimento da pesquisa envolvem os domínios de ecossistemas de software e de governança de ecossistemas de software.

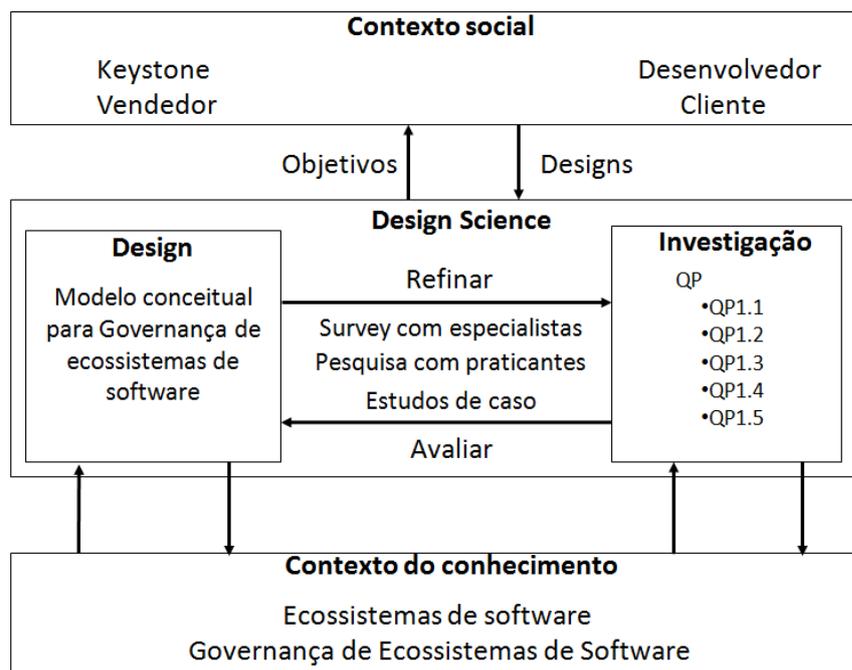


Figura 7: Aplicação do framework de Design Science de Wieringa (2014) para este estudo

3.2.1 Investigação do problema

Esta fase envolve a busca de informações sobre o problema a ser investigado, mas sem ainda tentar solucioná-lo (Capítulo 4). A investigação visa o entendimento e a descrição do problema, além da explicação do que possivelmente pode acontecer se nada for feito para resolvê-lo. Nesta fase são identificados os principais envolvidos e afetados pelo problema (*stakeholders*), os objetivos a serem atingidos, as causas/razões que originam o problema, os efeitos e as contribuições quando os objetivos forem alcançados.

O problema desta pesquisa consiste na necessidade de conhecer os mecanismos e elementos envolvidos na governança de ecossistemas de software para que possam ser

governados de modo a permitir o aumento da saúde do ecossistema e o seu crescimento ao longo do tempo. Os principais *stakeholders* são os atores do ecossistema (*keystone*, desenvolvedores, fornecedores, clientes). O fenômeno investigado consiste na governança de ecossistemas de software. As causas que originam o problema é a ausência na academia e na prática de um corpo de elementos bem definidos e com relacionamentos entre si que proporcionem uma visão holística sobre como realizar a governança de ecossistemas de software. Uma vez que esses elementos e seus relacionamentos sejam conhecidos e governados, atores de um ecossistema de software poderão conduzir a governança do ecossistema de forma sistemática, podendo aumentar a saúde do mesmo.

Para melhor investigar o problema, foi realizada uma revisão sistemática da literatura. A Revisão Sistemática da Literatura (em inglês: Systematic Literature Review - SLR) é um método amplamente utilizado para responder questões de pesquisa específicas, examinando um tópico de investigação particular ou fenômeno de interesse através da sistemática identificação, avaliação e interpretação de estudos disponíveis (Kitchenham e Charters, 2007). As quatro primeiras subquestões de pesquisa orientaram a condução da revisão sistemática da literatura:

- *QP1.1. Como a governança de ecossistemas de software é definida?*
- *QP1.2. Quais são os mecanismos propostos pela literatura para governar ecossistemas de software?*
- *QP1.3. Como a saúde de ecossistemas é caracterizada pela literatura?*
- *QP1.4. Quais métricas para medir a saúde de ecossistemas de software foram propostas pela literatura?*

Estas questões forneceram uma visão geral de como a comunidade acadêmica define governança e saúde de ecossistemas de software visto que os dois temas estão amplamente relacionados. Os mecanismos identificados por meio da revisão sistemática da literatura serviram de insumos para a extração de elementos de governança do ecossistema de software. Os resultados da SLR foram publicados no artigo Alves et al (2017) e no capítulo de livro Alves et al (2017).

3.2.2 Design da Solução

Esta fase consiste na proposição e especificação da solução para resolver o problema (Capítulo 5). A solução pode ser proposta através de descrição em linguagem

natural, diagramas, modelos matemáticos, protótipos ou pela combinação destes. A solução nesta pesquisa é apresentada por meio de um modelo conceitual de governança de ecossistemas de software construído por meio da combinação das propostas de Noy e McGuinness (2001) e Sjoberg et al. (2018), conforme apresentado na Figura 8.

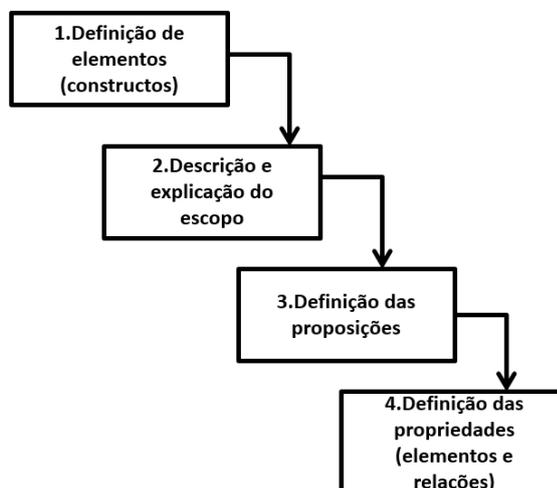


Figura 8:. Passos da construção do modelo conceitual para governança de ecossistemas de software.

A quinta subquestão de pesquisa (*QPI.5. Quais são os elementos envolvidos na governança de ecossistemas de software e como estes elementos estão relacionados?*) orientou a condução desta etapa de design da solução. Para identificar os elementos envolvidos na governança do ecossistema de software, foram adotados os resultados da Revisão Sistemática da Literatura cuja lista de artigos incluídos consta no Anexo A. Estes estudos primários apoiaram a construção do modelo conceitual.

Para garantir o correto entendimento da pesquisa, é relevante distinguir mecanismos de elementos. Mecanismo de governança é caracterizado por um verbo e consiste em uma ferramenta gerencial utilizada pelos atores com o objetivo de influenciar a saúde de um ecossistema de software (Ex: gerir inovação, nutrir colaboração). Elemento de governança é indicado por um substantivo e consiste em um constructo a ser governado. No modelo conceitual é representado por classes, conforme a notação UML (Ex: inovação, colaboração).

De acordo com a Figura 8, para definir cada elemento do modelo, foram considerados tanto os estudos primários, bem como artigos que descrevem elementos

encontrados na literatura em geral porque nem todos os estudos incluídos na revisão sistemática forneceram definições precisas dos elementos investigados (Passo 1). Posteriormente foram extraídos conjuntos de citações dos artigos incluídos na SLR para descrever o contexto de cada elemento (Passo 2). Depois foram criadas proposições para sintetizar as citações (Passo 3). As proposições consistem na essência do modelo conceitual. Todas as proposições emergiram dedutivamente dos artigos primários analisados na revisão sistemática da literatura e se caracterizam por serem empiricamente testáveis, ou seja, passíveis de refutação ou aceitação na observação do contexto investigado. As proposições servem para explicar os fenômenos que são o foco da teoria em construção por meio da simples associação entre elementos e demonstram como estes elementos interagem entre si (Sjoberg et al, 2008). Cada proposição recebeu um código no formato Pn (por exemplo, P1, P2,..., Pn), em que n representa a ordem numérica da proposição no modelo conceitual para governança de ecossistemas de software. Na etapa final (Passo 4) foi utilizada a Linguagem de Modelagem Unificada (do inglês Unified Modeling Language - UML) para representar os elementos e seus respectivos relacionamentos. A notação UML (<https://www.uml.org/>) foi escolhida dada sua ampla adoção na elaboração de modelos conceituais, bem como sua facilidade de uso e de interpretação (Tilakaratna e Rajapakse, 2017).

3.2.3 *Validação da Solução*

Esta fase consiste em avaliar se a solução desenhada de fato atende os objetivos esperados pelos *stakeholders* (Capítulo 6). Questões relevantes a serem consideradas nesta fase são:

- O projeto implementado no contexto do problema, satisfaz os critérios identificados na fase de investigação?
- Projetos ligeiramente diferentes poderiam satisfazer os critérios identificados se implementados no mesmo contexto?

Estas questões remetem a avaliação da validade interna e validade externa do projeto. De acordo com Wieringa (2014), uma das formas mais simples de validar um artefato é mediante a execução de survey com especialistas. Os especialistas são responsáveis por “observar”, por imaginação, a forma como o artefato se comportaria no mundo real (Wieringa, 2014).

O modelo conceitual foi validado por meio de um survey qualitativo com especialistas da área de ecossistemas de software. De vinte especialistas convidados através de email, dezessete participaram da pesquisa. Os especialistas escolhidos atuam na academia e na indústria da América do Sul, América do Norte e Europa. Os especialistas também são membros de comitês de conferências e workshops de ecossistemas de software. O survey elaborado consistiu de três partes (Apêndice B):

- **Parte 1 - Questões demográficas:** nove questões demográficas sobre os participantes que permitiu explorar diferenças nas respostas de acordo com as características do especialista.
- **Parte 2 - Avaliação de proposições:** 15 questões para avaliar as proposições do modelo de governança de ecossistemas de software. Cada pergunta tinha três possíveis respostas: "concordo", "não concordo" e "não sei".
- **Parte 3 - Avaliação do modelo completo:** dez perguntas adicionais com três respostas possíveis: "sim", "não" e "não sei". Estas perguntas visaram avaliar o modelo conceitual de acordo com os seguintes critérios (Sjøberg et al., 2008):
 - **Testabilidade:** Consiste no quanto uma teoria construída é passível de ser empiricamente refutada.
 - **Suporte empírico:** refere-se ao grau em que uma teoria é apoiada por estudos empíricos que confirmam sua validade.
 - **Poder explicativo:** Consiste no quanto o escopo de uma teoria é simples, com poucas suposições ad hoc e relacionado com o que já é bem entendido na área.
 - **Parcimônia:** Consiste no grau em que uma teoria é economicamente construída com o mínimo de conceitos e proposições.
 - **Generalidade:** refere-se a amplitude do escopo de uma teoria e o grau em que é independente de configurações específicas.
 - **Utilidade:** refere-se ao grau em que uma teoria suporta as áreas relevantes da indústria de software.

Um formulário do Google foi utilizado para elaborar o survey. Além disso, foi criado um site contendo instruções, informações adicionais, glossário de elementos e uma breve explicação da notação UML para ajudar os especialistas a responder o survey (<https://sites.google.com/site/softwareecogovemodel/home> - Apêndice A). Antes de enviar o convite para especialistas, foi realizado um estudo piloto para avaliar e calibrar as questões, detectar perguntas ambíguas e consertar instruções inadequadas, conforme orientações de Kitchenham e Pfleeger (2008). O estudo piloto foi realizado com um pesquisador que possui grande experiência em modelagem conceitual e governança. Após os ajustes, os instrumentos do survey foram liberados para os especialistas.

Os dados obtidos foram tratados por meio de análise temática (Cruzes e Dybå, 2011). Uma palavra chave representante da ideia central das opiniões dadas pelos especialistas foi selecionada e definida como tema. Após comparações consecutivas, o tema foi associado a um conjunto de opiniões considerando-se os sinônimos relacionados a este tema. Por exemplo, palavras e verbos citados pelos entrevistados, referindo-se à "gestão" e "controle" foram associados ao tema "governança". Termos como "evolução", "estabilidade" e "propensão ao crescimento" foram associados ao tema "saúde".

O seguinte critério foi utilizado para excluir proposições: se a quantidade de respostas correspondentes a "não sei" mais a quantidade de respostas equivalente a "não concordo" fosse maior do que a quantidade de "concordo", a proposição deveria ser excluída. Para modificar as proposições, foram analisados se os argumentos dos especialistas eram sólidos e se possuíam exemplos claros. Portanto, os feedbacks dos especialistas orientaram a modificação, assim como o aprimoramento de proposições e, consequentemente, o refinamento do modelo conceitual.

2.3.4 Implementação da Solução

Esta fase consiste na aplicação da solução no contexto investigado (Capítulo 7). Wieringa (2014) afirma que a definição de implementação pode ser confusa por ser normalmente considerada como a aplicação de fato do artefato em um contexto real. Entretanto, a definição do que consiste em uma implementação depende de qual e para que a solução foi projetada (Wieringa, 2014).

Nesta pesquisa, definiu-se por implementação a aplicação do modelo conceitual para o contexto de ecossistemas *open source*, híbridos e proprietários. Para tal, foi

realizado um conjunto de entrevistas com profissionais atuantes em ecossistemas de software como gestores, desenvolvedores independentes ou como colaboradores de empresas certificadas por ecossistemas de software. Estes profissionais foram denominados de “praticantes”. A realização das entrevistas teve o objetivo de investigar a aplicabilidade do modelo conceitual em diferentes ecossistemas de software onde os profissionais atuam, tais como Apple, Microsoft, Software AG, Google e Eclipse. Os praticantes selecionados foram contatados por meio do LinkedIn e e-mails. O principal critério para selecionar um praticante foi sua ativa participação em um ecossistema de software. Para isto, foi realizada uma análise do perfil do praticante na rede social anteriormente mencionada. Também foram levadas em consideração sugestões de pesquisadores da área de ecossistemas de software sobre possíveis praticantes que poderiam enriquecer o estudo.

Cinco de um total de 25 praticantes convidados participaram da pesquisa. Para entrevistá-los, foi desenvolvido roteiro de entrevista semiestruturado (Apêndice C), incluindo questões introdutórias demográficas (por exemplo: papel, organização) bem como questões relacionadas à experiência dos praticantes no ecossistema de software. Estas questões abordaram as proposições do modelo não para testá-las, como ocorreu na fase anterior, mas para investigar a ocorrência e dinâmica dos elementos de governança no contexto de ecossistemas de software no qual os praticantes atuam.

Os praticantes foram orientados a consultarem o site do modelo conceitual para compreender os elementos e relacionamentos envolvidos e para terem acesso ao glossário em caso de dúvidas sobre a definição de algum elemento.

A técnica de entrevista semiestruturada permitiu a elaboração de um roteiro com perguntas pré-definidas que puderam ser alteradas durante a conversação. Todas as entrevistas foram registradas por meio de um gravador de áudio com o consentimento dos praticantes e depois foram transcritas com o apoio do Software Scribe Express (<https://express-scribe.br.uptodown.com>).

Para analisar os dados das entrevistas também foi utilizada a técnica de análise temática sob as diretrizes de Cruzes e Dyba (Tore). Nesta fase, extratos relevantes de entrevistas foram agrupados e rotulados formando gradualmente um conjunto de

categorias relacionadas. O software Weft QDA apoiou o procedimento de análise dos dados.

A análise da aplicabilidade do modelo conceitual por meio de entrevistas com praticantes permitiu obter novos insights sobre a ocorrência dos elementos de governança na prática. Entretanto, julgou-se necessário, ainda nesta fase de implementação da solução, investigar um conjunto de ecossistemas através de estudos de caso. A condução dos estudos de caso fora executada pela autora desta tese. O propósito foi analisar como os elementos, relacionamento e proposições propostas pelo modelo conceitual ocorrem na prática em diferentes ecossistemas de software. Para tal, o modelo conceitual foi implementado por meio de aplicações para os ecossistemas do Eclipse, SAP e GNOME. Estes três ecossistemas foram escolhidos por terem grande relevância no mundo corporativo e por serem amplamente pesquisados por autores da área de ecossistemas de software (Manikas, 2016). Os ecossistemas Eclipse e GNOME, em especial permitem amplo acesso a documentação a partir de repositórios abertos.

De forma mais específica, o Eclipse foi escolhido porque apesar de ser considerado um ecossistema *open source*, é governado por um conselho formado por membros que precisam pagar uma taxa para fazer parte dele. Entretanto, qualquer desenvolvedor pode fazer parte da comunidade do Eclipse sem a obrigatoriedade de pagamento. Logo, este ecossistema possui caráter híbrido. O GNOME é caracterizado como totalmente *open source* e o SAP é um exemplo de ecossistema proprietário. As diferenças existentes entre estes ecossistemas ampliam a possibilidade de tornar o modelo conceitual mais rico e adaptável às características de ecossistemas de software distintos.

O método de estudo de caso foi selecionado por possibilitar que fenômenos complexos sejam pesquisados dentro de seus contextos (Baxter e Jack, 2008). Estudos de caso são fortemente recomendados por Pries-Heje et al (2008), como estratégia de avaliação de pesquisas baseadas em Design Science por permitir aprofundar a avaliação do objeto de estudo considerando-se uma realidade concreta.

Nesta pesquisa, os estudos de caso sobre os ecossistemas Eclipse, GNOME e SAP tiveram um caráter exploratório (Merriam, 2002), uma vez que visou investigar como a governança de software ocorre em cada caso estudado. Os dados foram coletados por meio de pesquisa documental. Esta técnica de coleta se caracteriza por restringir a análise

de documentos que se constituem de fontes primárias (Lakatos e Marconi, 2010). Como fonte de dados foram pesquisados documentos em portais dos ecossistemas selecionados, websites sobre notícias de TI e artigos acadêmicos sobre os ecossistemas em estudo. Os dados foram tratados por meio de análise de conteúdo. Segundo Flick (2009) a análise de conteúdo é um dos procedimentos clássicos para analisar o material textual agregado em produtos da mídia e em documentos formais.

A condução dos estudos de caso envolveu as seguintes etapas:

- I. Explorar a documentação do ecossistema de software selecionado;
- II. Avaliar as proposições do modelo conceitual considerando a ocorrência de elementos e relacionamentos propostos dentro do ecossistema estudado;
- III. Instanciar o modelo conceitual para o ecossistema investigado incluindo elementos que ainda não tenham sido identificados anteriormente;
- IV. Identificar se novos elementos são passíveis de inclusão definitiva no modelo conceitual ou se não devem ser incluídos por comprometer o seu grau de generalização.

3.2.5 Avaliação da Implementação

Esta fase consiste em avaliar a efetividade da aplicação da solução no contexto investigado (Capítulo 7). Na fase anterior mencionou-se que o modelo conceitual foi implementado por meio da aplicação para o contexto de três estudos de caso. Na avaliação da implementação, utiliza-se os resultados obtidos na fase de implementação para refiná-lo. Para executar esta etapa, a autora desta tese avaliou os resultados dos estudos de casos realizados e o modelo conceitual foi ajustado de acordo com as necessidades identificadas.

3.2 Síntese do capítulo

Neste capítulo foi apresentado o método de pesquisa utilizado, assim como os instrumentos de coleta e de análise de dados. O método de pesquisa escolhido consistiu em Design Science Research. As fases do ciclo deste método foram apoiadas por Revisão Sistemática da Literatura, Survey com especialistas, pesquisa com praticantes e estudos de caso. A contextualização do método de Design Science Research foi realizada na

Seção 3.1. O ciclo de Design Science escolhido foi apresentado na Seção 3.2. Cada fase do referido ciclo, classificadas em investigação do problema, validação da solução, implementação da solução e avaliação da implementação, foi descrita nas Subseções 3.2.1 a 3.2.5.

4. REVISÃO SISTEMÁTICA DA LITERATURA

Neste capítulo, é descrita a revisão sistemática da literatura realizada durante a fase de investigação do problema do ciclo de Design Science, conforme descrito na Figura 5 do capítulo 3. A revisão sistemática da literatura é o meio pelo qual o pesquisador pode realizar um mapeamento do estado da arte, do conhecimento existente e do que já foi desenvolvido em determinada área. As etapas para a realização de uma revisão sistemática da literatura envolvem o planejamento, a execução e a análise de resultados. A fase de planejamento consiste em projetar todos os detalhes necessários para que a revisão seja realizada e requer (Kitchenham e Charters, 2007):

- **Definição do protocolo de revisão:** consiste em especificar as questões de pesquisa a serem respondidas e o método a ser seguido para respondê-las;
- **Definição da estratégia de busca:** Consiste em especificar o processo de busca automática dos artigos. Para complemento, podem ser definidas estratégias de busca manual.
- **Definição dos critérios de inclusão e exclusão:** Consiste em estabelecer critérios que especificam o processo de decisão sobre quais publicações encontradas pela estratégia de busca serão incluídas na revisão;
- **Definição de critérios de qualidade:** Classificação das publicações selecionadas de acordo com uma taxonomia para avaliar o nível do artigo em termos de consistência dos objetivos de pesquisa, clareza da descrição do contexto, adequação do método selecionado ao tipo de estudo, etc.
- **Definição do processo de interpretação dos resultados:** Consiste em definir como os dados relevantes serão extraídos, analisados, armazenados e reportados.

A fase de execução envolve a busca, a seleção e avaliação de estudos primários em conformidade com critérios de inclusão e exclusão definidos no protocolo. Uma vez selecionados os estudos, os dados dos artigos incluídos podem ser extraídos e sintetizados durante a fase de análise dos resultados.

Nesta tese, a revisão sistemática da literatura teve como objetivo sintetizar o número crescente de estudos no domínio dos ecossistemas de software (Hansen et al., 2014) e (Bosch 2012). Em particular, foi investigado quais tipos de mecanismos de governança têm sido propostos na literatura. Esta pesquisa difere de revisões sistemáticas anteriores que tinham o objetivo de estudar o campo dos ecossistemas de software de uma maneira geral (Barbosa et al., 2013) e de caracterizar a saúde do ecossistema (Hyrynsalm et al, 2015).

4.1 Fases da Revisão Sistemática da Literatura

Nesta seção são mostradas as etapas envolvidas na condução da revisão sistemática da literatura. A seção agrega a elaboração do protocolo de pesquisa no qual foram definidas as questões a serem investigadas, as strings de busca, os critérios de inclusão e de exclusão, além do processo de busca de extração e de análise dos dados.

4.1.1 Definição das questões de pesquisa

As três primeiras subquestões de pesquisa definidas neste estudo orientaram a condução da revisão sistemática da literatura:

- *QP1.1. Como a governança de ecossistemas de software é definida?*

Na QP1.1, foram apresentadas e discutidas as definições disponíveis para governança de ecossistemas de software propostas pelos estudos primários. Em seguida, as definições disponíveis foram comparadas e foi proposta uma definição integrada para o termo ‘governança de ecossistema de software’. Os mecanismos de governança foram classificados com base nos dados coletados dos estudos primários seguindo uma abordagem de análise temática (Cruzes e Dybå, 2011).

- *QP1.2. Quais são os mecanismos propostos pela literatura para governar ecossistemas de software?*

Os objetivos ao responder à segunda questão (QP1.2) consistiram em identificar os mecanismos propostos pela literatura atual que regem os ecossistemas de software e classificar o mecanismo de governança adotado por ecossistemas proprietários e de código aberto.

Selecionar mecanismos de governança de ecossistemas de software apropriados pode nutrir a saúde do ecossistema (Bosch e Bosch-Sijtsema, 2014).

- *QP1.3. Como a saúde é caracterizada pela literatura?*

Na terceira questão (QP1.3), foi examinado como a saúde de um ecossistema de software é definida por artigos da área de ecossistemas de software.

- *QP1.4 Quais métricas para medir a saúde de ecossistemas de software foram propostas pela literatura?*

Na quarta questão (QP1.4) foram investigadas as métricas disponíveis para medir a saúde de ecossistemas de software. Essas métricas são consideradas os principais indicadores de desempenho dos gerentes e determinam quão bem eles governam seus ecossistemas de software.

4.1.2 Processo de busca

Seguindo a orientação de Kitchenham e Charters (2007), um protocolo foi desenvolvido para especificar os passos e critérios envolvidos na realização da revisão sistemática. Um protocolo de revisão inclui detalhes de como diferentes tipos de estudos devem ser encontrados, avaliados e sintetizados (Brereton et al, 2007). No protocolo foram definidas as questões de pesquisa, as estratégias de busca adotadas para identificar artigos primários relevantes, a *string* de busca para usar nas bases de dados, os critérios de exclusão / inclusão e os critérios de avaliação de qualidade. Além disso, foi determinado o processo de extração e de análise de dados.

A estratégia para coletar estudos incluiu quatro etapas: (i) busca automática nas bases de dados eletrônicas, (ii) pesquisa manual de revistas, conferências e workshops, (iii) análise das listas de referência de outros estudos secundários em ecossistemas de software.

As seguintes bases de busca foram selecionadas para a execução da pesquisa automática: ACM Digital Library, IEEE Xplore Digital Library, Science Direct e SpringerLink. As *strings* de busca adotadas foram: "software ecosystem" e "platform

ecosystem". Optou-se por usar termos genéricos para evitar o risco de limitar o processo de pesquisa. Nos estágios iniciais da revisão, tentou-se usar a string "software ecosystem" AND "governance". No entanto, considerou-se que o resultado do uso combinado dessas palavras-chave foi muito limitado. Em particular, não foram identificados muitos artigos relevantes. Mesmo sabendo que uma *string* de busca genérica iria retornar um grande número de documentos, tornando o processo de seleção muito demorado, preferiu-se adotar uma estratégia mais abrangente possível. Observando recomendações de experiências anteriores sobre as limitações relacionadas a indexação das bases de dados de engenharia de software (Brereton et al, 2007), foi realizada uma busca manual nas seguintes revistas, conferências e workshops:

- Information and Software Technology;
- Journal of Software Systems;
- International Conference on Software Business;
- International Workshop on Software Ecosystem.

Para complementar a busca manual foi incluída a estratégia de análise das referências das seguintes revisões em ecossistemas de software: (Manikas and Hansen, 2013), (Barbosa et al., 2013), (Hyrnsalm et al., 2015), (Franco - Bedoya et al., 2014) e (Fotrousi, 2014). Embora o escopo e questões de pesquisa destas revisões sejam diferentes das definidas nesta proposta, a análise da lista de artigos referenciadas nessas revisões visou corrigir quaisquer omissões de estudos oriundas dos outros procedimentos de busca utilizados. Sete estudos [S13, S17, S18, S25, S26, S41, S61] foram obtidos a partir da análise dos estudos secundários descritos acima.

4.1.3 Avaliação da qualidade dos artigos

Para avaliar a qualidade dos artigos foi adotada a taxonomia utilizada por Brhel et. al (2015). Os critérios de avaliação da qualidade são descritos na Tabela 2.

Tabela 2: Critérios da avaliação da qualidade dos artigos

-
1. Os objetivos de pesquisa são claros? Há uma questão de pesquisa explicitamente verbalizada?
 2. Há uma adequada descrição do contexto onde a pesquisa foi realizada?
 3. Aplicável apenas para pesquisas empíricas:
 - 3.1. O método de pesquisa é explicitamente declarado?
 - 3.2. Qual método de pesquisa foi escolhido?
-

4.1.4 Extração e análise dos dados

Foi utilizada uma planilha do Excel para armazenar os dados dos artigos selecionados. Dois pesquisadores extraíram os dados dos estudos. Várias reuniões de discussão foram realizadas com todos os pesquisadores envolvidos para comparar extrações, esclarecer incertezas, chegar a um acordo sobre as divergências e executar verificações de inconsistências. Para responder a QP1, simplesmente procurou-se o termo "governança" nos estudos primários e foi verificado se o artigo forneceu uma definição para a governança no contexto dos ecossistemas de software. Para responder a QP2, utilizou-se análise temática como método de síntese, seguindo os passos recomendados por Cruzes e Dybå (2011). Códigos relevantes de estudos primários foram identificados. Em seguida os códigos foram incorporados em temas-chave para abordar a QP2 no qual considerou-se que os mecanismos de governança foram encapsulados em termos semelhantes, tais como: "estratégia" "gerir", "governar", "controlar" e "fatores críticos". O termo "saúde" foi identificado nos estudos primários para responder a QP3 e para responder a QP4, foram usadas definições de produtividade, robustez e criação de nicho [S76, S25, S77] para classificar as métricas de saúde.

4.1.5 Critérios de inclusão e exclusão

Foram adotados os seguintes critérios de inclusão para selecionar os artigos: (i) estudos escritos em Inglês e (ii) estudos que respondem a pelo menos uma questão de pesquisa. Os critérios de exclusão adotados foram: (i) estudos secundários (mapeamento e revisões sistemáticas), (ii) relatórios técnicos, resumos e *whitepapers*, (iii) artigos duplicados referentes ao mesmo estudo.

Os critérios de inclusão e exclusão foram aplicados na análise dos títulos, do resumo e das palavras-chave dos artigos encontrados nas bases de busca definidas, utilizando as *strings* adotadas. Isto permitiu determinar a relevância dos estudos e a resposta a pelo menos uma questão de pesquisa. Observou-se que o tema global de ecossistemas de software já está bem estabelecido na literatura. No entanto, o conceito específico de governança ainda é bastante difuso. O conteúdo completo de artigos foi

também examinado durante o processo de seleção para garantir que o estudo está relacionado com algum aspecto da governança de ecossistema de software.

A figura 6 apresenta os passos realizados e o número de estudos selecionados em cada etapa. No total as buscas automática e manual retornaram inicialmente 997 artigos. A busca automática foi realizada a partir de abril de 2015 e não foi definida restrição com relação ao período de abrangência da pesquisa. Todo o processo de condução da revisão sistemática teve duração de quase dois anos, tendo sido iniciada no início de abril de 2015 e finalizada em dezembro de 2016. Artigos baseados em títulos e resumos que não atendiam aos critérios de inclusão foram excluídos. Os artigos restantes que atendiam aos critérios de inclusão foram avaliados considerando-se o título e o resumo. Artigos que provavelmente responderiam a pelo menos uma questão de pesquisa foram selecionados para leitura de todo o conteúdo. Sempre que havia dúvida com relação ao teor do artigo este era incluído para uma análise mais aprofundada do seu conteúdo completo. Após essa etapa foram incluídos 592 estudos. Depois da leitura do conteúdo completo dos artigos foram selecionados 92 estudos primários. Em uma última etapa, uma avaliação da qualidade de cada artigo foi conduzida. Após esta etapa restaram 89 estudos listados no Anexo A.

4.2 Resultados da Revisão Sistemática da Literatura

Nesta seção são mostrados os resultados obtidos após a condução da revisão sistemática da literatura. Além de responder às questões de pesquisa mencionadas na seção 4.2.1, são mostrados dados demográficos relacionados a quantidade de estudos, abordagem de pesquisa utilizada, canais de publicação mais populares e ecossistemas de software mais pesquisados pelos estudos empíricos. Também é realizada uma breve comparação da revisão realizada com outras revisões e mapeamentos relacionados com ecossistemas de software.

4.2.1 Visão geral dos estudos

89 estudos foram identificados na revisão sistemática da literatura, conforme listado no Anexo A. A Figura 9 mostra a distribuição de artigos publicados em cada ano. A lista final incluiu artigos publicados entre 1992 e 2016. Um número crescente de

estudos foi publicado nos últimos 15 anos, o período máximo de publicação é de 2012 a 2015 (49 artigos, 55%). Percebe-se que em 2016 foram incluídos apenas 4 estudos. Isto ocorreu porque processo de busca foi concluído em janeiro de 2016.

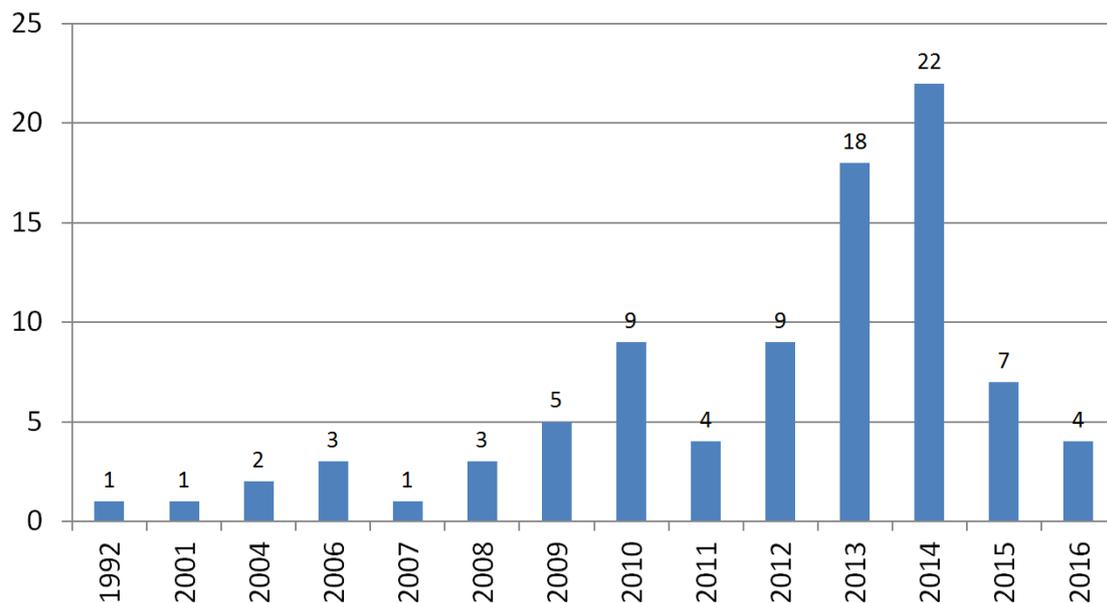


Figura 9: Número de estudos por ano

A Figura 10 mostra que os canais de publicação mais populares são ECSA (8 artigos, 8,9%), IWSECO de software (7 artigos, 7,8%), IST (6 artigos, 7%), JSS (6 artigos, 6,7%). Em particular os eventos são dedicados para área da área de ecossistemas de software: IWSECO de software, MEDES, EWSECO, WEA e DEST. Estes resultados confirmam que o campo do ecossistema de software vem recebendo crescente atenção das comunidades acadêmicas de sistemas de informação, *software business* e engenharia de software.

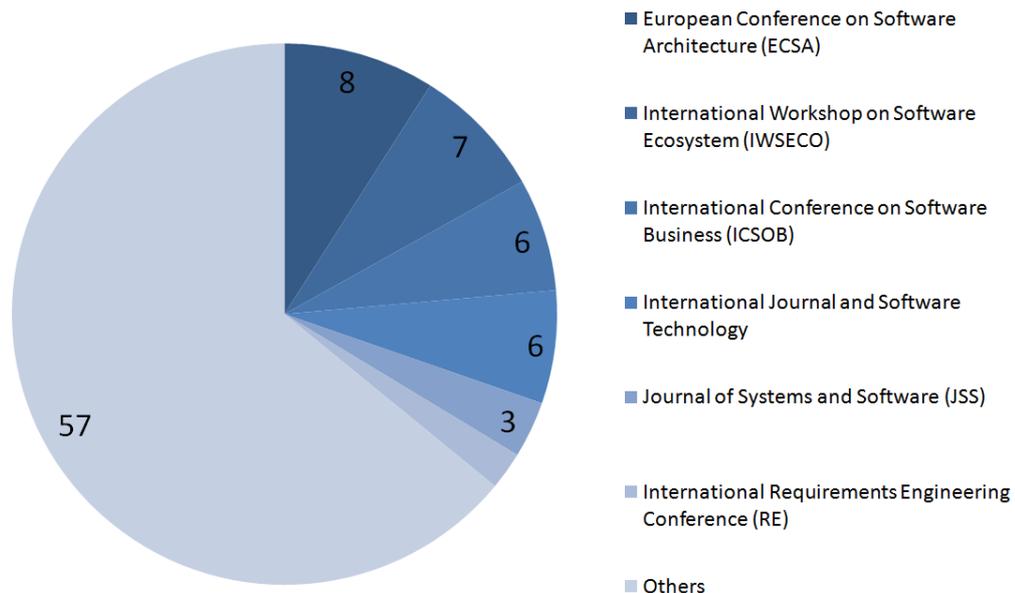


Figura 10: Canais de publicação mais populares

A Figura 11 apresenta os principais ecossistemas de software investigados pelos estudos. Apple e Android são os ecossistemas mais frequentemente examinados (ambos com 4 estudos) seguidos por Siemens, IBM e GX Software (com três estudos para cada ecossistema). Foi observado que ecossistemas *open source* e proprietários foram igualmente analisados. Isto indica que os pesquisadores estão se concentrando nas questões gerenciais específicas enfrentadas por ambos os tipos de ecossistema. A diversidade dos ecossistemas de softwares investigados reforça a ampla perspectiva que a comunidade desta área está ganhando em relação às diferenças entre os ecossistemas *open source* e proprietários.

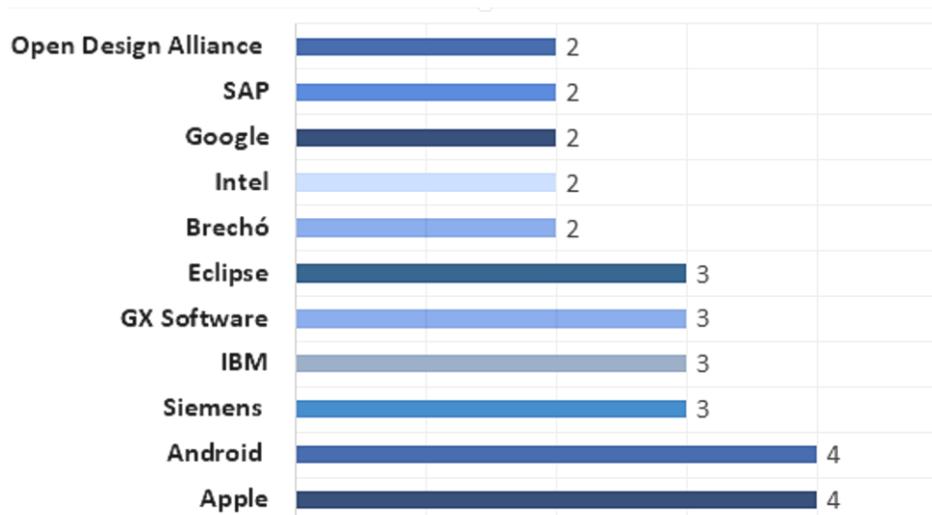


Figura 11: Ecossistemas de software investigados pelos estudos incluídos

A Figura 12 mostra o tipo de pesquisa adotado pelos estudos conforme a classificação proposta por Wieringa et al. (2006). Petersen et al. (2015) fornece critérios de decisão úteis sobre como classificar os estudos. De acordo com a classificação, os estudos podem ser categorizados em seis tipos de pesquisa, a saber:

- *Proposta de Solução* - O artigo propõe uma solução nova ou uma melhoria significativa de uma técnica existente sem uma validação completa.
- *Artigo filosófico* - O artigo propõe uma estrutura conceitual e uma nova maneira de ver as coisas.
- *Artigo de Opinião* - O artigo apresenta a opinião do autor sobre algo.
- *Documento de Experiência* - O artigo descreve a experiência prática do autor que é geralmente um profissional da indústria.
- *Pesquisa de Validação* - O artigo descreve uma validação empírica de uma solução feita no laboratório. Exemplos de estudos incluem: estudo de caso, análise matemática, prototipagem, experiências laboratoriais e simulação.
- *Pesquisa de Avaliação* - O artigo apresenta uma avaliação industrial real de uma solução. Inclui: estudo de caso industrial, experiências controladas com profissionais, pesquisa orientada por profissionais, pesquisa-ação e etnografia.

O tipo mais comum de estudo identificado na revisão é a pesquisa de avaliação (53 estudos, 59%). Este resultado sugere que a maioria dos estudos exploram o campo

sob uma perspectiva prática através da realização de estudos empíricos de ecossistemas reais. 21 estudos são classificados como artigos filosóficos. 11 estudos propõem uma solução como novas técnicas, modelos e métodos. Dois estudos apresentam pesquisa de validação. Finalmente, foram identificados dois artigos de experiência. Não foi encontrado nenhum documento de opinião na lista de estudos primários.

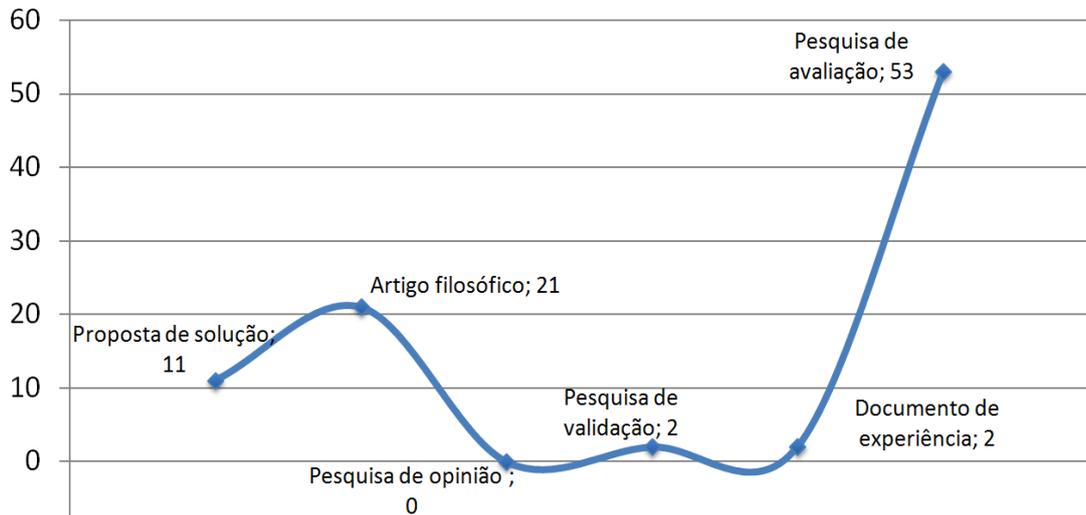


Figura 12: Abordagem de pesquisa adotada pelos estudos incluídos

4.2.2 QP1.1 - Como a governança de ecossistemas de software é definida?

Os resultados mostram que o conceito de governança está ganhando importância na literatura de ecossistemas de software. 9 estudos [S1, S4, S20, S21, S28, S32, S51, S54, S62] definem explicitamente o que é governança de ecossistema de software. Jansen [S34] propõe que a governança é um dos principais domínios do Modelo Corporativo Aberto. O estudo adota a definição de governança dada por Dubinsky e Kruchten [S12], que considera a governança como “a maneira que uma organização é gerida, incluindo os seus poderes, responsabilidades e processos de decisão”. Segundo Jansen et al. [S32], também envolve a atribuição de papéis e direitos de decisão, medidas e políticas. Uma decisão fundamental de governança que os *keystones* devem tomar é quanto poder é dado à comunidade e quanto controle ela mantém para si mesma. Jansen e Cusmano [28] e van Angeren et al. [S54] consideram que a governança do ecossistema “envolve o uso de procedimentos e processos estratégicos para controlar, manter ou mudar o ecossistema”. O Estudo [S54] também afirma que a governança de ecossistemas de software “engloba aspectos técnicos e gerenciais, incluindo o gerenciamento da plataforma de software e

suas interfaces, a definição de modelos de parcerias e o estabelecimento de barreiras à entrada”.

Baars e Jansen [S4] propõem uma estrutura para a governança de ecossistemas de software e definem a governança de ecossistemas de software como: “procedimentos e processos pelos quais uma empresa controla, altera ou mantém sua posição atual e futura em um ecossistema de software em todos os diferentes níveis de escopo”. Estudos [S1], [S28] e [S62] adotam a mesma definição. Albert et al [S1] apresentam uma abordagem de governança de ecossistema de software para permitir o controle da arquitetura de TI com base no gerenciamento de ativos de software. Em [S28], Jansen e Cusumano propõem um modelo de governança para a preservação e melhoria da saúde do ecossistema. Wnuk et al., [S62] avaliam o modelo proposto em [S28] por meio de um estudo de caso em um ecossistema de software dependente de hardware.

Foi observado que vários estudos primários discutem a tensão clássica entre modelos de governança abertos e fechados. Jansen et al., [S32] sugerem que as empresas se beneficiam da abertura de seus modelos de negócios. Isto inclui o compartilhamento de conhecimento estratégico, explicitando a estratégia do ecossistema para todos os atores e coordenando ações. Os autores propõem um grau de abertura para avaliar o quão aberta uma organização é. Em [S13], os autores não definem o que é governança no contexto de ecossistemas, mas fornecem uma rica discussão sobre as tensões entre modelos de governança abertos e fechados à medida que as plataformas amadurecem. O estudo propõe que os modelos de governança híbridos são mais adequados para plataformas proprietárias e compartilhadas. Esse modelo é caracterizado pelo controle centralizado sobre a tecnologia de plataforma e responsabilidades compartilhadas pela comunidade do ecossistema.

Ghazawne et al., [S20] argumentam que a governança dos ecossistemas de software envolve “um ato de equilíbrio delicado do *keystone* tentando manter o controle da plataforma, ao mesmo tempo que procura expandir a diversidade de potenciais desenvolvedores”. Segundo Tiwana et al. [S51], a governança geralmente se refere a “quem decide o que em um ecossistema”. O estudo [S51] investiga a evolução dos ecossistemas centrados em plataformas e propõe que a governança pode ser analisada a partir de três facetas: i) como os direitos de decisão são divididos entre o proprietário da

plataforma e os desenvolvedores de aplicativos, ii) quais tipos de mecanismos de controle são usados pelo proprietário da plataforma e iii) como a propriedade é regulamentada se a plataforma for propriedade de uma única empresa ou compartilhada por vários proprietários. Tiwana et al [S51] também afirma que a governança do ecossistema “envolve compartilhamento de responsabilidades e autoridade, alinhamento de incentivos e compartilhamento de interesses”. Goldbach e Kemper [S21] adotam a mesma definição de governança de plataforma dada pelo estudo [S51] para entender como os mecanismos de controle impostos pelo proprietário da plataforma afetam todo ecossistema de software. Todos os estudos primários que respondem a essa questão de pesquisa sugerem que um dos principais desafios enfrentados pelos proprietários de plataformas é equilibrar seus próprios objetivos estratégicos com as metas e atividades dos participantes do ecossistema de software. Esse delicado equilíbrio é crítico para que os ecossistemas de software prosperem.

7 estudos [S10, S28, S32, S41, S54, S55, S62] indicam que a governança do ecossistema influencia a saúde e a sustentabilidade dos ecossistemas. Isso significa que as estratégias de governança e as decisões administrativas tomadas pelos *keystones* afetarão a evolução saudável de todo o ecossistema. Os principais estudos sugerem que as métricas de saúde fornecem indicadores operacionais sobre como os ecossistemas de software são governados. Por exemplo, se um modelo de governança aberto for adotado pelo ecossistema, mais autonomia será dada aos atores para atuar no seu crescimento e expansão. Caso contrário, em um modelo de governança fechado o *keystone* detém poder e controle substanciais sobre os atores. Consequentemente, o *keystone* tem mais responsabilidade em relação à prosperidade e à saúde geral do ecossistema. Definir a estratégia de abertura é uma decisão importante que os orquestradores devem tomar ao estruturar o modelo de governança para seus ecossistemas de software. Esta decisão terá um impacto significativo na evolução da arquitetura corporativa de sistemas integrados. A seguinte definição integrada para governança de ecossistemas de software é proposta concluindo esta seção:

Governança consiste em todos os processos pelos quais um ator cria valor, coordena relacionamentos e define níveis de controle.

4.2.3 QP1.2 - Quais são os mecanismos propostos pela literatura para governar ecossistemas de software?

Conforme discutido na seção anterior, apenas 9 estudos definem formalmente o que constitui a governança de ecossistemas de software. No entanto, um número substancial de estudos primários descreve mecanismos para governar os ecossistemas de software (Tabela 3). Mecanismos de governança de ecossistema de software são definidos nesta pesquisa como ferramentas gerenciais utilizadas pelos atores com o objetivo de influenciar a saúde de um ecossistema. Foi observado que frequentemente os autores usam termos como "orquestração" e "gestão" para se referir ao que pode ser entendido como um mecanismo de governança. Para classificar os 89 estudos, foram propostas três categorias principais de mecanismos de governança:

- **Criação de Valor:** envolve mecanismos para gerar e distribuir valor para todo o ecossistema. Mecanismos de criação de valor são geralmente propostos e nutridos pelo *keystone*, que deve entender como criar valor que é apreciado por desenvolvedores e clientes. Neste contexto, é importante identificar fontes de valor (tais como licenças e modelos de receitas) e estimular a co-criação de valor entre os atores, por meio da inovação, da rede colaborativa e da partilha de custos. Como resultado, o ecossistema pode atrair e reter parceiros que se beneficiarão mutuamente do valor distribuído dentro do ecossistema de software. Esta categoria aborda todos os incentivos e benefícios que os atores podem ganhar no ecossistema de software.
- **Coordenação de atores:** descreve mecanismos para manter a consistência e integração de atividades, relacionamentos e estruturas do ecossistema para parceiros e clientes. Como resultado, os participantes do ecossistema podem trabalhar de forma harmoniosa e eficaz. Nesta dimensão foram identificados mecanismos para estimular modelos de parceria, definir papéis e responsabilidades para os atores e melhorar os canais de comunicação dentro do ecossistema e nutrir colaboração. Além disso, os estudos primários propõem mecanismos para gerenciar questões críticas, tais como: conflitos, recursos, riscos e expectativas.

- **Controle e abertura organizacional:** captura a notória tensão entre estratégias abertas versus fechadas como um desafio de governança central. Mecanismos desta dimensão representam quanto controle será retido pelo *keystone* para garantir sua posição de poder e quanta autonomia será dada para os atores tomarem suas próprias decisões de forma independente. De um lado, os *keystones* podem suportar autonomia, distribuir poder e compartilhar conhecimento sobre *roadmaps* tecnológicos e decisões arquiteturais. Por outro lado, os *keystones* podem manter o controle definindo requisitos de entrada, estabelecendo padrões de qualidade e certificações.

Tabela 3 mostra a classificação dos mecanismos de governança propostos pelos estudos primários. Foram utilizadas as categorias apresentadas anteriormente (criação de valor, coordenação dos atores e controle e abertura organizacional) para classificar os mecanismos. Foi observado que os mecanismos mais citados são: atrair e manter parceiros (28 artigos, 31%), compartilhar conhecimento (20 artigos, 22%) e promover inovação (25 artigos, 28%). Não é possível afirmar que esses sejam os mecanismos de governança mais importantes, pois vários estudos sugerem que a governança deve corresponder ao contexto específico e as orientações de mercado envolvidos no ecossistema [S4, S13, S25, S32, S43].

Tabela 3: Mecanismos de governança em ecossistemas de software

Mecanismos de Governança		Estudos	Número de estudos
Criação de valor	Promover inovação	S61, S7, S32, S50, S40, S48, S52, S9, S47, S3, S7, S45, S10, S8, S35, S27, S18, S17, S19, S61, S38, S57, S24, S43, S44	25
	Gerenciar licenças	S16, S32, S41, S40, S1, S3, S6, S58, S28, S2, S51, S63, S8, S27, S18, S13, S17, S57, S31, S22, S24	21
	Criar modelos de receita	S7, S3, S45, S58, S4, S5, S7S6, S28, S10, S62, S30, S23, S27, S61, S38, S53, S57, S23, S36, S39	20
	Atrair e manter variados parceiros	S61, S32, S29, S52, S47, S45, S15, S58, S4, S6, S10, S62, S55, S46, S63, S35, S27, S18, S17, S61, S38, S53, S57, S42, S19, S23, S26, S36	28
	Estimular investimento e compartilhar custos	S61, S56, S3, S45, S8, S27, S22, S23, S43	9
Coordenação de atores	Criar modelos de parceria	S32, S56, S54, S4, 28, S62, S55, S49, S30, S8, S27, S53, S31, S19, S24	15
	Definir regras para gerenciar relacionamentos	S32, S40, S29, S56, S52, S9, S3, S4, S5, S46, S2, S63, S35, S27, S57, S42, S36	17
	Estabelecer regras e responsabilidades	S41, S50, S40, S56, S3, S15, S4, S5, S49, S46, S51, S63, S27, S13, S42, S26, S37	17
	Estabelecer efetivos canais de comunicação	S41, S29, S48, S52, S9, S3, S11, S14, 28, S16, S27, S31, S37	13
	Gerenciar conflitos	S32, S52, S15, S8, S27, S57, S31, S42, S19	9
	Gerenciar recursos	S1, S52, S9, S47, S3, S15, S10, S46, S20, S35, S42, S26, S36, S44	14
	Gerenciar riscos	S50, S40, S56, S52, S58, S46, S30, S8, S18, S17, S57, S22, S39, S43	14
	Gerenciar expectativas	S47, S49, S16	3
	Nutrir colaboração	S61, S50, S52, S46, S58, 28, S62, S55, S49, S46, S35, S17, S42, S44	14
Controle e abertura organizacional	Apoiar autonomia	S7, S50, S52, S3, S48, S4, S7, S46, S20, S51, S35, S18, S17, S61, S42	15
	Compartilhar informação	S16, S32, S50, S40, S29, S48, S52, S3, S4, S11, S62, S30, S20, S35, S18, S17, S61, S57, S31, S37	20
	Distribuir poder	S32, S50, S52, S3, S15, S46, S16, S51, S27, S37	10
	Definir barreiras de entrada	S54, S45, S4, S28, S62, S30, S18, S38, S53, S24, S36	11
	Compartilhar decisões arquiteturais	S16, S29, S48, S1, S52, S9, S47, S3, S5, S58, S28, S62, S2, S51, S27, S11, S14	17
	Definir padrões e certificações de qualidade	S32, S41, S50, S40, S56, S58, S28, S62, S55, S30, S38, S57, S22	13

4.2.4 QP1.3 - Como a saúde é caracterizada pela literatura?

Saúde é um termo originalmente formulado no campo dos ecossistemas naturais. Costanza [S68] define um ecossistema natural saudável como “estável e sustentável, mantendo sua organização e autonomia ao longo do tempo e sua resiliência ao estresse”. Den Hartigh et al [S10] e Iansiti e Levien [S25] se utilizaram dos conceitos de saúde de ecossistemas naturais para definir o termo ‘saúde’ em ecossistemas de negócio. A caracterização de ‘saúde’ de foi inicialmente proposta por Iansiti e Levien [S25], como uma maneira de medir o desempenho dos ecossistemas de negócios. Os autores emprestaram analogias da biologia para explicar que, da mesma forma que na natureza, a saúde de ecossistemas de negócios depende do relacionamento e da dependência mútua entre os atores do ecossistema. Isso significa que cada ator pode favorecer ou prejudicar a saúde de todo o ecossistema.

Como sugerido por Hansen et al. [18], os ecossistemas de software podem ser considerados um tipo de ecossistema de negócio em que uma plataforma tecnológica intermedeia interações entre os atores. Foi observado que os estudos primários frequentemente adotam as mesmas definições do domínio de ecossistemas de negócios para medir a saúde do software ecossistemas. Em geral, os proprietários de plataformas têm uma forte responsabilidade de sustentar o desempenho do ecossistema. Como Iansiti e Levien [S25] apontam, os proprietários de plataformas não promovem a prosperidade dos outros por razões altruístas. Eles fazem isso porque é uma estratégia muito benéfica para ambos os lados (*keystones* e parceiros). Fotrousi et al. [28] afirmam que para criar valor em um ecossistema de software o proprietário de uma plataforma precisa garantir que o ecossistema seja saudável e sustentável. O desempenho geral dos ecossistemas de software depende das ações e decisões tomadas por cada ator de forma individual. Um ecossistema é saudável quando fornece benefícios mútuos para os atores [S83].

A sustentabilidade é um conceito fortemente relacionado à saúde do ecossistema. Dhungana et al. [S70] define um ecossistema de software sustentável como aquele que “pode aumentar ou manter sua comunidade de usuários / desenvolvedores por longos períodos de tempo e pode sobreviver a mudanças inerentes como novas tecnologias ou novos produtos (por exemplo, concorrentes) que podem mudar a população (comunidade

de usuários, desenvolvedores, etc.) ou ataques/sabotagem significativos à plataforma do ecossistema”. Para analisar como autores no campo de ecossistemas de software entendem o termo saúde, a Tabela 4 apresenta diferentes definições de saúde para ecossistemas de software propostas na literatura.

A partir da Tabela 4, foi sintetizada seguinte definição: *os ecossistemas são saudáveis quando exibem longevidade e propensão ao crescimento. A saúde do ecossistema de software é determinada pela capacidade de um ecossistema produzir persistentemente resultados significativos (produtividade), sobreviver a rupturas do mercado (robustez) e criar nichos no ecossistema (criação de nicho).*

Tabela 4: Definições para saúde de ecossistemas de software encontradas nos estudos

Definição	Estudos	Número de estudos
1. Pode ser mensurada como produtividade, robustez e criação de nicho.	S41, S64, S65, S82, S54, S71, S44, S28, S10, S62, S55, S47, S30, S21, S25, S76, S89, S77, S83, S66, S57	22
2. Longevidade e prosperidade para crescer.	S65, S54, S28, S10, S21, S34, S80, S66, S69	9
3. Pode ser dividido em três componentes: saúde do componente de software, saúde da plataforma de software, e saúde do ecossistema de software.	S41, S65	2
4. A habilidade de o ecossistema perdurar permanecendo variável e produtivo ao longo do tempo	S82, S83, S66, S68	4
5. Provê informação estratégica sobre o estado e a trajetória do ecossistema.	S41, S10, S25	3
6. É influenciada por três forças: atores, software e orquestração.	S41, S82, S83	3
7. É principalmente definida como a habilidade para prover crescimento de oportunidades duráveis para os membros do ecossistema e para quem depende dele.	S77, S83	2
8. É uma medida que caracteriza a performance de todo ecossistema.	S77, S57, S69	3

4.2.4 QP1.4 - Quais métricas para medir a saúde de ecossistemas de software foram propostas pela literatura?

Medir a saúde dos ecossistemas de software informa como o ecossistema está evoluindo e como os mecanismos de governança estão contribuindo para a futura sustentabilidade de atores individuais, assim como de todo o ecossistema. Portanto, é possível argumentar que fornecer um extenso catálogo de métricas de saúde ajuda os atores a selecionar as melhores estratégias para sobreviver e obter benefícios do ecossistema.

Um ecossistema saudável atrai e retém atores que podem criar valor e inovações para a plataforma [S54, S38]. Medir a saúde pode esclarecer o funcionamento e a evolução prevendo mudanças e estados futuros de um ecossistema [S82]. Mhamdia [S66] amplia as três dimensões da saúde de Iansiti e Levien, introduzindo as dimensões de criatividade e de satisfação dos stakeholders. O estudo de Mhamdia propõe um modelo conceitual para medir a saúde dos ecossistemas de software. Em uma SLR recente, Manikas [16] concluiu que a governança bem-sucedida do ecossistema de software deve fazer uso adequado dos recursos do ecossistema, aumentar a produtividade, apoiar a robustez e promover a saúde do ecossistema. Em [S69], den Hartigh sugere que a mensuração da saúde pode ser usada como um instrumento de governança do ecossistema. Essas observações mostram a relevância de selecionar métricas de saúde adequadas, porque elas são medidas operacionais da governança de ecossistemas de software.

Um grande número de métricas para avaliar a saúde dos ecossistemas de software tem sido proposto na literatura. No entanto, estudos anteriores não fornecem um catálogo abrangente destas métricas. A Tabela 5 apresenta uma classificação das métricas de saúde propostas pelos estudos primários. Na revisão sistemática, 52 métricas foram identificadas e classificadas nos três elementos de saúde (produtividade, robustez, criação de nicho). Adotando análise temática para sintetizar as evidências, as métricas foram identificadas nos estudos primários em grupos com base em seu significado similar. Por exemplo, as métricas ‘número de direitos autorais’ e ‘número de *trademarks*’ propostas por Huang et al. [S24] e Ceccagnoli et al. [S8] foram integradas dentro da métrica ‘número de patentes’. Outras variáveis, como o ‘número de novos desenvolvedores por mês’ [S80], a ‘porcentagem de desenvolvedores com status de parceiro’ [S54] foram

classificadas em uma métrica mais genérica chamada de ‘colaboradores/desenvolvedores ativos’.

Em alguns estudos, os autores não mencionam explicitamente o termo métricas para avaliar a saúde dos ecossistemas, mas, após uma análise e interpretação cuidadosa, consideraram-se alguns termos como tipos de métrica. Por exemplo, West e Wood [S61] usam o termo “fatores de sucesso” para apresentar termos que outros estudos claramente classificam como métricas de saúde. Outro exemplo é Mizushima et al [S40] que apresentam um conjunto de mecanismos para formar uma comunidade OSS bem-sucedida, mas esses mecanismos são notoriamente tipos de métricas por serem quantificáveis: ‘eventos para desenvolvedores’, ‘compartilhamento de conhecimento’. Em resumo, o procedimento para sintetizar evidências consistiu em começar a preencher a tabela com métricas que foram explicitamente citadas como métricas de saúde pelos estudos primários. Em seguida, foram incluídos dados que estavam intimamente relacionados às métricas de saúde identificadas inicialmente, mas que os autores não os descrevem como tal. Para evitar uma alta granularidade de informações, foram listadas apenas métricas mencionadas por pelo menos dois estudos. Em essência, nenhuma métrica pode cobrir todos os aspectos constitutivos da saúde do ecossistema de software. É necessário selecionar uma combinação de medidas para avaliar a saúde e o desempenho resultante do ecossistema.

Também foram classificados estudos que propõem métricas de saúde para ecossistemas abertos e proprietários, como descrito na Tabela 6. Foi identificado que 24 estudos (26%) apresentam métricas para avaliar a saúde dos ecossistemas proprietários e 25 estudos (28%) que propõe métricas de saúde para ecossistemas de código aberto. Este resultado mostra um bom equilíbrio de pesquisas sobre a saúde de ecossistemas proprietários e abertos. No entanto, como destacado por Alami et al. [S65], os ecossistemas podem não divulgar dados relevantes relacionados a algumas métricas. Por exemplo, os dados relativos aos clientes que abandonam a plataforma podem ser considerados um sinal de um ecossistema insalubre. De fato, este tipo de dado estratégico e sensível pode não ser facilmente obtido. Esta situação é um desafio para investigar e estabelecer comparações em relação à saúde dos ecossistemas. Dados sobre ecossistemas abertos podem ser adquiridos usando técnicas de mineração de repositórios. Isso pode ser uma vantagem ao se investigar este tipo de ecossistema.

Os orquestradores de ecossistemas de software podem usar informações que derivam das métricas de saúde para tomar decisões bem informadas sobre sua estratégia. Por exemplo, a mecanismo de governança ‘atrair e manter variados parceiros’ pode ser operacionalizado em termos das métricas de saúde ‘construção de comunidade/modelo de parceria, a satisfação dos stakeholders/contribuintes, nível de proximidade percebida/apoio do orquestrador. Ao avaliar problemas relacionados a estas métricas, os atores podem tomar decisões sobre como melhorar a atração e manutenção de variados parceiros. Uma contribuição chave da QP1.3 é a Tabela 5. Ela fornece um extenso catálogo de métricas de saúde que pode ser um guia útil para que os orquestradores governem e gerenciem com sucesso os ecossistemas de software. Por conseguinte, selecionando métricas de saúde, os atores podem direcionar o ecossistema para um caminho sustentável.

Tabela 5: Métricas associadas à saúde de ecossistemas de software

Métricas para avaliar a saúde		
Produtividade	Estudos	Número de estudos
Fator total de produtividade	S10, S34, S77, S83, S69	5
Melhoria da produtividade	S84, S73, S10, S34, S76, S77, S83	7
Novos projetos relacionados	S74, S76, S34, S80	4
Eventos para desenvolvedores	S40, S56, S71, S3, S28, S62, S30, S34	8
Número e tamanho de commits	S84, S70, S85, S64, S71, S83	6
Técnicas de orquestração	S41, S29, S58, S83	4
Frequência de releases	S41, S40, S65, S29, S71, S89, S80	7
Linha de código	S41, S64, S73, S71, S78	5
Aumento de clientes/usuários	S61, S67, S65, S83	4
Abandono de clientes	S65, S4	2
Tamanho e efetividade da rede	S82, S48, S10, S87, S38, S88	6
Contribuidores/desenvolvedores ativos	S67, S64, S65, S5, S73, S74, S4 S30, S81, S87, S89, S80, S38, S83 S19 S79 S69 S88	18
Número de Apps/projetos/extensões	S65, S74, S71, S89, S80, S38	6
Número de patentes	S10, S24, S8, S66, S69	5
Tecnologias/ inovações introduzidas	S61, S32, S40, S10, S34, S76, S77, S83, S66	9
Responsividade	S34, S69	3
Retorno do capital investido	S25	1
Robustez	Estudos	Número de estudos
Taxa de sobrevivência	S10, S30, S34, S25, S76, S77, S83	7
Persistência da estrutura	S48, S10, S34, S77, S83	5
Predicabilidade	S10, S34, S77, S83	5
Obsolência limitada	S10, S34, S77, S83	4
Continuidade/ estabilidade da rede	S85, S72, S40, S48, S10, S30 S34, S83	8
Qualidade de artefatos e modelos de certificação	S32, S72, S67, S50, S65, S29, S54, S30, S34, S38, S83	11
Consistência do núcleo da rede	S74, S34	2
Outbound links to other ecosystems	S85, S32, S34, S80	4
Mudança de custos	S75, S10, S30, S34	4
Marketing e vendas	S84, S70, S32, S40, S47, S28, S62	7

Número de downloads	S8,4 S74, S4, S78, S79, S69	6
Construção da comunidade/ modelo de parceria	S84, S32, S67, S64, S29, S54, S47, S4, S28, S62, S30, S81, S87	13
Adaptabilidade	S70, S72, S41, S29, S28, S83	6
Conectividade	S85, S82, S48, S54, S74, S37, S10, S55, S30, S26, S34, S87, S83, S66, S69	15
Densidade	S54, S55, S26, S69	4
Centralização	S82, S10, S55, S30, S89, S78, S69, S88	8
<i>Market share</i>	S65, S10, S34, S89, S38, S69	6
<i>Platform findability</i>	S32, S65, S89, S69	4
<i>Platform stickiness</i>	S61, S65, S75, S34	4
Número de parceiros	S85, S67, S64, S40, S54, S4, S28 S10, S55, S26, S25, S81, S87, S79, S69	15
Satisfação do Stakeholder/ Colaborador	S72, S40, S47, S34, S77, S38, S83 S66	8
Crescimento do lucro	S41, S40, S28, S10, S62, S66	6
Aumento de receita	S28, S10, S66	3
Criação de nicho	Estudos	Número de estudos
Criação de valor e inovações	S61, S84, S70, S40, S47, S71, S75, S28, S10, S8, S34, S25, S76, S18, S77, S83, S66	17
Variedade	S40, S65, S10, S25, S76, S86 S77, S83	8
Número de novos projetos	S70, S10, S34	3
Visibilidade no Mercado/ Reputação	S84, S70, S32, S64, S40, S56, S47, S10, S34, S89, S38	3
Barreiras de entrada	S64, S40, S54, S28, S62, S24, S8, S18, S38, S78	10
Nível de transparência/abertura	S70, S16, S32, S41, S50, S40, S29, S3, S36, S28, S62, S30, S24, S8, S86, S54, S86, S38, S78, S33	20
Média do número de linguagens suportadas	S65, S80, S38	3
Modularidade (Criação de subcomunidades)	S40, S28, S87, S86, S80, S77, S79, S88	8
Número de markets	S36, S34, S38	3
Nível d eintimidade percebida/ Suporte do orquestrador	S67, S64, S40, S29, S47, S34, S38	7
Crescimento de vendas	S40, S24, S8	3
Reciprocidade	S86, S79	2

Tabela 6: Tipos de ecossistemas para os quais métricas de saúde foram propostas

Tipo de ecossistema	Estudos	Número de estudos
Ecossistemas abertos	S84, S70, S16, S85, S32, S67, S64, S40, S65, S82, S73, S74, S74, S71, S4, S28, S34, S81, S87, S89, S80, S78, S79, S69, S88	25
Ecossistemas proprietários	S84, S16, S32, S41, S54, S3, S75, S36, S5 S28, S10, S55, S30, S24, S8, S26, S25, S76, S18, S86, S77, S38, S66, S33	24

4.3 Ameaças à validade

A condução da SLR sofreu ameaças à validade semelhante a qualquer outra revisão sistemática da literatura. Uma das limitações principais em uma revisão é o viés em procedimentos de seleção e de extração de dados (Kitchenham e Charters, 2007). Ecossistemas de software é um campo multidisciplinar que abrange estudos de engenharia de software, sistemas de informação e gestão do conhecimento. Para limitar a ameaça de não incluir estudos primários relevantes, foi adotada uma estratégia de busca com palavras-chave genéricas para recuperar o maior número possível de artigos relacionados ao tópico da pesquisa. A busca automática foi complementada com buscas manuais nas principais revistas, conferências e workshops onde estudos em ecossistemas de software foram publicados. Além disso, também foram analisados estudos primários de outras revisões da literatura publicadas na área. Dada a familiaridade dos envolvidos na SLR com o tema de pesquisa, antes de realizar este estudo, já se tinha conhecimento de uma série de estudos relevantes que cobrem a governança e a saúde de ecossistemas. Estes estudos foram analisados cuidadosamente e, após reuniões de discussão entre todos os pesquisadores envolvidos, foi decidido que esses estudos deveriam ser incluídos porque eles são fonte importante de conhecimento para responder às questões de pesquisa.

Para mitigar o impacto do viés de seleção, foi definido um protocolo de revisão com critérios claros de inclusão e exclusão para cada etapa de seleção. Na primeira etapa de seleção, um grande número de estudos irrelevantes foi removido ao se analisar o título e o resumo. Um pesquisador realizou essa tarefa. Na segunda etapa de seleção, dois pesquisadores examinaram o conteúdo dos estudos e verificaram a consistência dos resultados da seleção preliminar. Os estudos primários também foram analisados quanto ao atendimento a critérios de qualidade.

Com relação ao viés na extração de dados, houve alguns problemas para extrair informações relevantes de estudos primários. Esse problema foi mais crítico para responder a RQ1.2. Foi observado que os estudos utilizam terminologias diferentes para descrever aspectos relacionados a mecanismos de governança. Em várias ocasiões, foi necessário interpretar as informações subjetivas fornecidas pelos artigos. Para minimizar o viés de interpretação, foi realizada uma leitura cuidadosa e foram realizadas várias reuniões de discussão entre os pesquisadores durante a fase de extração de dados.

4.4 Comparação com outras revisões da literatura

A área de ecossistemas de software recebeu uma série de revisões da literatura nos últimos anos. Foram identificadas quatro revisões sistemáticas de literatura e cinco estudos de mapeamento. Alguns estudos cobrem a área de ecossistemas de software em geral, como (Barbosa e Alves, 2011), (Barbosa et al., 2013) (Manikas e Hansen, 2013) e (Manikas, 2016). Outras revisões focam aspectos distintos de um ecossistema de software como garantia da qualidade (Axelsson e Skoglund, 2015), modelos de qualidade (Franco-Bedoya et al., 2014), saúde (Hyrnsalmi et al., 2015), indicadores de desempenho (Fotrousi et al., 2014) e inovação aberta (Papatheocharous et al., 2015).

Barbosa e Alves (2011) publicaram o primeiro estudo de mapeamento em ecossistemas de software. A pesquisa analisou 44 estudos primários e os resultados trouxeram uma perspectiva geral da área identificando benefícios, desafios e os principais temas de investigação em ecossistemas de software. Em Barbosa et al. (2013), os autores estenderam a revisão analisando ecossistemas de software sob uma perspectiva tridimensional: técnica, empresarial e social.

Manikas e Hansen (2013) publicaram uma revisão sistemática da literatura cobrindo todo o campo de ecossistemas de software. Os autores analisaram 90 estudos primários e abordaram questões gerais de pesquisa. As principais contribuições incluíram uma comparação de definições de ecossistemas de software e uma descrição dos atores mais comuns encontrados na literatura. Na revisão, que cobriu estudos entre 2007-2012, os autores reconheceram que a área ainda era imatura devido à falta de modelos analíticos e de estudos industriais.

Em Manikas (2016), o autor atualizou a revisão que englobou 231 estudos publicados até 2014. Este estudo é a revisão mais recente que analisa a área de ecossistemas de software em sua totalidade. Manikas (2016) identificou que dos estudos primários incluídos que investigaram um ecossistema real, quase metade analisou um ecossistema proprietário. Os ecossistemas proprietários mais populares identificados em Manikas (2016) são Apple e Microsoft. Esse resultado é semelhante ao nosso, conforme apresentado na Figura 8. Com relação aos ecossistemas *open source*, os autores encontraram resultados semelhantes, com Eclipse e Android sendo os ecossistemas mais populares analisados pelos estudos primários de ambas as SLR. Comparando a figura 9.

com os resultados de Manikas (2016), foram identificadas várias similaridades referente aos canais de publicação mais populares no campo dos ecossistemas de software.

Um resultado importante de Manikas (2016) é a confirmação de que a área obteve um rápido aumento no número de publicações e um número crescente de estudos empíricos. Estes resultados podem ser interpretados como um sinal de maturidade da área. Na Figura 8, percebe-se resultado semelhante ao elevado número de estudos primários relatando uma pesquisa de avaliação (49%), que inclui uma análise industrial real de uma solução. No entanto, Manikas concluiu que a área ainda carece de teorias para explicar fenômenos específicos no contexto de ecossistemas de software. Numa revisão não-sistemática sobre ecossistema de software, Hansen e Dyba (2012) identificaram várias teorias de outros campos que foram adotadas pela comunidade de pesquisa de ecossistemas de software. Ambas as revisões corroboram a importância das teorias para apoiar a maturidade do campo de ecossistemas de software.

Hyrnsalmi et al (2015) conduziu uma SLR com 38 estudos primários para caracterizar a saúde de ecossistemas de software. Os autores listaram como os estudos primários definem a saúde de ecossistemas de software e analisaram os fatores que afetam a sustentabilidade de um ecossistema de software. A revisão realizada nesta tese se difere da revisão realizada por Hyrnsalmi et al (2015) pelo fato de o segundo apenas listar definições de saúde em ecossistemas de software sem discuti-las ou propor uma definição própria para o termo. Hyrnsalmi et al (2015) apesar de propor uma classificação de saúde baseada nas forças e fatores que a afetam dentro de um ecossistema de software, não indica quais métricas podem ser utilizadas para mensurá-la. Hyrnsalmi et al (2015) também não investiga a ocorrência do termo saúde em estudos que tratam separadamente de ecossistemas abertos e de ecossistemas fechados.

Axelsson e Skoglund (2015) mapearam a literatura de garantia da qualidade em ecossistemas de software. Os autores identificaram apenas seis estudos primários. Os achados foram bastante superficiais e a análise dos estudos foi muito descritiva. Os estudos primários foram simplesmente resumidos na revisão. A principal contribuição deste estudo de mapeamento é uma agenda para enfrentar os desafios nesta área específica de pesquisa de ecossistema de software.

Franco-Bedoya et al. (2014) realizaram uma SLR superficial, com base em 17 estudos primários, com o objetivo de identificar medidas para avaliar a qualidade dos ecossistemas *open source*. A principal contribuição da revisão é a proposta de um modelo de qualidade denominado QuESO que organiza medidas encontradas na literatura em diversas classificações de qualidade. O modelo aborda três aspectos de ecossistemas abertos: plataforma, comunidade e rede.

Fortrousi et al (2015) executaram um estudo de mapeamento para fornecer uma visão geral sobre o uso de indicadores de desempenho (KPI) para o gerenciamento de ecossistemas de software. O estudo analisou 34 artigos. Os autores identificaram que os objetivos do ecossistema apoiados por KPI abrangem melhorias nos negócios, crescimento, estabilidade, melhoria da qualidade e sustentabilidade. Uma contribuição chave da revisão é a classificação dos atributos de medição.

Finalmente, Papatheocharous et al. (2015) realizaram um estudo de mapeamento de ecossistemas e inovação aberta para sistemas embarcados, nos quais foram identificados 260 estudos. Uma limitação importante do estudo é a falta de uma lista completa dos estudos primários. Este tipo de informação é essencial para que outros pesquisadores possam comparar estudos secundários e avaliar a confiabilidade da revisão realizada. As questões de pesquisa são muito genéricas. De fato, a revisão não fornece uma discussão clara sobre as principais sobreposições entre ecossistemas e áreas de inovação aberta.

A revisão realizada na presente pesquisa difere dos estudos secundários descritos anteriormente por abordar uma área específica de pesquisa em ecossistema de software, que é a governança de ecossistemas de software. Na SLR é fornecida uma análise aprofundada da área, caracterizando o que significa a governança de ecossistemas (RQ1), e classificando os mecanismos de governança encontrados na literatura (RQ2). Portanto, além de fundamentar a elaboração do modelo conceitual, a SLR traz uma nova contribuição para sintetizar o corpo de conhecimento em governança de ecossistemas de software.

4.5 Síntese do capítulo

Este capítulo apresentou os resultados da Revisão Sistemática da Literatura realizada na fase de investigação do problema do ciclo de Design Science Research (Figura 5). A revisão foi guiada por um conjunto de questões de pesquisa (4.1.1), realizada segundo as orientações de Kitchenham e Charters (2007) e incluiu 89 artigos encontrados através de buscas automáticas e manuais (4.1.2), selecionados por meio de critérios de inclusão e exclusão (4.1.5) previamente definidos. A partir dos resultados concluiu-se: i) que governança consiste em todos os processos pelos quais um ator cria valor, coordena relacionamentos e define níveis de controle; ii) os mecanismos de governança podem ser categorizados em dimensões de criação de valor, coordenação de atores e controle e abertura organizacional; iii) A saúde do ecossistema de software é determinada pela capacidade de um ecossistema produzir persistentemente resultados significativos (produtividade), sobreviver a rupturas do mercado (robustez) e criar nichos no ecossistema (criação de nicho). iv) as métricas associadas a produtividade, robustez e criação de nicho mais mencionadas pela literatura são respectivamente contribuidores/desenvolvedores ativos, número de parceiros e nível de transparência/saúde. O capítulo ainda apresentou ameaças a validade relacionadas com a revisão sistemática da literatura (4.3) e comparou os resultados com os de outras revisões da área (4.4).

5 MODELO CONCEITUAL PARA GOVERNANÇA DE ECOSISTEMAS DE SOFTWARE

Este capítulo relata os resultados da fase de design da solução do ciclo de Design Science de Wieringa (2014) e corresponde ao processo de proposição do modelo conceitual para governança de ecossistemas de software. O paradigma conceitual foi escolhido para a modelagem porque modelos conceituais são baseados em uma abordagem estruturada que pode ser usada para entender áreas de estudo representadas na forma de uma figura ou mapa (Allen e March, 2012). A modelagem conceitual é vista como um processo transformador que apoia a aprendizagem organizacional, reunindo diversas perspectivas e valores para conceituar e representar constructos complexos de maneira clara e sistemática (Sutherland e Katz, 2005). Modelos baseados na abordagem conceitual facilitam o aprendizado sobre conceitos e seus relacionamentos e são utilizados em uma variedade de domínios para esclarecer ideias e organizar o conhecimento (Novak e Canas, 2008).

O modelo de governança proposto possui um caráter conceitual, pois utiliza uma linguagem notacional para fornecer uma representação clara de um conjunto de elementos e das relações existentes entre eles. Os elementos fazem parte da área de conhecimento de governança de ecossistema de software e ajudam a explicar o domínio em questão (Noy e McGuinness, 2001).

O modelo conceitual apoia os atores encarregados da governança de diferentes tipos de ecossistemas de software. Os principais benefícios do modelo proposto são: (1) compreender o contexto da governança de ecossistemas de software, (2) facilitar a comunicação e entendimento de pesquisadores sobre a área de governança de ecossistemas de software, (3) atuar como uma referência para que os profissionais construam seus próprios modelos de governança de acordo com seu contexto e suas necessidades específicas.

Para construir o modelo conceitual constructos oriundos dos mecanismos de governança foram transformados em elementos (também denominados de entidades ou classes) que compõem o modelo conceitual. Os constructos representam o vocabulário

do domínio estudado. No contexto desta pesquisa, os elementos surgiram a partir de mecanismos de governança identificados na revisão sistemática da literatura (descritos no capítulo 5). Na Seção 5.1, é descrita a lista de elementos envolvidos na governança do ecossistema de software gerada com base na tabela dos mecanismos de governança (Tabela 3). Na Seção 5.2, são apresentadas as proposições do modelo conceitual para governança de ecossistemas de software. Finalmente, na Seção 5.3, o modelo conceitual completo é expresso na notação UML.

5.1 Elementos envolvidos na governança de ecossistemas de software

Foram identificados 28 elementos que descrevem a governança do ecossistema de software com base nos estudos primários da revisão sistemática da literatura. Esta revisão investigou mecanismos relacionados à governança de ecossistemas de software através de um rigoroso processo de busca usando um protocolo de revisão detalhado. Os mecanismos de governança indicaram quais elementos precisam ser governados. Enquanto os mecanismos de governança são representados por verbos seguidos de substantivos, constructos são compostos apenas por substantivos. Por exemplo, se o mecanismo citado for "promovendo a inovação" no ecossistema de software, o elemento a ser abordado deve ser a inovação. Na **Erro! Fonte de referência não encontrada**. Cada elemento é listado e detalhado com base em uma definição dada pela literatura de ecossistemas de software ou por estudos encontrados na literatura em geral. A pesquisa de artigos gerais que estão fora da lista dos estudos primários foi necessária por dois motivos: 1) os artigos incluídos na Revisão Sistemática da Literatura não fornecem definições para todos os elementos identificados; 2) há elementos que são característicos não só do domínio de ecossistemas de software, mas de diversas áreas correlatas à negócios (ex: acordo, conflito, etc.).

Tabela 7: Elementos da governança de ecossistemas de software

Elemento	Descrição	Estudos
Ator (Actor)	Organização ou indivíduo que participa do ecossistema de software assumindo um ou mais papeis de atuação (Manikas e Hansen, 2013).	1, 5, 12, 13, 59, 61, 32, 29, 52, 47, 45, 15, 58, 4, 6, 10, 62, 55, 46, 63, 35, 27, 18, 17, 61, 38, 53, 57, 42, 19,23, 26, 36
Acordo (Agreement)	Conjunto de regras que definem o processo, as estratégias e as limitações da resolução de conflitos para atender as	15, 52

	necessidades de todas as partes envolvidas (Moore, 2014).	
Modelo de associação (Associate model)	Consiste em um conjunto de estratégias para manter, gerir, agrupar e expandir os parceiros de um ecossistema de software (van Ageren et al, 2011).	32, 56, 54, 4, 8,62, 55, 49, 30, 8, 27, 53, 31,19, 24
Certificação (Certification)	Processo que valida a conformidade com uma expectativa ou padrão, demonstrando que o software foi produzido de uma maneira específica (Voas and Ghosh, 2005).	28, 50, 53
Coopetição (Coopetition)	Colaboração entre empresas concorrentes, na esperança de alcançar resultados mutuamente benéficos (Valença <i>et al</i> , 2014).	40, 52, 61, 50, 52, 46, 58, 28, 62, 55,49, 46
Colaboração (Collaboration)	Processo recursivo em que duas ou mais pessoas ou organizações trabalham juntos em direção a uma interseção de objetivos comuns (Mistrik <i>et al</i> , 2010).	61, 50, 52, 46, 58, 28, 62,55,49, 46, 35, 17, 42, 44
Competição (Competition)	Ato de tentar alcançar os mesmos objetivos ou obter os mesmos benefícios que outro indivíduo ou organização também deseja (Mead, 2002)	40, 52
Canal de comunicação (Communication channel)	Consiste em meios de envio e recebimento de mensagens e de divulgação de informação que permitem a participação de todas as partes interessadas em discussões de trabalhos em curso, independente da sua localização (Knauss <i>et al</i> , 2014).	41, 29, 48, 52, 9, 3, 11, 14, 28, 16, 27, 31, 37
Conflito (Conflict)	Consiste no processo interativo manifestado em incompatibilidade, desacordo ou discordância dentro ou entre entidades (Rahim, 2002).	32, 52, 15, 8, 27, 57, 31, 42, 19
Barreira de entrada (Entry barrier)	Consiste em um conjunto de condições necessárias sob as quais um stakeholder é avaliado antes de formalmente passar a fazer parte de um grupo (Gawer and Henderson, 2007).	54, 45, 4, 28, 62, 30, 18, 38,53, 24, 36
Saúde (Health)	Capacidade que uma organização possui para permanecer sustentável e produtiva ao longo do tempo (Manikas and Hansen, 2013).	34, 41, 54 , 44,28, 10, 62, 55, 47, 30, 21,25, 57
Produtividade (Productivity)	A capacidade com a qual um ecossistema converte insumos em resultados (Iansiti and Levin, 2014)).	41, 65, 54, 44, 28, 10, 62, 55, 47, 30, 21, 25, 57
Criação de nicho (Niche creation)	A capacidade que um ecossistema possui de aumentar significativamente a diversidade, criando recursos valiosos e nichos (Iansiti and Levin, 2014).	41, 65, 54, 44, 28, 10, 62, 55, 47, 30, 21, 25, 57

Robustez (Robustness)	A capacidade de um ecossistema de sobreviver a rupturas, como mudanças tecnológicas e de mercado (Iansiti and Levin, 2014).	41, 65, 54, 44, 28, 10, 62, 55, 47, 30, 21, 25, 57
Informação (Information)	Acúmulo de dados organizados que representam um domínio (Shin <i>et al</i> , 2001).	16, 32, 50, 40, 29, 48, 52, 3,4, 11, 62, 30, 20, 35, 18, 17,61, 57, 31, 37
Inovação (Innovation)	Consiste em um processo iterativo iniciado pela percepção de uma nova oportunidade de mercado para a invenção baseada em tecnologia que leva à tarefas de desenvolvimento, produção e venda visando o sucesso comercial desta invenção (Garcia e Calantone, 2002).	13, 61, 7, 32, 50, 40, 48, 52, 9, 47, 3, 7, 45, 10,8, 35, 27, 18, 17, 19, 60, 61, 38, 57, 24, 43, 44
Padrão de qualidade (Quality standard)	Conjunto de requisitos que visam orientar, avaliar e mensurar a excelência da qualidade de determinado serviço ou produto (Gillies, 2011).	32, 41, 50, 40, 56, 58, 28, 62, 55, 30, 38, 57,22
Relacionamento (Relationship)	Aliança entre uma ou mais organização com troca mútua de benefícios (Van Angeren <i>et al</i> , 2011).	32, 40, 29, 56, 52, 9, 3, 4, 5, 46, 2, 63, 35, 27,57, 42, 36
Responsabilidade (Responsibility)	Dever ou obrigação e, por conseguinte, utilização dos direitos devidos à designação que gera consequências boas ou ruins passíveis de recompensa ou de punição, respectivamente (Kaler, 2002)	41, 50, 40, 56, 3, 15, 4, 5, 49, 46, 51, 63, 27,13, 42, 26, 37
Modelo de receita (Revenue model)	Define como uma organização é compensada por cada serviço ou bem fornecido (Popp, 2011).	1, 7, 3, 45, 58, 4, 5, , 28, 10, 62, 30, 23, 27,61, 38, 53, 57, 23, 36, 39
Papel (Role)	Tipo de atuação que define as responsabilidades e o grau de relacionamento da organização ou do indivíduo com o ecossistema de software an Angeren <i>et al</i> , 2011).	41, 50, 40, 56, 3, 15, 4, 5, 49, 46, 51, 63, 27, 13, 42, 26, 37
Transparência (Transparency)	Consiste no grau de clareza, veracidade ou completude envolvido na execução e divulgação de ações de responsabilidade (Fox, 2007).	15, 33, 37
Confiança (Trust)	Disposição em ser vulnerável, com base na avaliação da benevolência, competência e integridade da outra parte (Hurni and Huber, 2014).	15, 37
Criação de valor (Value creation)	Processo através do qual um negócio se torna melhor na concepção e entrega de um produto ou serviço (Gronroos, 2008).	19, 22, 53

5.2 Proposições do Modelo Conceitual

Nesta seção, respondeu-se a questão de pesquisa: “RQ2 - Como os elementos envolvidos na governança do ecossistema de software estão relacionados?” Para mapear e explicar as relações entre os elementos, foram extraídas das citações de artigos incluídos na revisão sistemática da literatura. Depois foram criadas treze proposições que sintetizam o corpo da literatura no campo da governança de ecossistemas de software. Portanto, as proposições emergiram exclusivamente dos artigos incluídos na revisão. Nos parágrafos seguintes, estas proposições são apresentadas. Cada proposição é apoiada por evidências da literatura e é ilustrada com exemplos reais de ecossistemas de software proprietários e de código aberto. A lista contendo todas as proposições estão contidas na Tabela 8.

Tabela 8: Lista de proposições do modelo conceitual

Proposição
P1: Coopetição pode gerar inovação.
P2: Inovação pode gerar criação de valor.
P3: Um modelo de associação guia a criação de valor.
P4: Um modelo de associação define um conjunto de papéis, define barreiras de entrada e governa relacionamentos influenciando a saúde do ecossistema.
P5: Saúde de ecossistemas de software pode ser mensurada por produtividade, robustez e criação de nicho.
P6: Acordos guiam a resolução de conflitos em relacionamentos em um ecossistema de software.
P7: Atores participam em um relacionamento.
P8: Modelos de receita com compensações monetárias e não monetárias atraem atores.
P9: Canais de comunicação trazem os atores a um alinhamento e pode permitir transparência para construir confiança.
P10: Canais de comunicação permite propagação da informação.
P11: Certificações e padrões de qualidade atuam como barreira de entrada para o ecossistema de software.
P12: O exercício de um papel por um ator traz responsabilidade.
P13: Responsabilidade promove confiança em relacionamentos

P1: A coopetição pode gerar inovação (Coopetition may generate innovation).

Evidência: Em um ecossistema de software, os participantes simultaneamente competem e cooperam entre si (Mizushima e Ikawa, 2011). Coopetição é entendida como a colaboração entre os concorrentes de negócios na esperança de benefícios mútuos. Portanto, é o principal motor da inovação (Valença et al, 2014). Coopetição leva a relação entre as empresas a um novo nível, onde os atores trabalham juntos para identificar

requisitos inovadores e oferecer novas soluções para atender às necessidades do mercado (Valença et al, 2014). Por exemplo, Eclipse, IBM, Oracle e SAP SE trabalham com uma variedade de organizações sob uma política antitruste de concorrência para desenvolver produtos e serviços em um processo contínuo, onde a inovação aumenta o lucro para toda a plataforma.

P2: A inovação pode promover a criação de valor (Innovation may promote value creation).

Evidência: a inovação pode produzir situações de ganho para o dono da plataforma, fabricantes de produtos e clientes (Gawer e Cusumano, 2019). Um aumento na inovação gera um aumento na criação de valor para a plataforma (Gawer e Cusumano, 2019). Por exemplo, a Microsoft colabora com concorrentes, clientes e parceiros para criar as melhores soluções para uma grande parte dos usuários de TI do mundo. Apenas 7% da receita total gerada em torno da Microsoft realmente é produzida pela própria Microsoft.

P3: Um modelo de associação orienta a criação de valor (An associate model guides value creation).

Evidência: Um modelo de associação permite que os atores se juntem e criem valor de acordo com seus papéis e posições específicos no ecossistema (Gawer e Cusumano, 2019), (van Angeren et al, 2013). Por exemplo, os atores se juntam ao ecossistema do Eclipse para capturar oportunidades de negócios. Assim, eles trazem novas funções e serviços que promovem a criação de valor. Os baixos requisitos de entrada do modelo de associação do Eclipse favorecem a participação de membros variados no ecossistema e permitem a criação de valor de várias maneiras.

P4: Um modelo de associação define um conjunto de papéis, define as barreiras de entrada e rege os relacionamentos, influenciando a saúde do ecossistema (P4: An associate model defines a set of roles, defines entry barriers, and governs relationships, influencing the health of the ecosystem).

Evidência: Relações complexas são construídas entre os atores dentro e fora do ecossistema (Mizushima e Ikawa, 2011). Relacionamentos com atores fracos podem desempenhar um papel importante na desestabilização do ecossistema (van Angeren et

al, 2016). Gerenciar relacionamentos entre atores é um fator chave de sucesso para a evolução saudável de um ecossistema de software (Iansiti e Levien, 2004). A introdução de um modelo de associação pode fortalecer os atores fracos e aproximar novos potenciais parceiros. Os modelos de associação também são um veículo para definir as barreiras de entrada, como taxa anual e certificações (van Angeren et al, 2016). O modelo de associação do Eclipse, por exemplo, possui um programa de associação com cinco classes diferentes. Para cada classe, os membros intensificam seu envolvimento com o ecossistema, aumentando o grau de responsabilidade da sua contribuição e, conseqüentemente, tornando o ecossistema do Eclipse mais saudável.

P5: A saúde do ecossistema de software pode ser medida pela produtividade, robustez e criação de nicho (Software ecosystems health can be measured by productivity, robustness, and niche creation).

Evidência: A saúde de um ecossistema é determinada pela capacidade de um ecossistema de produzir consistentemente resultados significativos (produtividade), sobreviver a rupturas do mercado (robustez) e criar nichos no ecossistema (criação de nicho) (Iansiti e Levien, 2014). No ecossistema do Eclipse, por exemplo, a produtividade é medida pelos seguintes indicadores: número de linhas de projeto de código e número de novos projetos divulgados. A robustez é medida pela relevância das contribuições, pela qualidade do código do projeto e pelo número de projetos ativos. A criação de nicho é medida através do aumento de parceiros e do número de renovações anuais de parceiros.

P6: Acordos podem orientar a resolução de conflitos em relacionamentos em um ecossistema de software (Agreements can guide the resolution of conflicts in relationships in a software ecosystem).

Evidência: Conflitos entre atores podem ocorrer em ecossistemas de software quando os relacionamentos não têm poder claramente estabelecido e quando os mecanismos de coordenação são incipientes ou mesmo inexistentes (Valença et al, 2014). Segundo Fricker (2009), os acordos são resultados de negociações bem sucedidas entre as partes interessadas e são uma maneira de resolver conflitos. O ecossistema do Eclipse, por exemplo, não tem um documento formal que possa ser chamado de acordo, mas tem um comitê responsável por resolver possíveis divergências tanto no nível técnico quanto

no nível do negócio. Os acordos são amplamente utilizados por este comitê para apoiar as partes envolvidas nos conflitos.

P7: Atores participam de um relacionamento (Actors participate in a relationship).

Evidência: Um ecossistema de software é um conjunto de atores funcionando como uma unidade e interagindo com um mercado compartilhado de software e serviços, juntamente com as relações entre eles (Jansen e Cusumano, 2012). Essas relações são frequentemente apoiadas por uma plataforma tecnológica comum ou mercado e operam através da troca de informações, recursos e artefatos (Jansen e Cusumano, 2012). O Android, por exemplo, é desenvolvido pela Open Handset Alliance, liderada pelo Google desde 2008. Essa aliança é um grupo de 84 empresas especializadas em hardware, software e telecomunicações, que participam do desenvolvimento e distribuição do Android de diversas formas (Che e Perry, 2014).

P8: Modelo de receita com fluxos de compensação monetários e não monetários atrai atores (Revenue model with monetary and non-monetary compensation streams attract actors).

Evidência: De acordo com Manikas e Hansen (2014), um modelo de receita híbrido com ganhos monetários e não monetários representa uma vantagem competitiva para um ecossistema de software porque facilita a adesão de diferentes atores nos relacionamentos permitindo que o ecossistema se expanda em várias direções. Existem vários benefícios não monetários que são essenciais para pequenos atores, como a imagem do *keystone* e a possibilidade de interagir e trocar experiências com outros atores. Compensações monetárias incluem participação nos lucros e descontos nos pagamentos de taxas anuais.

P9: Os canais de comunicação unem os atores e podem promover a transparência para construir confiança (Communication channels bring actors into alignment and may foster transparency to build trust).

Evidência: Segundo Knauss et al (2014), os canais de comunicação permitem o alinhamento entre os parceiros e tornam o ecossistema atrativo para novos atores. Canais de comunicação eficazes exigem um nível elevado de transparência. Se uma informação

relevante for removida ou omitida do canal de comunicação, a transparência será ameaçada e a confiança poderá ser quebrada (Fricker, 2010). A transparência é um dos princípios da Fundação Eclipse. Para atender a esse princípio, o conselho de administração fornece relatórios periódicos a todos os membros. Esses relatórios divulgam todas as atualizações das políticas e regras do Eclipse e reportam o status financeiro da fundação.

P10: Canais de comunicação permitem a propagação de informações (P10: Communication channels enable information propagation).

Evidência: A interação entre os parceiros aumenta a necessidade de gestão do conhecimento, a fim de apoiar a comunicação e a propagação eficiente da informação (van Angeren et al, 2013). Os canais de comunicação são projetados para permitir o compartilhamento de informações entre os atores (Jansen e Cusumano, 2012). No ecossistema do Eclipse, livros e tutoriais, bem como eventos, programas de treinamento e consultoria, estão disponíveis no portal da comunidade, onde todos os participantes do Eclipse estão agregados. Esses recursos auxiliam a comunicação entre os membros do ecossistema do Eclipse.

P11: Certificações e padrões de qualidade atuam como uma barreira de entrada para o ecossistema de software (Certifications and quality standards act as an entry barrier to the software ecosystem).

Evidência: De acordo com Jansen e Cusumano (2012), para se tornar um parceiro, uma organização pode ter que certificar seu produto ou serviço, a fim de cumprir as diretrizes do ecossistema (van Angeren et al, 2013). Os ecossistemas de software podem incentivar a formação de relacionamentos distribuindo certificações com base em critérios de qualidade (Jansen e Cusumano, 2012). Segundo Schultis (2014), as certificações e os padrões de qualidade indicam o nível de qualidade esperado dos serviços e produtos dos parceiros, atuando como barreiras à entrada do ecossistema de software. O Microsoft CRM, por exemplo dá grande importância às certificações. A empresa sabe quantos engenheiros certificados estão em cada organização parceira e quais habilidades estão faltando no momento.

P12: O exercício de um papel por um ator tem responsabilidade (The exercise of a role by an actor carries responsibility).

Evidência: A responsabilidade é um fator importante na governança do ecossistema (Baars e Jansen, 2012). Sem membros nomeados das organizações parceiras sendo responsáveis pelo ecossistema, a execução adequada da estratégia de governança não pode ser garantida (Baars e Jansen, 2012). Na Fundação Eclipse, por exemplo, um ator com um papel maior no ecossistema tem maior responsabilidade sobre suas atividades. A responsabilidade no Eclipse é descrita no documento do contrato de adesão. Este documento lista cada conjunto de direitos e responsabilidades para cada classe de membro.

P13: A responsabilidade promove a confiança nos relacionamentos (Responsibility promotes trust in relationships).

Evidência: Se a confiança é quebrada em um relacionamento, ela pode ser reparada por atores que assumem responsabilidade pessoal pela violação (Ficker, 2010). A confiança, no entanto, só pode ser reparada se a violação for um evento isolado e se o risco de futuras decepções for efetivamente mitigado (Ficker, 2010). Usando o exemplo do Eclipse mencionado na proposição P12, ao conhecer as responsabilidades associadas à sua classe, um membro se sentirá mais seguro para participar do ecossistema e isso pode aumentar a confiança dos outros membros do ecossistema.

5.3 Representação do Modelo Conceitual

Nesta seção, são apresentados os elementos, as relações e as proposições do modelo conceitual para governança de ecossistemas de software usando a notação UML. Os elementos são representados por meio de caixas retangulares (Seção 5.1), enquanto as proposições são representadas através de verbos nas relações entre os elementos (Seção 5.2). Um breve guia para os símbolos da notação UML usados no modelo é mostrado na Figura 13.

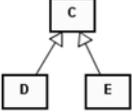
Símbolos notacionais	Descrição
	A é um elemento do modelo conceitual.
	A linha indica o relacionamento entre elementos. No modelo conceitual R é representado por um verbo na terceira pessoa do singular do indicativo. A seta indica a direção do relacionamento.
	O elemento A tem relação com o elemento B.
	Representa uma herança.
	D e E são subtipos de C.
	P é uma proposição baseada em evidências da literatura que explica o relacionamento entre os elementos.

Figura 13: Símbolos notacionais de UML usados no modelo conceitual

O esquema da Figura 14 descreve a estrutura geral da construção do modelo conceitual. Observa-se que o modelo conceitual é baseado nos estudos primários obtidos pela revisão sistemática da literatura, a partir da qual foram extraídos vários elementos ou construtos que estão inter-relacionados no modelo. É possível que um elemento tenha subelementos. Essa classificação em subelementos é representada pelo conceito de herança em UML. As proposições explicam as relações entre os elementos e foram construídas com base em evidências dos estudos primários da SLR. Na Figura 15 é mostrado o modelo conceitual proposto em UML. Uma verificação das proposições e das evidências associadas na Seção 5.2 facilita a análise do modelo, que possui 28 elementos e 13 proposições distribuídas em 27 relacionamentos.

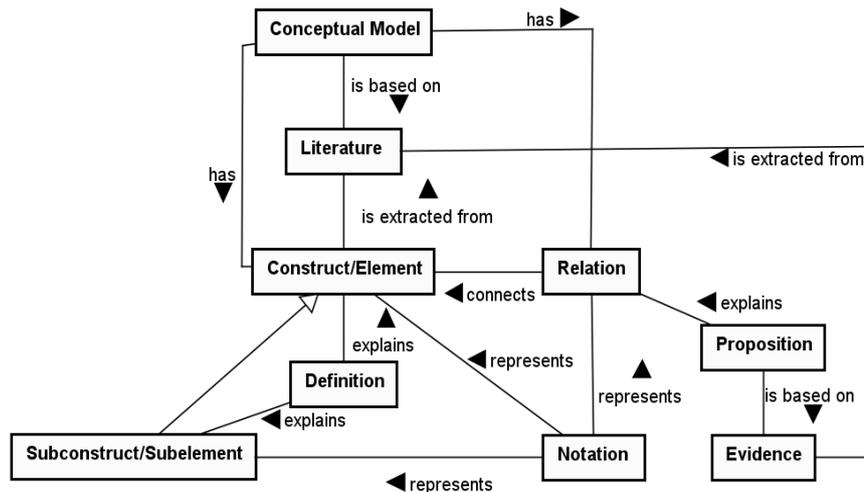


Figura 14: Estrutura da construção do modelo conceitual proposto

No modelo, os elementos centrais com maior influência sobre outros elementos são *ator*, *modelo de associação* e *saúde*. *Ator* é o elemento central do ecossistema de software. A composição de um ecossistema de software define que tipos de atores existem, com que frequência esses atores interagem e quais papéis eles assumem (Jansen et al., 2009). Assim, cada *ator* desempenha um *papel* importante na estabilidade e sustentabilidade global de um ecossistema (Dhungana et al., 2010). Manikas e Hansen (2013) fornecem uma definição detalhada dos tipos mais comuns de atores em um ecossistema de software. O conjunto de atores varia de acordo com o tipo de ecossistema. Nos ecossistemas proprietários, existe um *ator* chamado *keystone* que é responsável por governar todo o ecossistema, por estabelecer regras e procedimentos e por orquestrar o relacionamento entre os atores. Nos ecossistemas de código aberto, não há um único *keystone* todos os atores trabalham de forma colaborativa e um comitê é responsável pela tomada de decisões. Existem ecossistemas de software que adotam o conceito de governança aberta, no qual um conselho de atores representa várias organizações. Essa diretoria age de maneira semelhante a um *keystone*, no entanto, as ações de governança são compartilhadas entre essa diretoria. Parker et al. (2017) afirmam que o papel dos atores, especialmente dos desenvolvedores, é tão importante que os ecossistemas definem estratégias de “evangelismo” para gerenciar as contribuições dos desenvolvedores que se tornaram centrais para o sucesso da plataforma e para manter estes colaboradores no ecossistema de software. Segundo Parker et al. (2017) é devido a contribuição dos desenvolvedores que os ecossistemas da Apple, Google e Microsoft são tão bem-

sucedidos. Portanto, o ecossistema de software é influenciado pelo tipo de *ator*, pelos *papéis e responsabilidades* assumidos por ele e pela forma como o *ator* se comporta em relação a outros atores.

Segundo Bosch (2009), o *modelo de associação* é a principal ferramenta para atingir os objetivos de negócio do ecossistema de software. Van Angeren et al (2013) afirmam que o *modelo de associação* é um instrumento relevante usado tanto por ecossistemas proprietários quanto abertos, para criar grupos de atores em torno deles próprios e gerenciá-los. Jazayeri et al (2017) sugerem o uso do *modelo de associação* como um meio tático para escolher parceiros que fornecem soluções de valor ao ecossistema. O modelo de associação pode potencialmente melhorar a *saúde* do ecossistema (Aarnoutse et al., 2014). O *modelo de associação* ainda pode ser estruturado de maneiras diferentes. Suas características afetam os elementos relacionados (*criação de valor, relacionamento, saúde, papel*) determinando a progressão ou não do ecossistema de software. Um modelo de associação com regras flexíveis e baixas barreiras de entrada, por exemplo, permite a adesão de atores variados e amplia as possibilidades de inovação. No entanto, isso pode permitir a adição de atores fracos que podem prejudicar o ecossistema. A decisão de adotar o modelo de associação mais apropriado requer um equilíbrio cuidadoso para que a evolução do ecossistema de software não seja comprometida.

Segundo den Hartigh et al (2013), a *saúde* é um forte instrumento de governança de software, pois possibilita a análise do desempenho do ator, auxilia o engajamento do ator e monitora o ecossistema a partir de uma perspectiva financeira e de negócio. Métricas de saúde são úteis para informar como o ecossistema está evoluindo, para avaliar mudanças e para prever estados futuros (Manikas e Kontogiorgios, 2015). Den Hartigh et al (2006) listam um conjunto de métricas para *robustez, produtividade e criação de nicho*. Eles argumentam que o desafio é determinar métricas que forneçam precisão sobre os indicadores de saúde. As métricas para saúde devem ser lógicas e fáceis de usar para os gerentes, mas o problema com modelos de mensuração complexos é que não são usados porque os gerentes têm dificuldade de traçar a medida de forma clara ou lógica (den Hartigh et al., 2006). Obviamente, métricas pouco claras utilizadas incorretamente culminam em dados que não representam a situação real do ecossistema de software. Consequentemente, ações de governança baseadas em dados errados dificilmente tratarão

os problemas existentes relacionados à saúde do ecossistema. Burkard et al. (2011) mencionam a dificuldade de coletar dados relacionados a métricas de saúde porque alguns desses dados não estão disponíveis. Portanto, outro ponto é que os dados resultantes da mensuração da saúde precisam ser disponibilizados e precisam ser usados para realmente desenvolver o ecossistema de software. Caso contrário, todo o esforço colocado no processo de mensuração será perdido.

O fato de alguns elementos (por exemplo, *inovação*, *competição* e *conflito*) terem menos impacto sobre os outros não os torna menos relevantes. Os conflitos em torno do grau de abertura e compartilhamento da inovação, por exemplo, têm sido um dos problemas atualmente enfrentados pelos profissionais (Msiska e Nielsen, 2018), (Linaker et al., 2018). O desafio é decidir o quanto e como os limites do ecossistema de software deverão ser abertos para a melhoria do processo de inovação. Em relação à *coopetição*, pesquisadores têm investigado o problema da dualidade (Oruganti e Mangematin, 2018): organizações que trabalham em colaboração são as mesmas que competem umas com as outras para obter o valor gerado provocando tensões dentro do ecossistema. Nesse contexto, é importante determinar políticas de concorrência e compartilhamento de valor para garantir que a equidade entre os atores seja reforçada.

É necessário ressaltar que todos os elementos identificados no modelo conceitual são importantes para definir ações adequadas de governança. É a governança de todos eles que permitirá a sustentabilidade do ecossistema de software.

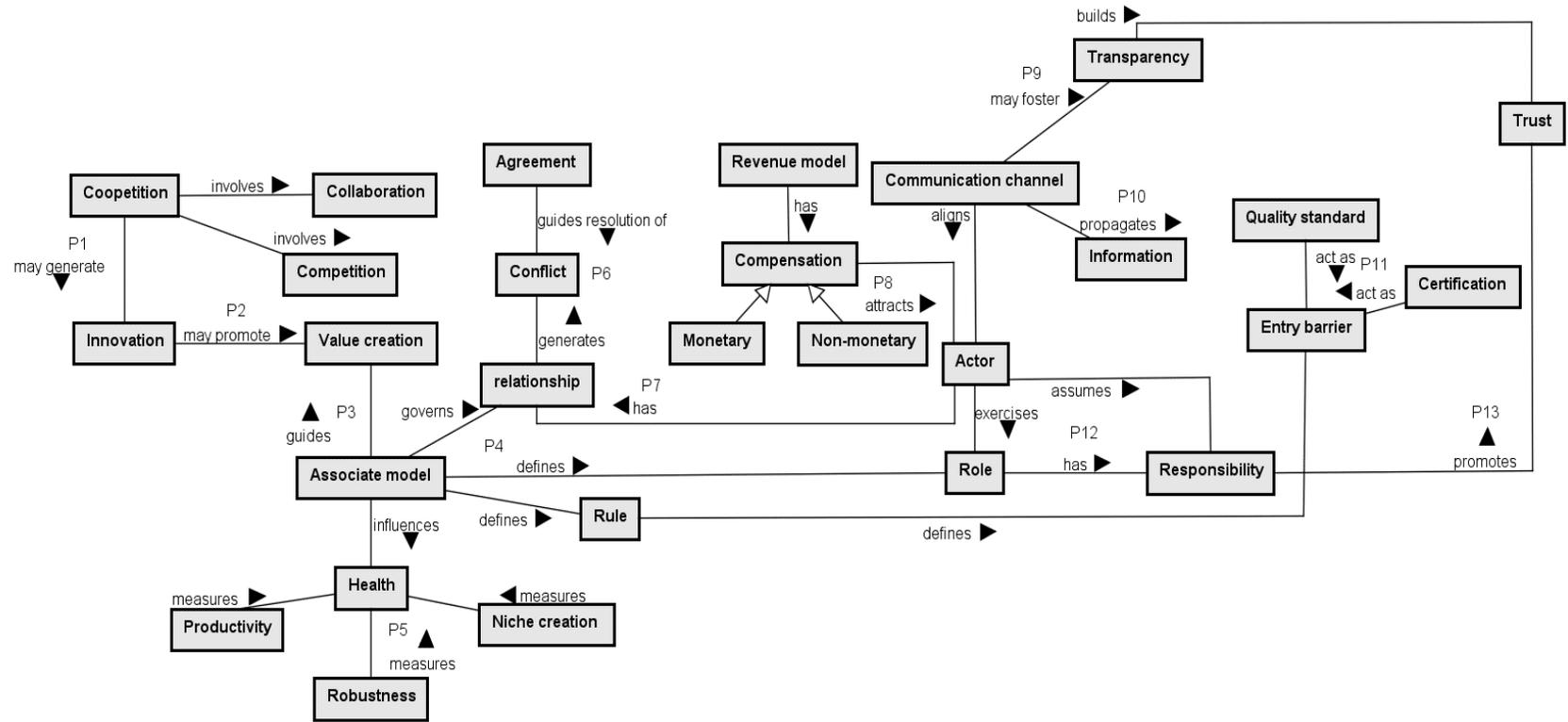


Figura 15: Modelo conceitual para governança de ecossistemas de software

5.4 Síntese do capítulo

Este capítulo apresentou os resultados da fase design da solução do ciclo de Design Science Research. Um conjunto de elementos (5.1) oriundos da revisão sistemática da literatura foi definido por meio de conceitos dados tanto por artigos da área quanto por estudos da literatura em geral. Treze proposições baseadas em evidências dos artigos incluídos na revisão sistemática da literatura foram formuladas (5.2) e representação por meio de UML (5.3) seguindo-se o paradigma conceitual, que consiste na representação de constructos/elementos e relacionamentos entre eles para explicar um fenômeno em um determinado contexto. Os elementos e relacionamentos representados por meio do modelo conceitual explicam o fenômeno da dinâmica de governança no contexto de ecossistemas de software. Três elementos do modelo conceitual se destacaram por possuir maior influência sobre outros elementos: *ator*, *modelo de associação* e *saúde*, sendo ator o elemento central de um ecossistema de software. Embora estes três elementos tenham se destacado entre os demais, todos os elementos do modelo conceitual devem ser governados para aumentar a saúde e sustentabilidade do ecossistema de software.

6 VALIDAÇÃO DO MODELO CONCEITUAL ATRAVÉS DE SURVEY COM ESPECIALISTAS

Este capítulo descreve os resultados do survey realizado com especialistas para validar o modelo conceitual para governança de ecossistemas de software proposto nesta tese e corresponde à fase de validação da solução do ciclo de design Science de Wieringa (2014). Inicialmente, é apresentado o perfil dos especialistas (Seção 6.1). Em seguida é detalhada a interpretação dos especialistas sobre a governança de ecossistemas de software e são reportadas opiniões destes especialistas sobre a relevância e os benefícios inerentes ao tema (Seção 6.2). Esses tópicos são importantes para identificar a posição dos especialistas sobre governança e entender suas respostas ao survey. A Seção 6.3 enfoca as opiniões dos especialistas sobre as proposições do modelo conceitual. Finalmente, o modelo conceitual de governança de ecossistemas de software é avaliado à luz dos critérios propostos por (Sjøberg, 2008) na Seção 5.4.

6.1 Caracterização demográfica

Dezessete especialistas da área de ecossistemas de software responderam ao survey. Deste total, 60% são doutores e 76,5% trabalham na academia. 41% dos especialistas investigaram o campo dos ecossistemas de software por mais de seis anos; apenas 5,9% começaram a pesquisar ecossistemas de software há menos de dois anos. 65% dos especialistas têm experiência prática em ecossistemas de software, que agrega desde o desenvolvimento de aplicativos para plataformas bem conhecidas a ações de consultoria. As informações demográficas dos especialistas são sintetizadas na Figura 16. O conhecimento e experiência dos especialistas possibilitaram uma avaliação profunda do modelo conceitual proposto, conforme observa-se na qualidade das respostas. Em particular, alguns especialistas são renomados autores da área. No entanto, por razões éticas seus nomes não foram revelados.

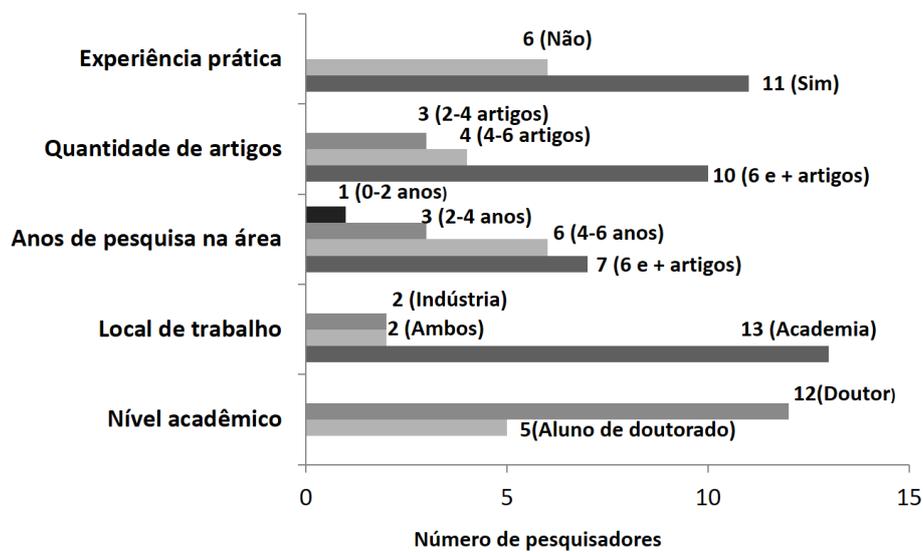


Figura 16: Características dos especialistas

6.2 Definições e benefícios da governança de ecossistemas de software

Quatorze especialistas (82%) deram sua opinião sobre o significado de “governança de ecossistema de software” (Tabela 9).

Tabela 9: Definições de governança fornecidas por especialistas

Definição de governança dada pelos especialistas
“Todas as ações, decisões e processos que visam gerenciar um ecossistema de software.” [E01]
“A governança envolve todos os fatores para gerenciar e controlar o ciclo de vida do ecossistema. Não inclui apenas práticas e processos, mas inclui definições e aplicações de estratégias para sobreviver ao longo do tempo.” [E02]
“Controle estratégico do ecossistema, com o objetivo de maximizar as metas organizacionais”. [E03]
“Gerencia a longo prazo o planejamento para a participação no ecossistema e para seu funcionamento (dependendo dos papéis existentes neste ecossistema)” [E04].
“Capacidade de coletar informações, processá-las e tomar decisões sobre aspectos centrais do ecossistemas de software, como sua arquitetura e saúde.” [E07]
“Define formas para gerenciar o ecossistema de software.” [E08]
“Ações, processos e procedimentos pelos quais uma entidade direciona um ecossistema de software para a direção desejada .” [E10]
“Conjunto de atividades para avaliar, gerenciar, controlar e melhorar o ecossistema de software e agregar valor a seus atores / stakeholders.” [E11]
“Gerenciamento deliberado de um ecossistema, visto da perspectiva de um ator (geralmente o ator central).” [E12]

<i>“Refere-se ao gerenciamento da colaboração com terceiros desenvolvedores que contribuem com soluções complementares para o ecossistema.” [E13]</i>
<i>“Envolve a definição de processos, políticas e regras gerais para orquestrar os participantes e seus relacionamentos para garantir que eles se beneficiem do ecossistema e co-criem valor.” [E14].</i>
<i>“É o conjunto de regras e papéis que definem como um ecossistema é gerenciado.” [E15]</i>
<i>“É o conjunto de regras para o ecossistema, fornecendo infraestrutura.” [E16]</i>
<i>“É um caminho para de coordenar os atores do ecossistema para atingir seus objetivos técnicos, empresariais e sociais.” [E17]</i>

Os verbos mais usados para descrever as ações de governança foram “controlar” e “gerenciar”. O termo “processo” também foi citado pelos especialistas E01, E10 e E14 e destaca a definição de Baars e Jansen (2012), que define governança de ecossistema de software como um conjunto de procedimentos e processos pelos quais uma empresa controla, altera ou mantém sua posição atual e futura em um ecossistema de software. O especialista E02 argumentou que a governança de ecossistemas de software envolve mais do que processos; engloba estratégias de sobrevivência. Da mesma forma, os especialistas E3 e E10 também associaram a governança do ecossistema à definição de estratégias. Sua compreensão está alinhada com a definição dada por Angeren et al. (2016), que consideram que a governança de ecossistemas de software envolve o uso de procedimentos e processos estratégicos para controlar, manter ou mudar o ecossistema de software. O especialista E12 observou o poder de um ator central para estabelecer princípios de governança no ecossistema. Como argumentado por Tiwana et al (2010), governança envolve a definição de quem decide o que em um ecossistema.

O especialista E13 argumentou que o principal foco da governança do ecossistema são desenvolvedores. Eles representam indivíduos ou organizações externas ao ecossistema, que geralmente fornecem valor indireto à plataforma. Esta visão tende a representar um modelo diferente de governança como o adotado por ecossistemas *open source* que não está relacionado a regras rígidas. Nesse contexto, todos trabalham de forma colaborativa e os atores trabalham para obter retornos não monetários (por exemplo, fama, conhecimento).

Quinze especialistas analisaram a relevância da governança para ecossistemas de software. O especialista E01 enfatizou a conectividade promovida pela governança do

ecossistema: “Um ecossistema de software tem conexões entre sistemas de software e a governança é obrigatória para garantir que essas conexões permaneçam”. Outros especialistas (E07, E08, E10, E11 e E13) mencionaram a relação entre governança e saúde, destacando que a governança permite a sustentabilidade e a evolução do ecossistema. Foi observado também que a governança pode equilibrar a concorrência (E02, E03 e E05), garantir a realização de metas (E04, E14 e E15) e aumentar a criação de valor (E06, E12 e E17). Na Tabela 10 são organizadas as opiniões dos especialistas sobre os principais benefícios da governança de ecossistemas de software.

Tabela 10: Benefícios de governança do ecossistema de software segundo especialistas

Benefícios da Governança de Ecossistemas de Software	
1. Preservar conexões	<i>“...governança é obrigatória para garantir que essas conexões permaneçam.” [E01]</i>
2. Preservar saúde	<i>“A importância inclui a preservação da saúde do ecossistema...” [E07][E08][E10][E11][E13]</i>
3. Balancear competição	<i>“Sem governança, o ecossistema será assumido por outros. Novas batalhas competitivas são travadas pelos ecossistemas.” [E02][E03]</i>
4. Prover vantagens competitivas	<i>“Governança pode oferecer vantagens competitivas para o ecossistema de software. ” [E05]</i>
5. Permitir criação de valor	<i>“Governança pode ser necessária para assegurar que a mútua criação de valor seja assegurada para todos os atores necessários.” [E06][E12][E17]</i>
6. Garantir o alcance dos objetivos estratégicos	<i>“A governança do ecossistema de software é essencial para orientar a evolução do ecossistema, permitindo que ele alcance os objetivos estratégicos esperados...” [E04][E14][E15]</i>

6.3 Avaliação das Proposições

Na Tabela 17, é apresentada uma síntese da validação dos especialistas para cada proposição do modelo conceitual. Foi contabilizado o total de especialistas que "concordam", "discordam" ou "não tenho uma opinião" em relação a cada proposta. Nos parágrafos seguintes são analisadas as proposições com avaliações divergentes por especialistas. Com base nas repostas foi necessário ajustar as proposições P3 e P4. Enquanto as proposições P14 e P15 foram eliminadas do modelo conceitual porque

tiveram um alto nível de discordância pelos especialistas. Vale destacar que o modelo conceitual apresentado na Figura 15 reflete a validação final pelos especialistas.

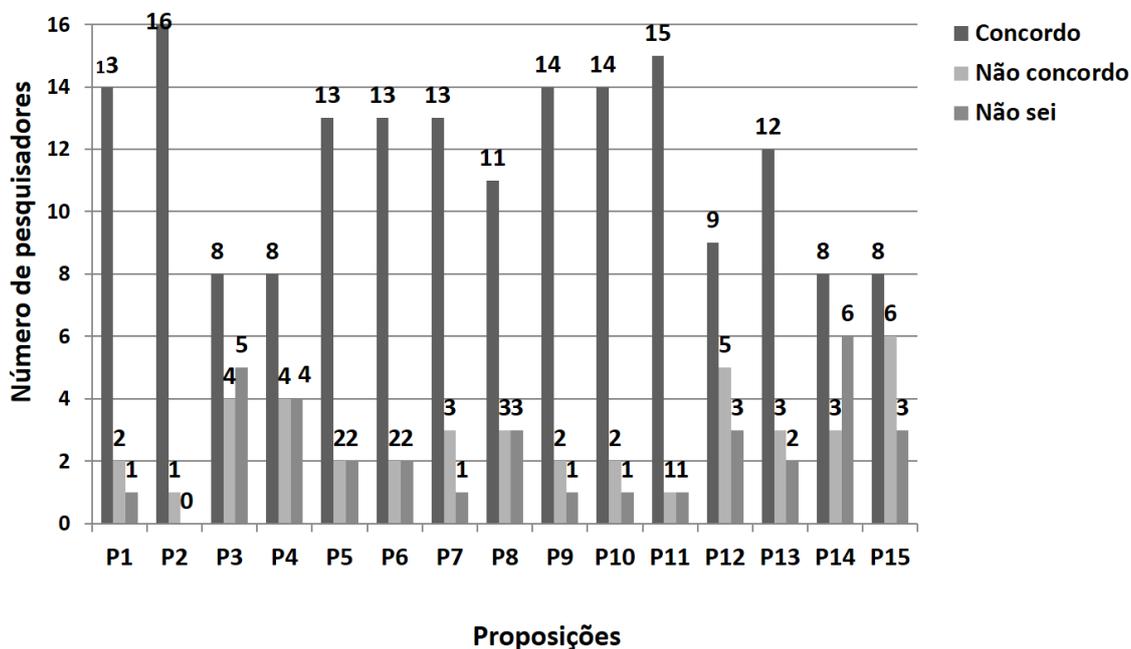


Figura 17: Validação das proposições por especialistas

P3 - Um modelo de associação orienta a criação de valor. 47,1% dos especialistas concordaram com essa proposição enquanto 23,5% discordaram e 29,4% não souberam o que responder (mesmo após consultar um glossário estes especialistas afirmaram não ter familiaridade com o termo "modelo de associação"). Os especialistas não concordaram com essa proposição, possivelmente devido a problemas de interpretação. Dois deles (E5 e E13) argumentaram que a proposição restringe as contribuições de desenvolvedores terceiros. No entanto, a autora desta tese acredita que todos os atores internos e externos que mantêm qualquer relação com o ecossistema devem ser orientados sobre as regras de criação de valor. Portanto, a proposição não é restritiva. Esta proposição foi ajustada de acordo com as sugestões propostas pelos especialistas.

P4 - Um Modelo de associação define um conjunto de papéis e governa os relacionamentos, influenciando a saúde do ecossistema. 47,1% dos especialistas concordaram com essa proposição, enquanto 23,5% não concordaram e 29,4% não

souberam o que responder. Novamente a falta de familiaridade dos especialistas com o termo "modelo de associação" afetou a análise da proposição. Um dos especialistas (E11) afirmou que o termo "saúde" não foi definido na proposição: "*O termo 'saúde do ecossistema' pode ter muitas interpretações diferentes*". Em nossa visão, se um modelo de associação não define os papéis que guiam as parcerias, isto afeta a saúde do ecossistema, que é a capacidade do ecossistema de se manter e permanecer variável e produtivo ao longo do tempo (Manikas, 2016). À luz deste argumento as melhorias propostas foram incluídas para facilitar a interpretação.

P14 - Recursos são eficientemente compartilhados entre atores com estratégias de distribuição de recursos. 47,1% dos especialistas concordaram com essa proposição, enquanto 17,6% não concordaram e 35,3% não souberam responder. A maioria dos especialistas argumentou que as evidências relacionadas a proposição P14 eram vagas. Um dos especialistas (E1) declarou: "*em alguns ecossistemas open source, as estratégias de distribuição são, na melhor das hipóteses, implícitas / não vinculantes; Não sei até que ponto uma estratégia explícita / vinculante melhora a eficiência*". Outro participante (E10) argumentou que a proposição estava melhor alinhada com os ecossistemas abertos, o que poderia influenciar a generalização do modelo. A proposição também foi criticada devido ao termo "eficientemente" (E10): "*como se pode medir / avaliar se os recursos são eficientemente compartilhados? E como avaliar se esse compartilhamento é mais eficiente quando uma estratégia de distribuição está presente?*" Portanto, achou-se pertinente excluir a proposição P14, dada a alta taxa de respostas "não concordo" (17,6%) e "não sei" (35,5%). Este resultado mostrou que a proposição P14 não está suficientemente clara do ponto de vista dos especialistas. Portanto, não há evidências práticas suficientes para apoiá-la.

P15 - Recursos atuam como uma barreira de entrada para ecossistemas de software. 47,1% dos especialistas concordaram, 35,3% não concordaram e 17,6% não souberam responder a essa proposição. Mais uma vez, o percentual de discordância e dos que não souberam responder foi superior a cinquenta por cento. Os especialistas não viram relação entre recursos e barreiras de entrada. Um deles (E01) alegou que essa proposição não representava o contexto de ecossistemas abertos: "*na maioria dos ecossistemas abertos contribuições são livremente propostas, de tal forma que a disponibilidade de um recurso não pode ser considerada uma barreira de entrada para*

o ecossistema”. Outro participante (E15) não considerou a proposição como válida, já que “os ecossistemas precisam ser um ambiente de ganhos; portanto, um ator precisa contribuir para que o ecossistema permaneça saudável”. Dada tal divergência de opiniões e alta porcentagem de discordância, optou-se por eliminar a proposição P15 do modelo conceitual.

6.4 Avaliação do Modelo Conceitual de acordo com os Critérios de Sjøberg

Nesta seção o modelo conceitual foi validado de acordo com os critérios propostos por Sjøberg (2008). Conforme descrito no método estes critérios visam analisar um modelo ou uma teoria com base em sua testabilidade, suporte empírico, poder explicativo, parcimônia, generalidade e utilidade.

Testabilidade: apesar da definição de um glossário e do uso de evidências para explicar cada proposição, os especialistas tiveram alguma dificuldade em entender e avaliar algumas proposições. Das quinze proposições, apenas duas foram refutadas. Os resultados revelam a necessidade de estudos adicionais para validar plenamente as proposições. Por meio de estudos de caso é possível confirmar a consistência e compreensibilidade dos elementos e seus respectivos relacionamentos. Portanto, a testabilidade do modelo é baixa (Tabela 11).

Tabela 11: Testabilidade do modelo conceitual

Questão	Testabilidade		
	Sim (%)	Não (%)	Não sei(%)
Os elementos e relações do modelo são compreensíveis?	35,3%	11,8%	52,9%
Os elementos e relações do modelo são internamente consistentes (sem relações contraditórias) e livres de ambiguidades?	41,2 %	17,6%	41, 2 %
Existem elementos e relações empiricamente refutáveis no modelo?	35,3 %	11,8%	52, 9%

Suporte empírico: os especialistas afirmaram que os estudos incluídos na revisão sistemática da literatura e utilizados para elaborar as proposições fornecem suporte

empírico suficiente para formular o modelo conceitual para governança de ecossistemas de software. Portanto, o suporte empírico do modelo conceitual é alto (Tabela 12).

Tabela 12: Suporte Empírico do Modelo Conceitual

Suporte empírico			
Questão	Sim (%)	Não (%)	Não sei(%)
1. Existem estudos empíricos que confirmam a validade dos elementos e relações do modelo?	58.8%	5.9%	35.3%

Poder explicativo: O Especialista E07 argumentou que alguns elementos (por exemplo, "confiança" e "conflito") não são geralmente compreendidos, enquanto outros especialistas (E11, E13 e E14) observaram que vários elementos ainda estão sob investigação por pesquisadores da área. Portanto, percebe-se a necessidade de estudos empíricos adicionais para consolidar tais construtos. Como só foram realizados estudos de casos posteriormente à aplicação do survey, considera-se o nível de poder explicativo do modelo como moderado (Tabela 13).

Tabela 13: Poder explicativo do modelo conceitual

Poder explicativo			
Questão	Sim (%)	Não (%)	Não sei (%)
Os elementos e relações do modelo são bem compreendidos pela comunidade de ecossistemas de software?	52.9%	23.5%	23.5%
Existem suposições ad hoc relacionadas aos elementos e relacionamentos?	29, 4%	11,8 %	58, 8 %
O modelo para governança de ecossistema de software é simples de entender?	76,5%	17,6%	5,9%

Parcimônia: Ao longo da construção do modelo vários itens foram excluídos porque a alta quantidade de elementos e relacionamentos dificultava a compreensão. O refinamento final foi realizado baseado na revisão dos especialistas. Mesmo agregando os elementos com o mínimo de símbolos notacionais UML, o modelo conceitual resultante ainda é grande. Portanto, a parcimônia do modelo é baixa (Tabela 14).

Tabela 14: Parcimônia do modelo conceitual

Parcimônia			
Questão	Sim (%)	Não (%)	Não sei (%)
Foi utilizado o mínimo de elementos e relacionamentos para construir o modelo?	31,3%	12,5%	56,3%

Generalidade: Segundo os especialistas, o modelo conceitual é mais orientado para um ecossistema de software proprietário. Eles argumentaram que extensões podem ser necessárias para atender às especificidades de um contexto *open source*. Assim, conclui-se que o nível de generalidade do modelo é moderado, com base nas respostas dos especialistas (Tabela 15).

Tabela 15: Generalidade do modelo conceitual

Generalidade			
Questão	Sim (%)	Não (%)	Não sei (%)
O modelo para governança de ecossistemas de software é generalizável para diferentes tipos de ecossistemas, como aberto e proprietário?	52.9%	35%	11.8%

Utilidade: pode-se argumentar que o modelo tem utilidade teórica porque contribui para aumentar o corpo de conhecimento da área. O modelo também tem utilidade prática, dada a visão holística que oferece sobre questões de governança. No entanto, o modelo não fornece indicações de como operacionalizar a governança dentro do ecossistema. Portanto, a utilidade do modelo é moderada (Tabela 16).

Tabela 16: Utilidade do modelo conceitual

Utilidade			
Questão	Sim (%)	Não (%)	Não sei (%)
O modelo possui utilidade prática e / ou teórica?	64.7%	0.0%	35.3%

6.5 Síntese do capítulo

Este capítulo apresentou os resultados da fase de validação da solução do ciclo de Design Science Research (Figura 5). A validação foi feita através de um survey com dezessete especialistas que possuem conhecimento tanto prático quanto teórico sobre ecossistemas de software (6.1). Os especialistas apresentaram suas opiniões sobre a definição de governança e sobre os benefícios da sua adoção (6.2) sendo “preservar saúde” o benefício mais mencionado. Os especialistas avaliaram o modelo com relação a validade das proposições (6.3). Das quinze proposições inicialmente formuladas, duas foram refutadas devido ao alto nível de discordância quanto a sua aceitação (P14 e P15) e duas foram totalmente reformuladas (P3 e P4). Os especialistas também avaliaram o modelo conceitual como um todo sob os critérios de testabilidade, suporte empírico, poder explicativo, parcimônia, generalidade e utilidade. Estes critérios foram definidos por Sjöberg (2018) e são amplamente utilizados por pesquisadores para validar teorias baseadas em hipóteses ou proposições. A validação do modelo por meio do survey possibilitou o seu refinamento para posterior aplicação.

7 IMPLEMENTAÇÃO DO MODELO CONCEITUAL

Este capítulo descreve a fase de implementação da solução do ciclo de design Science de Wieringa (2014), conforme descrito na Figura 5.

7.1 Implementação sob a visão de praticantes

7.1.1 Visão geral

Para analisar a viabilidade prática do modelo de governança do ecossistema de software, foram realizadas entrevistas com profissionais que conforme descrito no método são aqui denominados de praticantes (Figura 17). A experiência destes profissionais com relação a governança nos ecossistemas de software em que estes atuam permitiu obter novos insights sobre a ocorrência dos elementos e relacionamentos do modelo conceitual na prática.

Tabela 17: Características dos Praticantes

Praticante			Organization		
ID	Papel	Tempo na organização	Nome da organização	Papel da organização	Ecossistema de software
PA01	Gerente de vendas	2 anos	Netbull	Fornecedor	Microsoft
PA02	Gerente de projetos e processos	8 anos	Software AG	Keystone	Software AG
PA03	Desenvolvedor	3 anos	Motorola	Fornecedor	Eclipse Foundation
PA04	Desenvolvedor	1,5 anos	Have Health	Fornecedor	Google
PA05	Gerente de projetos	3 anos	Alliance Academy	Parceiro de educação	Apple

As questões contidas no tópico B do roteiro de entrevistas (Apêndice C) foram essenciais para compreender como a governança é conduzida nos ecossistemas de software nos quais os praticantes atuam. Ao responder estas questões, os praticantes validaram as proposições do modelo conceitual mesmo sem ter conhecimento teórico relacionado a ecossistemas de software. As respostas foram tratadas por meio de análise temática com o auxílio do software Weft QDA. Na seção em sequência são mostrados os resultados das entrevistas considerando-se as proposições do modelo conceitual.

7.1.2 Resultados das Entrevistas

Três praticantes (PA01, PA02, PA05) confirmaram a existência de competição entre os atores em seus ecossistemas, que é um dos elementos do modelo conceitual que compõe a proposição P1. *“A colaboração existe, mas também há concorrência. Um exemplo muito forte é o Google: é nosso cliente; no entanto, se estamos falando de big data, a Software AG concorre com o Google, pois ambos oferecem armazenamento em nuvem. Então, mesmo sendo um cliente, também é um concorrente. (Em particular) A Software AG governa a concorrência (entre os atores), pois precisa cuidar do cliente e do fornecedor”*, argumentou o praticante PA02.

O praticante P05 destacou a visão de "coopetição", ou seja, competição e colaboração, que também consiste em um elemento que compõe a proposição P1: *“vemos casos de cópias e brigas de patentes entre a Apple e a Samsung, e os displays que a Apple desenvolve para o iPhone são displays da Samsung; então, a Apple depende da Samsung, mas há uma competição por patentes dessas empresas ”*. O praticante PA03 citou o conflito comum de interesses entre os atores: *“às vezes, uma empresa quer muito um determinado recurso, enquanto outros estão mais interessados em outros de acordo com seus negócios”*.

O praticante PA02 confirmou a relação entre competição e inovação (**Proposição P1**): *“usamos o Eclipse e o Software AG tem uma ferramenta de desenvolvimento que usa o Eclipse assim como o java usa o ambiente do Eclipse. Toda vez que o Java executa uma nova versão, nós o estudamos em nosso laboratório para verificar se é possível implementá-lo também em nossa ferramenta. Então, evoluímos nossa ferramenta de desenvolvimento devido à concorrência ”*. Segundo o praticante PA03, a comunicação pode promover a inovação. A proximidade entre as empresas e vários usuários sempre gera novas ideias para evolução de recursos.

O praticante P05 confirmou que a inovação pode promover a criação de valor (**Proposição P2**), mas destacou: *“A Apple nem sempre se preocupa com a inovação. Sua principal preocupação é a criação de valor [PA05]. ”* Os praticantes PA01 e PA03 discutiram os *modelos de associação* dos ecossistemas Eclipse e Microsoft. *“Existem diferentes tipos de mantenedores; de acordo com sua função, eles precisam pagar um valor e alocar recursos dentro da Eclipse Foundation; no entanto, eles têm maior autonomia para a tomada de decisões ”*, explicou PA03. O praticante PA01 reforçou a

relação entre o *modelo de associação* e a *criação de valor* (**Proposição P3**). Segundo ele, a Microsoft classifica os atores em dois níveis, "ouro" e "prata". Parceiros de ouro trabalham com projetos que custam cerca de duzentos mil dólares por ano, enquanto parceiros de prata estão envolvidos em projetos com taxas mais baixas. *“A Microsoft não limita o número de parceiros. Empresas com orçamento para investir em inovação e criar produtos baseados na plataforma Microsoft se destacam. Eles criam aplicativos, personalizam soluções baseadas na plataforma Microsoft e agregam valor à sua implementação e processo de construção de marca ”*, descreveu o praticante PA01.

A saúde (**Proposição 4 e proposição 5**) foi percebida através do crescimento dos ecossistemas de software. Segundo o praticante PA01, *“a Apple chegou neste ano a um trilhão de dólares”*. A Microsoft, a Apple, a Software AG e a Eclipse Foundation promovem seus produtos em eventos que se juntam a parceiros para apresentar soluções futuras. Isso significa que os ecossistemas estão preocupados com sua própria evolução. *“A Microsoft introduz o roteiro de novos produtos em eventos específicos para saber como o mercado recebe a solução e entende a aplicabilidade de novos produtos para problemas reais”* [PA01].

O praticante PA01 discutiu as noções de conflitos de negócios (**Proposição P6**): *“Se a Microsoft sugerir um parceiro de implantação A e o cliente escolher o parceiro de implantação B, a Microsoft não cessará essa parceria. Isso não influenciará essa decisão. No entanto, isso influencia o cliente a ver o parceiro A de uma maneira diferente, já que foi sugerido pela Microsoft ”*[PA01]. Em termos de conflitos sociais e técnicos, o profissional PA02 destacou que o gerente de projetos os resolve. Para tal, os gerentes de projeto precisam ser certificados na metodologia PRIME para resolver essas situações. Se o gerente não puder resolver o conflito, ele reporta o problema para um conselho que está em uma instância superior. *“Este conselho tem uma comissão que analisará cada caso para resolver o conflito; mas geralmente fica no nível de gerenciamento de projetos ”*, explicou o praticante PA02. De acordo com o profissional PA03, a Eclipse Foundation resolve conflitos técnicos relacionados a licenças de software por meio de acordos. Todas as colaborações devem seguir as regras de licença definidas pelo Eclipse Foundation. *“Essas colaborações precisam ser controladas; (elas) só são permitidas se não houver código proprietário ou incompatibilidade com a licença do Eclipse ”*, explicou o praticante PA03. Portanto, confirmamos a Proposição P6. Como os profissionais

trabalham em empresas que possuem parcerias com os ecossistemas de software mencionados na Tabela 17, a **Proposição P7** é naturalmente confirmada, ou seja, os atores participam dos relacionamentos.

Com relação à **Proposição P8**, que envolve as noções de compensação monetária e não monetária, foi observado que o praticante PA05 mencionou os descontos obtidos por sua participação no ecossistema da Apple, que ele utiliza para comprar produtos e serviços da empresa. O praticante PA04 indicou a experiência adquirida com sua participação: *“você cria uma rede diferenciada”*. De acordo com o praticante P01, os atores do ecossistema da Microsoft recebem benefícios monetários por seu desempenho: *“A Microsoft define metas para esses grandes parceiros; uma vez que essas metas sejam alcançadas no ano fiscal da Microsoft, ela oferece recompensas monetárias para os parceiros que tiveram um bom desempenho”*. De acordo com o praticante PA03, *“Outras pessoas têm diferentes interesses, como usar a marca do Eclipse, ter seu nome reconhecido ... um interesse mais pessoal.”* O praticante P02 destacou que a Software AG compartilha lucros com parceiros que trabalham em projetos conjuntos.

Os praticantes mencionaram vários canais de comunicação usados pelos ecossistemas de software para promover transparência e confiança (**Proposição 9**) e para propagar informações (**Proposição 10**). O ecossistema da Software AG promove a comunicação através de relatórios de status, que informam os clientes sobre o progresso dos projetos: *“A comunicação é feita através de apresentações, o gerenciamento é feito por meio de ferramentas de gerenciamento de projetos, e-mails, reuniões, web conferências ... [PA02]”*. O praticante PA02 também destacou que a comunicação ocorre abertamente entre as partes. Nenhuma das partes pode permanecer inconsciente de um assunto específico dentro do projeto: *“Se você tem um gerente de projeto da Software AG e tem um gerente de projeto da empresa parceira, essa comunicação é a mesma para ambos os lados [PA02].”* Por sua vez, no Eclipse a comunicação envolve listas de grupos de notícias, isto é, listas de discussão para coordenar todos os grupos. *“Há o portal do Eclipse, fóruns, wikis, etc., que serve tanto para a comunidade de usuários externos quanto para a comunicação da própria Fundação Eclipse”*, descreve PA03.

De acordo com os profissionais PA1 e PA2, os *keystones* exigem certificações e padrões de qualidade de seus parceiros (**Proposição 11**). *“A Microsoft não é flexível. O parceiro deve passar pelo processo de certificação a cada três anos. Eu posso estar em*

conformidade por dois anos, mas não cumprir as metas no terceiro ano. Então eu não sou mais um parceiro de prata ou ouro ”, relatou o praticante PA01. “Todos os parceiros que fornecem algum tipo de serviço ou produto para o Software AG precisam ser certificados e devem provar a existência dessas certificações”, adicionou o praticante PA02.

Em relação às **proposições P12** (O exercício de um papel por um ator traz responsabilidade) e **P13** (Responsabilidade promove confiança em relacionamentos), o praticante PA03 afirmou que quando a Software AG tem um parceiro com forte competência em um produto em particular, a Software AG delega a responsabilidade para entregar o produto a este parceiro, mas para o cliente esta parceria não é visível. A Software AG confia que o parceiro entregará o produto e os clientes confiam na capacidade do Software AG.

7.1.3 Avaliação da implementação por praticantes

O modelo conceitual para apoiar a governança de ecossistemas de software instanciado a partir da pesquisa com especialistas é mostrado na Figura 18. Os elementos do modelo conceitual são mostrados em branco. Em amarelo são representados elementos mencionados pelos praticantes.

De acordo com os praticantes, o processo de coopetição envolve conflitos de interesse entre os atores e que a comunicação, quando bem fomentada, pode promover a inovação. Os praticantes ainda concordaram que a briga por patentes pode gerar ainda mais competição e que licenças de software compatíveis entre si são relevantes ferramentas para controlar a colaboração, visto que se as licenças divergirem entre si, a aplicação não pode ser desenvolvida, logo, a colaboração não pode ser finalizada. Os praticantes ainda classificaram os tipos de conflitos em social, de negócio e técnico e afirmaram que cada ecossistema de software possui uma estratégia distinta para gerenciar estes conflitos. Segundo os praticantes, a informação em ecossistemas de software é difundida por meio de portais, wikis e conferências.

Percebe-se que o modelo instanciado não tem grandes variações em detrimento do modelo conceitual proposto. Apesar de todas as proposições terem sido confirmadas sentiu-se a necessidade de enriquecer as aplicações com estudos de caso com informações complementares sobre a hierarquia de governança normalmente definida pelos

ecossistemas de software, tipos de métricas de saúde, tipos de modelos de associação, tipos de compensação, etc, tanto para confirmar as proposições de forma mais concreta, quanto para investigar com maior profundidade como a governança é conduzida por ecossistemas distintos. Esta investigação é descrita nas próximas seções.

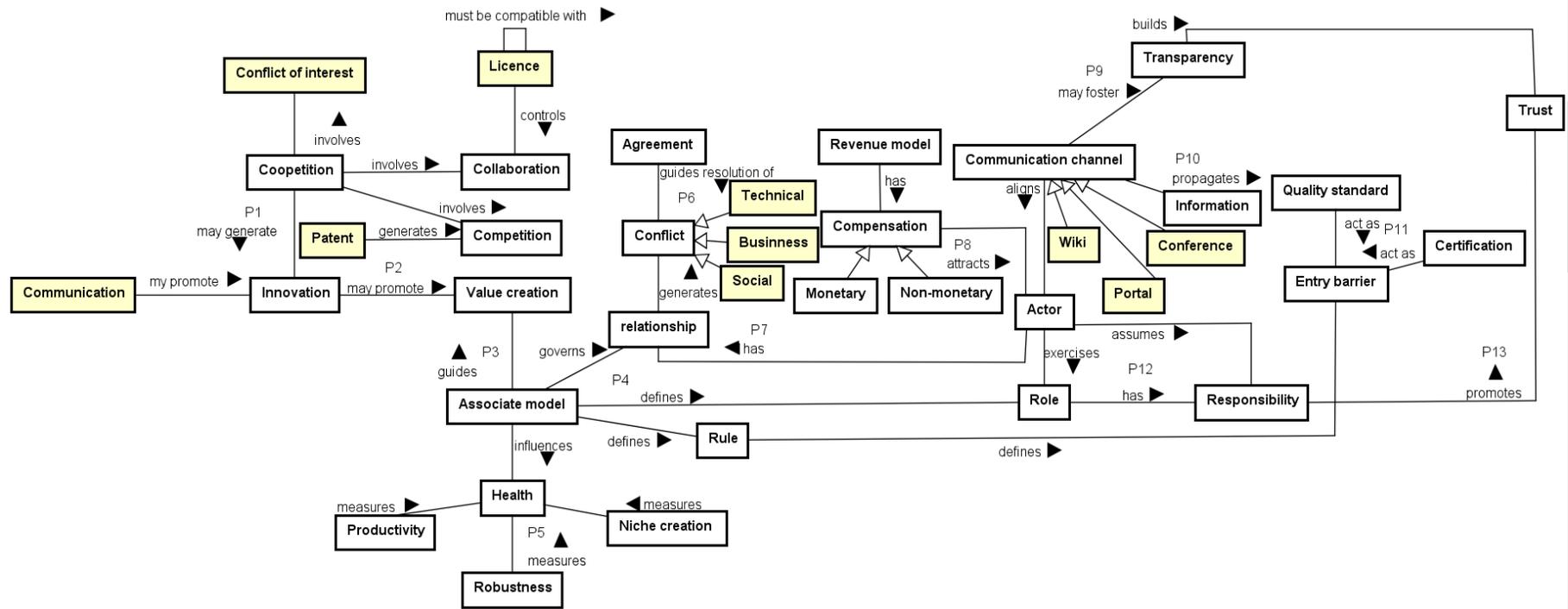


Figura 18:Aplicação do modelo conceitual para a perspectiva dos praticantes

7.2 Estudo de caso I- ecossistema de software do Eclipse

7.2.1 Visão Geral

O Eclipse foi fundado em 2001 pela IBM com o objetivo de se tornar a maior plataforma aberta de desenvolvimento integrado do mundo (Eckhardt et al, 2014). O Eclipse é baseado em Java, mas inclui uma ampla gama de plug-ins de diversos tipos. Além de ser uma bem-sucedida plataforma de desenvolvimento de software, o Eclipse apoia a modelagem e a implementação de aplicações web, o paradigma de orientação a serviços e o gerenciamento de dados (Mens et al, 2014).

Uma comunidade de organizações e indivíduos trabalha colaborativamente para a evolução dos produtos oferecidos pelo Eclipse. Esta comunidade é governada desde 2004 sob os princípios de meritocracia, transparência e igualdade de oportunidade para todos por uma corporação sem fins lucrativos chamada de Fundação Eclipse (Figura 19). A Fundação Eclipse tem a responsabilidade de administrar, cultivar, promover e divulgar a comunidade, os serviços e os produtos do Eclipse. As ações e a hierarquia da Fundação são baseadas em leis de governança definidas em documentos formais acessíveis ao público.

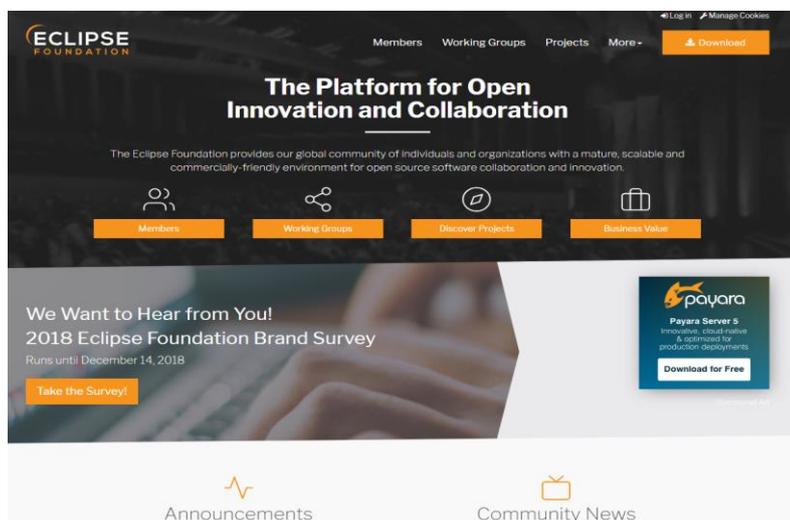


Figura 19: Página principal do ecossistema de software do Eclipse

7.2.2. *Aplicação das Proposições do Modelo Conceitual para o Eclipse*

P1: Coopetição pode gerar inovação: O Eclipse gerencia uma variedade de organizações sob uma política antitruste de competição para desenvolver produtos e serviços em um processo contínuo, onde a inovação aumenta o lucro para toda a plataforma. A ampla abertura à colaboração e apoio à inovação promovem a criação e o compartilhamento de valor. Ações de inovação no Eclipse são baseadas na estratégia de governança aberta e na definição de processos que permitem a colaboração de diferentes indivíduos e corporações na governança do ecossistema. A estratégia de governança aberta permite que membros de determinados níveis da Fundação possam mudar políticas, coordenar contribuições, mediar conflitos, além de apoiar o desenvolvimento e a evolução dos produtos do Eclipse. Para o contexto do Eclipse esta proposição é verdadeira.

P2: Inovação pode gerar criação de valor. A criação de valor é observada pelo uso dos plug-ins criados por diversos colaboradores, visto que os termos de licença do Eclipse permitem a ampla distribuição do código fonte. Para capturar oportunidades de negócio, vendedores se unem ao Eclipse trazendo novas funções e serviços (Mizushima e Ikawa, 2011), o que permite a cocriação de valor. Portanto, a proposição se adequa ao contexto do Eclipse.

P3: Um modelo de associação guia a criação de valor. Diferente de um ecossistema proprietário, que possui um modelo de parceria no qual um *keystone* governa e influencia toda as outras organizações parceiras, o modelo de associação do Eclipse baseia-se na adesão de membros que de acordo com sua categoria podem intervir na governança do ecossistema. Este modelo tem a premissa de ser utilizado para “ajudar os membros a ajudarem-se a si mesmos” (Baars e Jansen, 2010). O modelo de associação do Eclipse possui um programa de associação com cinco classes diferentes. Para cada classe, os atores intensificam seu envolvimento com o ecossistema, aumentando o grau de responsabilidade de sua contribuição e, conseqüentemente, tornando o ecossistema do Eclipse mais saudável. As classes do programa são: Membros associados, membros solucionadores, membros corporativos, Membros estratégicos e membros Committers (Eclipse Foundation, 2015). De acordo com Jansen et al (2011), para que um colaborador em potencial se torne um membro influente do ecossistema Eclipse, ele deve pagar taxas anuais à Fundação que podem ser reduzidas se a empresa que deseja fazer parte do

ecossistema ceder desenvolvedores em tempo integral aos projetos do Eclipse. A dedicação de programadores full-time resulta em uma comunidade altamente experiente e permite que o Eclipse se expanda em variadas direções. Os baixos requisitos de entrada do modelo de associação do Eclipse favorecem a participação de membros variados no ecossistema e permitem a criação de valor de várias maneiras. Para o contexto do Eclipse esta proposição é verdadeira.

P4: Um modelo de associação define um conjunto de papéis, define barreiras de entrada e governa relacionamentos influenciando a saúde do ecossistema. Para a maioria dos níveis de adesão do modelo de associação do Eclipse, requisitos adicionais e pagamento de taxas anuais precisam ser cumpridos. As organizações, por exemplo, devem liberar uma solução, com a incorporação ou a extensão da plataforma Eclipse no prazo de doze meses após se tornar um membro (van Angeren et al, 2011). Para diminuir as barreiras de entrada a determinados níveis, requisitos ou taxas são ajustadas às características organizacionais específicas, como o número total de empregados ou receita corporativa anual da organização (Eclipse Foundation, 2010). O atraso do pagamento das taxas culmina em acréscimo de juros e pode acarretar o desligamento do membro da Fundação pelo Quadro de Diretores. Este modelo de associação além de determinar as barreiras de entrada referentes a cada nível, lista um conjunto de direitos e obrigações que delimita a atuação do ator de acordo com o seu grau de relacionamento com o ecossistema. Logo, a proposição é válida para o Eclipse.

P5: Saúde de ecossistemas de software pode ser mensurada por produtividade, robustez e criação de nicho. No ecossistema Eclipse, a produtividade é medida pelos seguintes indicadores: linhas do código e número de novos projetos divulgados. A robustez é medida pela relevância das contribuições, pela qualidade do código do projeto e pelo número de projetos ativos. A criação de nicho é medida através do aumento de sócios e do número de renovações anuais de sócios. Para o contexto do Eclipse esta proposição é verdadeira.

P6: Acordos guiam a resolução de conflitos em relacionamentos em um ecossistema de software. O ecossistema Eclipse não possui um documento formal que possa ser chamado de acordo, mas há um comitê responsável por resolver possíveis divergências tanto em nível técnico quanto em nível de negócio. Os acordos são

amplamente utilizados por este comitê para apoiar as partes envolvidas nos conflitos. Logo, a proposição pode ser considerada verdadeira para a realidade do Eclipse.

P7: Atores participam em um relacionamento. Os comitês do Eclipse, assim como o Quadro de Diretores da Fundação são compostos por representantes de diferentes organizações como SAP, Google, Oracle, IBM, Bosch e OBEO que possuem relacionamentos entre si. Estes representantes são responsáveis por determinar políticas e estratégias para dirigir a Fundação e a comunidade (Eckhardt et al, 2014). Para o contexto do Eclipse esta proposição é verdadeira.

P8: Modelos de receita com opções monetárias e não monetárias atraem atores. A política de compensação de colaborações (Eclipse, 2008), acessível ao público no website do Eclipse, determina as regras para que um colaborador tenha a sua contribuição reconhecida. O objetivo deste documento é garantir que empresas não irão fornecer contribuições mínimas apenas visando o reconhecimento do Eclipse, sem de fato colaborar significativamente para a evolução do ecossistema. Ações de compensação não monetárias incluem: i) divulgação do nome, foto ou logo do colaborador no website do Eclipse, ii) direito do colaborador de utilizar a logomarca do Eclipse em seus produtos, iii) fortalecimento do relacionamento do colaborador com o Eclipse por meio de mudança de nível iv) Descontos em preço de produtos e serviços fornecidos por empresas que fazem parte do ecossistema v) possibilidade de o colaborador fazer parte do Quadro de Diretores e assim, influenciar a governança do ecossistema. Ações de compensação monetárias incluem o compartilhamento da receita resultante da venda dos produtos do Eclipse. Além de representantes de diferentes organizações, a comunidade do Eclipse é composta por um conjunto de desenvolvedores que não pagam taxas e não têm dedicação exclusiva à plataforma. Apesar de não possuírem influência sobre a governança do Eclipse e não poderem eleger os representantes da Fundação, estes desenvolvedores podem obter reconhecimento pessoal, experiência e visibilidade na comunidade do Eclipse. Para o contexto do Eclipse esta proposição é verdadeira.

P9: Canais de comunicação trazem os atores a um alinhamento e pode permitir transparência para construir confiança. A transparência é um dos princípios da Fundação Eclipse. Para atender a esse princípio, o conselho de administração fornece relatórios periódicos a todos os membros. Esses relatórios divulgam todas as atualizações das políticas e regras do Eclipse. Além disso, o diretor executivo emite relatórios

trimestrais sobre o status da fundação. Esses relatórios incluem: o status dos projetos de desenvolvimento, a prestação de contas de informações financeiras (relatório financeiro), informações sobre a adesão, remoção e colaboração dos membros, entre outras informações relevantes no que diz respeito à Fundação do Eclipse (Eclipse, 2015). A necessidade de transparência e de prestação de contas também é exigida dos membros que não fazem parte do quadro de diretores. Todos os membros são orientados a não fazerem parte de qualquer acordo no qual seja exigido confidencialidade, exceto se houver aprovação prévia do quadro de diretores. A política de reconhecimento determina que todas as conversas relacionadas a possíveis reconhecimentos de contribuições devem ocorrer no fórum público do Eclipse. Se a comunicação sobre o assunto for realizada via email, ou em reuniões fechadas, minutas contendo todos os detalhes acertados devem ser divulgadas. No geral as leis de Governança do Eclipse determinam que discussões de projetos, deliberações de minutas, planos de projetos, planos de implementação de novas funcionalidades e outros artefatos sejam públicos e de fácil acesso. Portanto, a proposição é válida para a realidade do Eclipse.

P10: Canais de comunicação permite propagação da informação. A fim de desenvolver o ecossistema Eclipse, a fundação Eclipse realiza eventos de marketing e conferências (EclipseSECO e Summit Eclipse Europa) anualmente nos EUA e na Europa, respectivamente. Nestas conferências, os desenvolvedores e os consumidores trocam opiniões e ideias sobre os produtos e a tecnologia do Eclipse. O Eclipse também oferece seminários, divulga seus produtos em portais online nos quais o uso destes produtos são demonstrados, além de possuir um canal de divulgação no youtube. A maior divulgadora dos produtos e serviços do ecossistema consiste na comunidade do Eclipse (<http://www.eclipse.org/community/>). Nela há um grupo ativo disposto a trocar experiências por meio de fóruns e listas de emails sobre o uso dos plug-ins e sobre o desenvolvimento dos projetos do Eclipse. Eventos, livros e tutoriais, além de programas de treinamento e de consultoria são disponibilizados no portal da comunidade. Todos estes recursos auxiliam na comunicação dos membros, na disseminação do Eclipse e consequentemente na sua evolução. O uso dos canais de comunicação também fortalece o relacionamento entre os membros e favorece a transparência da prestação de contas, que é divulgada nestes canais. Para o contexto do Eclipse esta proposição é verdadeira.

P11: Certificações e padrões de qualidade atuam como barreira de entrada para o ecossistema de software. Embora certificações e padrões sejam disponibilizados pelo Eclipse e que seja altamente recomendado que seus parceiros os adotem, documentação investigada não indica se adotar os padrões e certificações do Eclipse seja uma imposição obrigatória para fazer parte deste ecossistema, como acontece na Microsoft. É possível que estes elementos atuem como barreira, porém, a documentação encontrada não forneceu embasamento suficiente para determinar a veracidade desta afirmação. Portanto, para o contexto do Eclipse esta proposição não pôde ser confirmada.

P12: O exercício de um papel por um ator traz responsabilidade. O Eclipse possui uma hierarquia bem estruturada com papéis e responsabilidades bem definidos que são minuciosamente descritos em um documento que deve ser assinado quando uma organização ou indivíduo se torna membro de uma das classes do modelo de associação do Eclipse. Isto torna o ator ciente dos seus direitos e responsabilidades enquanto membro do ecossistema do Eclipse. Portanto, para o contexto do Eclipse a proposição é considerada verdadeira.

P13: Responsabilidade promove confiança em relacionamentos. Os atores no Eclipse podem compor o quadro de diretores ou suas comissões. O quadro de diretores possui três comissões permanentes que auxiliam na gestão da fundação: comissão consultiva de Propriedade Intelectual, comissão de finanças e comissão de compensação. A comissão de Propriedade intelectual é responsável por rever periodicamente as políticas de licença e de propriedade intelectual do Eclipse (Eclipse Foundation, 2016). A comissão de finanças é responsável por supervisionar todos os fundos financeiros da Fundação e por apresentar relatórios periódicos quanto a situação financeira do Eclipse. A comissão de compensação é responsável por compensar os membros do ecossistema de acordo com o nível de colaboração. (Eclipse Foundation, 2008). Sob a direção do diretor executivo e supervisão do Quadro de Diretores, há a Organização de Gerenciamento do Eclipse (EMO), que é responsável por definir grupos de trabalho para os projetos do Eclipse, resolver conflitos, garantir que as políticas sejam seguidas. Os projetos do Eclipse são orientados por dois conselhos: o Conselho de Planejamento (PC) e o Conselho de Arquitetura do Eclipse (EAC). Estes conselhos estão sob a direção da Organização de Gerenciamento do Eclipse (EMO). O Conselho de Planejamento é responsável pela coordenação de releases. O Conselho de Arquitetura do Eclipse é responsável por revisar

os processos de desenvolvimento do Eclipse. Os projetos também são monitorados pela Comissão de Gerenciamento de Projetos (PMC) e pelo líder de projeto. O PMC é responsável pela execução correta do processo de desenvolvimento do Eclipse. Conhecendo as responsabilidades associadas a cada classe, o ator sente-se mais seguro para participar do ecossistema de software, o que aumenta a confiança entre os membros do ecossistema. Para o contexto do Eclipse esta proposição é verdadeira.

7.2.3 Modelo Conceitual instanciado para o Eclipse

O modelo instanciado para o contexto do Eclipse é representado na Figura 20. Os elementos representados em retângulos brancos consistem em constructos do modelo conceitual proposto nesta tese e que existem na prática no ecossistema Eclipse. Os retângulos cor de rosa representam elementos do modelo conceitual que não puderam ser confirmados no contexto do referido ecossistema por insuficiência de informações fornecidas pela documentação investigada. Os elementos contidos nos retângulos amarelos representam elementos específicos identificados na dinâmica de governança do Eclipse. Estes novos elementos são considerados aplicações identificadas no contexto específico do ecossistema Eclipse. No entanto, é importante ressaltar que tais elementos podem não ocorrer em outros ecossistemas de software.

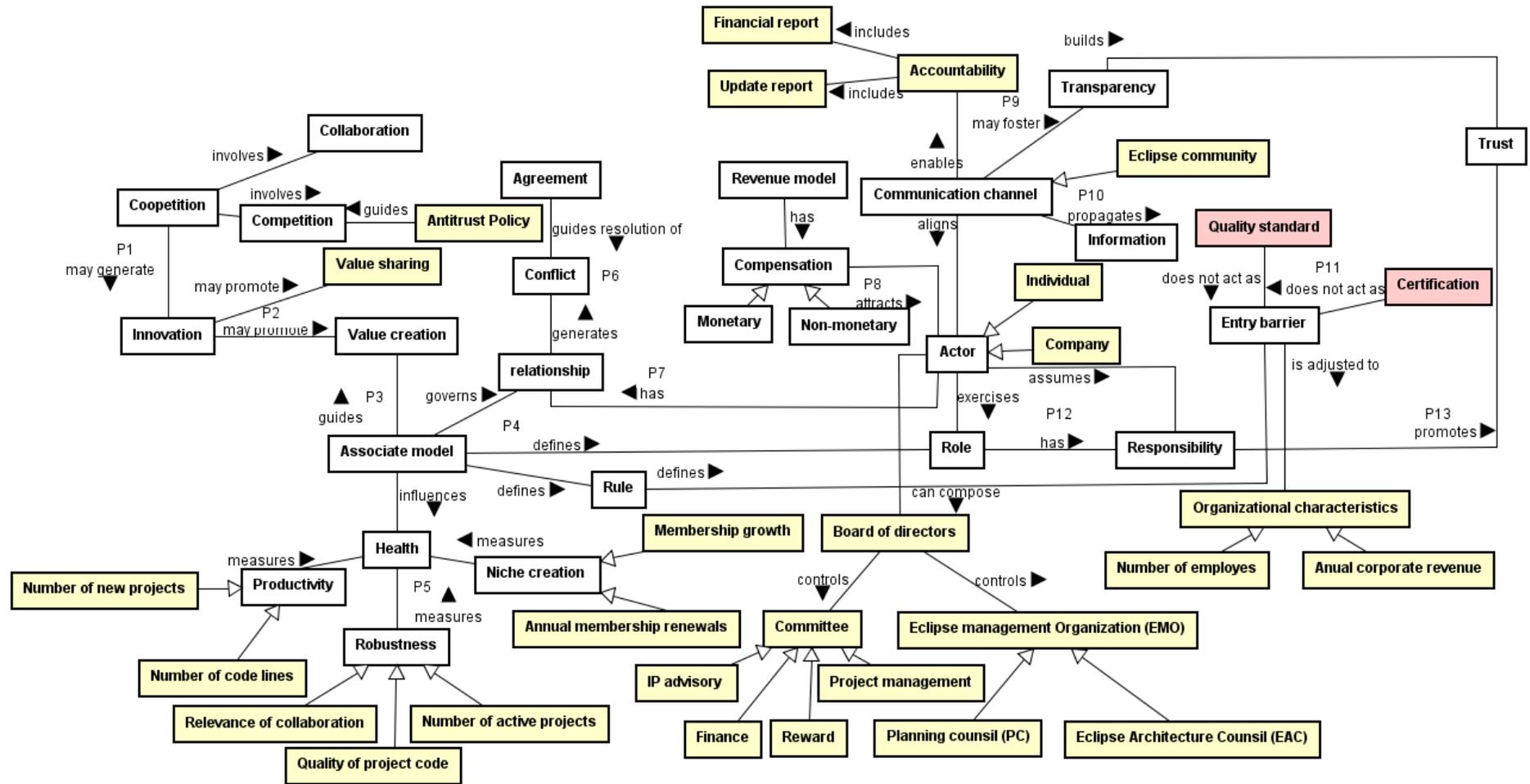


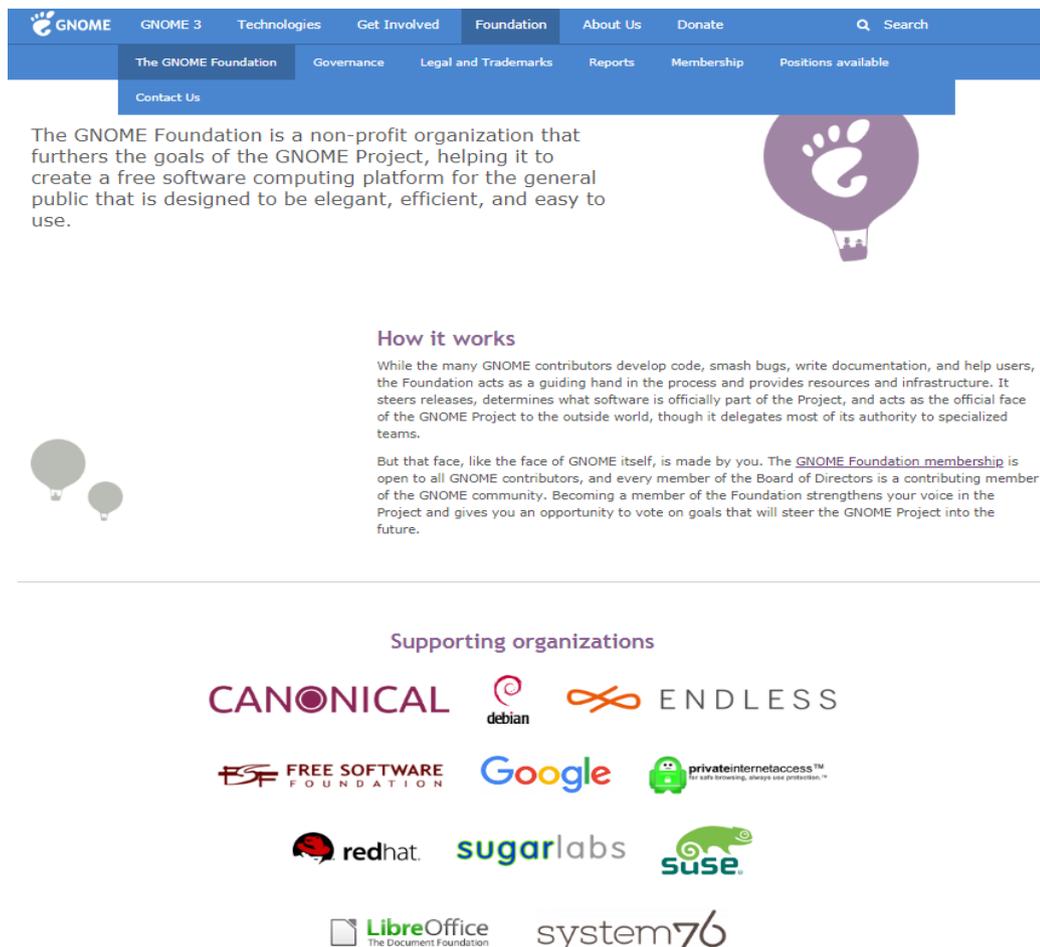
Figura 20: Aplicação do modelo conceitual para o Eclipse

7.3 Estudo de caso II – ecossistema de software GNOME

7.3.1 Visão geral

O ecossistema de software do GNOME tem o objetivo de ampliar o acesso à tecnologia por meio do desenvolvimento e distribuição de softwares gratuitos e utilizáveis por pessoas de todo mundo (

Figura 21). O GNOME possui uma coleção de software gratuito e de código aberto que compõe uma plataforma coesiva de software livre para o público em geral e é conhecido pelo seu longo histórico de desenvolvimento (Jergensen et al, 2011). Este ecossistema de software é mantido por uma grande comunidade de desenvolvedores e tem sido amplamente investigado pelos pesquisadores devido ao seu tamanho, longevidade, popularidade e evolução (Kouters et al, 2012).



The GNOME Foundation is a non-profit organization that furthers the goals of the GNOME Project, helping it to create a free software computing platform for the general public that is designed to be elegant, efficient, and easy to use.

How it works

While the many GNOME contributors develop code, smash bugs, write documentation, and help users, the Foundation acts as a guiding hand in the process and provides resources and infrastructure. It steers releases, determines what software is officially part of the Project, and acts as the official face of the GNOME Project to the outside world, though it delegates most of its authority to specialized teams.

But that face, like the face of GNOME itself, is made by you. The [GNOME Foundation membership](#) is open to all GNOME contributors, and every member of the Board of Directors is a contributing member of the GNOME community. Becoming a member of the Foundation strengthens your voice in the Project and gives you an opportunity to vote on goals that will steer the GNOME Project into the future.

Supporting organizations

CANONICAL debian ENDLESS

FREE SOFTWARE FOUNDATION Google privateinternetaccess™

redhat. sugarlabs suse.

LibreOffice system76

Figura 21: Página principal do ecossistema do GNOME

O GNOME foi criado em janeiro de 1997 e atualmente possui mais de 1400 projetos compostos por mais de 5000 colaboradores (Mens et al, 2014) que têm desenvolvido aplicações em C, C++, Python e C# para GNU e Linux. O ecossistema do GNOME é gerenciado pelo GNOME Foundation. O GNOME Foundation tem um quadro de diretores que gerencia os aspectos de negócios do ecossistema, mas os participantes dos projetos possuem alto grau de autonomia (Jergensen et al, 2011). Para fazer parte ativamente do GNOME, um indivíduo precisa apenas se cadastrar na comunidade, realizar contribuições, submete-las por meio do GIT-bz e após um processo de revisão, ter a contribuição aprovada (GNOME, 2018a).

O GNOME tem os seguintes princípios: i) Ser aberto e público, ii) Desenvolver softwares livres, iii) Ser baseado em meritocracia, ou seja, o indivíduo é reconhecido em termos não monetários pelo grau da sua contribuição, iv. Ser baseado em uma estrutura pouco burocrática e v) Ser independente, ou seja, livre de influências de organizações ou corporações.

7.3.2 Aplicação das proposições do modelo conceitual para o GNOME

P1: Coopetição pode gerar inovação. De acordo com as leis de governança do GNOME (GNOME Bylaws, 2002), um colaborador é definido como qualquer pessoa que contribua para a criação e melhoria dos projetos deste ecossistema através da criação de aplicações, documentação, tradução de relatórios e de código para outros idiomas, manutenção de recursos de todo o projeto e de outras atividades que beneficiem o GNOME. Um colaborador também pode contribuir relatando bugs das aplicações desde que tais contribuições estejam significativamente acima do nível esperado de um usuário comum. Todas as contribuições passam por um processo de revisão para serem aceitas (GNOME, 2018a). O ecossistema GNOME tem um grau de colaboração muito maior do que o encontrado em ecossistemas de software proprietários e um menor grau de competição (Mens et al, 2014). O baixo nível de competição não impede que novos plug-ins para Linux sejam criados. Para o GNOME pode-se afirmar que a colaboração pode gerar a inovação, mas o mesmo não se aplica a competição. Logo esta proposição é em parte verdadeira.

P2: Inovação pode gerar criação de valor. O foco do GNOME é que as aplicações desenvolvidas sejam acessíveis para todos. O ecossistema não possui fins

lucrativos, então a caracterização de valor para o GNOME não consiste em lucro, mas na entrega de um produto ou serviço que possa ser utilizado sem custo pelo usuário. Portanto, se o produto resultante da inovação for viável de ser utilizado, a proposição pode ser considerada verdadeira sob esta perspectiva de valor para ecossistemas abertos.

P3: Um modelo de associação guia a criação de valor. Qualquer membro eleito pode fazer parte do quadro de diretores sem necessidade de pagar taxas, qualquer indivíduo que se une ao GNOME pode contribuir sem que nada seja exigido dele. É possível afirmar que o modelo simplificado de associação do GNOME promove abertura para a criação de valor visto que não há requisitos de entrada que dificultem a adesão do membro ao ecossistema. No GNOME os membros não são classificados em bronze, prata e ouro, como acontece na Microsoft, por exemplo, onde de acordo com o nível do modelo de associação, o membro possui maior poder de atuação, mais recursos e conseqüentemente maior responsabilidade e propensão à criação de valor. Todos têm equidade na criação e compartilhamento de valor sem que nenhuma restrição seja orientada pelo modelo de associação. Considerando o conceito de valor para ecossistemas *open source*, esta abertura favorece a criação de valor e possibilita o aumento da interação entre os atores. Tudo o que é criado pode ser utilizado por todos, inclusive na cocriação de valor. Portanto, esta proposição é válida para o contexto do GNOME.

P4: Um modelo de associação define um conjunto de papéis, define barreiras de entrada e governa relacionamentos influenciando a saúde do ecossistema. O GNOME possui um modelo de associação simples no qual os indivíduos são colaboradores pertencentes a comunidade do GNOME, ao quadro de diretores da fundação, ao quadro consultivo ou ao comitê de eleição (GNOME, 2018b). A comunidade é composta por uma variedade de colaboradores que podem ser programadores, tradutores ou documentadores. O quadro de diretores é composto por colaboradores que atuam na governança do GNOME. O quadro consultivo é composto de indivíduos e organizações que patrocinam os projetos do Eclipse, mas que não têm poder de tomada de decisão. O comitê de eleição é responsável por organizar eleições para que um indivíduo seja o diretor executivo ou componha o quadro de diretores. Apesar de definir papéis, este modelo de associação não atua como barreira de entrada e não governa os relacionamentos. Ele apenas é caracterizado como um guia que indica quais papéis podem ser desempenhados, mas utilizando a documentação investigada

como base não é possível afirmar que o modelo de associação influencia a saúde do ecossistema do GNOME. Logo, para o contexto do GNOME esta proposição é em parte verdadeira.

P5: Saúde de ecossistemas de software pode ser mensurada por produtividade, robustez e criação de nicho. Anualmente, o GNOME contabiliza a quantidade de colaboradores ativos por projeto, a quantidade de mudanças incorporadas a cada projeto, a quantidade de releases por projeto, a quantidade de eventos realizados para desenvolvedores, os custos anuais do ecossistema, o número de parceiros, o custo associado a manutenção do ecossistema, o número de novos projetos, a visibilidade e reputação, assim como o seu nível de transparência (GNOME, 2017). De acordo com Mizushima e Ikawa (2011) eventos para desenvolvedores e frequência de releases se caracterizam como métricas para avaliar a produtividade de um ecossistema de software. Cardoso et al (2013) também caracterizam a quantidade colaboradores ativos como métrica de produtividade. A contabilização dos custos do ecossistema e a quantidade de parceiros são definidos respectivamente por Hyynsalmi et al (2014) e por Spauwen e Jansen (2013) como métricas de robustez. O número de novos projetos, a visibilidade/reputação e o nível de transparência são caracterizados por Dhungana et al (2010), McGregor (2010) e Monteith et al (2014) como métricas associadas a criação de nicho. Portanto, para o contexto do GNOME esta proposição é verdadeira.

P6: Acordos guiam a resolução de conflitos em relacionamentos em um ecossistema de software. Não foram encontrados insumos suficientes para analisar esta proposição. Portanto, não é possível afirmar se é verdadeira.

P7: Atores participam em um relacionamento. Durante 20 anos de evolução do GNOME, mais de 1400 projetos foram mantidos por mais de 5800 colaboradores distintos (GNOME, 2017). Estes projetos são patrocinados por organizações como Google, Redhat, Suse, Debian e Endless. Logo, para o contexto do GNOME esta proposição é verdadeira.

P8: Modelos de receita com compensações monetárias e não monetárias atraem atores. Os colaboradores do ecossistema do GNOME trabalham de forma voluntária e não têm benefícios financeiros diretos de seu envolvimento. As compensações recebidas envolvem ganho de experiência, reconhecimento pessoal e visibilidade na comunidade do GNOME. Logo, para o ecossistema do GNOME esta

proposição é em parte verdadeira visto que não foram encontrados registros de que há compensações monetárias para os atores no referido ecossistema.

P9: Canais de comunicação trazem os atores a um alinhamento e pode permitir transparência para construir confiança. A comunidade do GNOME é aberta para inscrição de qualquer indivíduo em listas de discussões que servem para reportar bugs e para troca de conhecimento. Decisões resultantes das reuniões do Quadro de diretores também são disponibilizadas em listas de emails e em uma página (GNOME, 2018b) que contém o nome dos participantes, a agenda e os pontos discutidos. Para cada ponto é fornecida a transcrição do diálogo entre os participantes, o que permite que todos os membros do GNOME conheçam os detalhes de todas as decisões tomadas. Relatórios anuais contendo reportes de eventos e de finanças também são disponibilizados anualmente na página principal do GNOME. A preocupação da fundação com o aumento da transparência por meio da divulgação de relatórios e de reuniões do quadro de diretores mostra a importância desse elemento para o GNOME. Em 2018 o GNOME recebeu o selo de prata de transparência da GuideStar, uma instituição que arrecada doativos para organizações não lucrativas. As ações para fomentar a transparência constroem a confiança dos membros com destaque para os que fazem parte do quadro consultivo, que se sentem mais seguros para investir no GNOME por meio de doativos financeiros. Logo, para o contexto do GNOME esta proposição é verdadeira.

P10: Canais de comunicação permite propagação da informação. O GNOME possui uma riqueza de informações disponíveis para a comunidade e usuários em geral. Estas informações são divulgadas por meio de sites, fóruns e relatórios. O GNOME também investe amplamente na difusão da comunicação por meio de eventos e conferências. Um conjunto de eventos denominado de hackfests (GStreamer Autumn, GTK+, Rust, GStreamer Soringer, Fractional Scaling) é realizado anualmente na Alemanha, em Londres, no México, na Espanha e no Canadá para difundir o paradigma do código aberto e para divulgar o GNOME (GNOME, 2017). Conferências como o GNOME.Asia e GUADEC atuam como canais para unir os colaboradores e para divulgação das tendências do GNOME. Logo, para o contexto do GNOME esta proposição é verdadeira.

P11: Certificações e padrões de qualidade atuam como barreira de entrada para o ecossistema de software. Qualquer indivíduo, com certificação e padrões de

qualidade ou não, pode se juntar ao ecossistema do GNOME. Como base na documentação encontrada e nas leis de governança deste ecossistema não é possível afirmar que a proposição é verdadeira para este contexto.

P12: O exercício de um papel por um ator traz responsabilidade. O fato de um colaborador atuar como programador não o impede de atuar paralelamente como tradutor, por exemplo. A responsabilidade é associada ao bom senso de colaboração. Os membros do GNOME compreendem a importância do seu papel para o ecossistema e se comprometem a cumpri-lo (GNOME, 2018b), mas esta não é uma obrigatoriedade imposta pelo GNOME. Portanto, não é possível afirmar que esta proposição seja verdadeira para o contexto deste ecossistema.

P13: Responsabilidade promove confiança em relacionamentos. No GNOME os colaboradores associados a um projeto podem migrar para outros projetos a qualquer momento. A colaboração em um projeto não dá ao indivíduo a responsabilidade de concluí-la. A confiança não é baseada em responsabilidades impostas pelo GNOME. A confiança é promovida pelo ideal que os membros compartilham baseado no paradigma do software livre: “soluções completas para todos”. Portanto, com base na documentação investigada não é possível afirmar que esta proposição é verdadeira.

7.3.3 Modelo Conceitual instanciado para o GNOME

O modelo instanciado para o ecossistema do GNOME é representado na Figura 22. Os elementos instanciados do GNOME são apresentados em amarelo, os retângulos brancos representam elementos propostos no modelo conceitual que também existem na governança do GNOME. Elementos e relacionamentos propostos no modelo conceitual que não puderam ser comprovados por meio da documentação investigada foram representados na cor rosa.

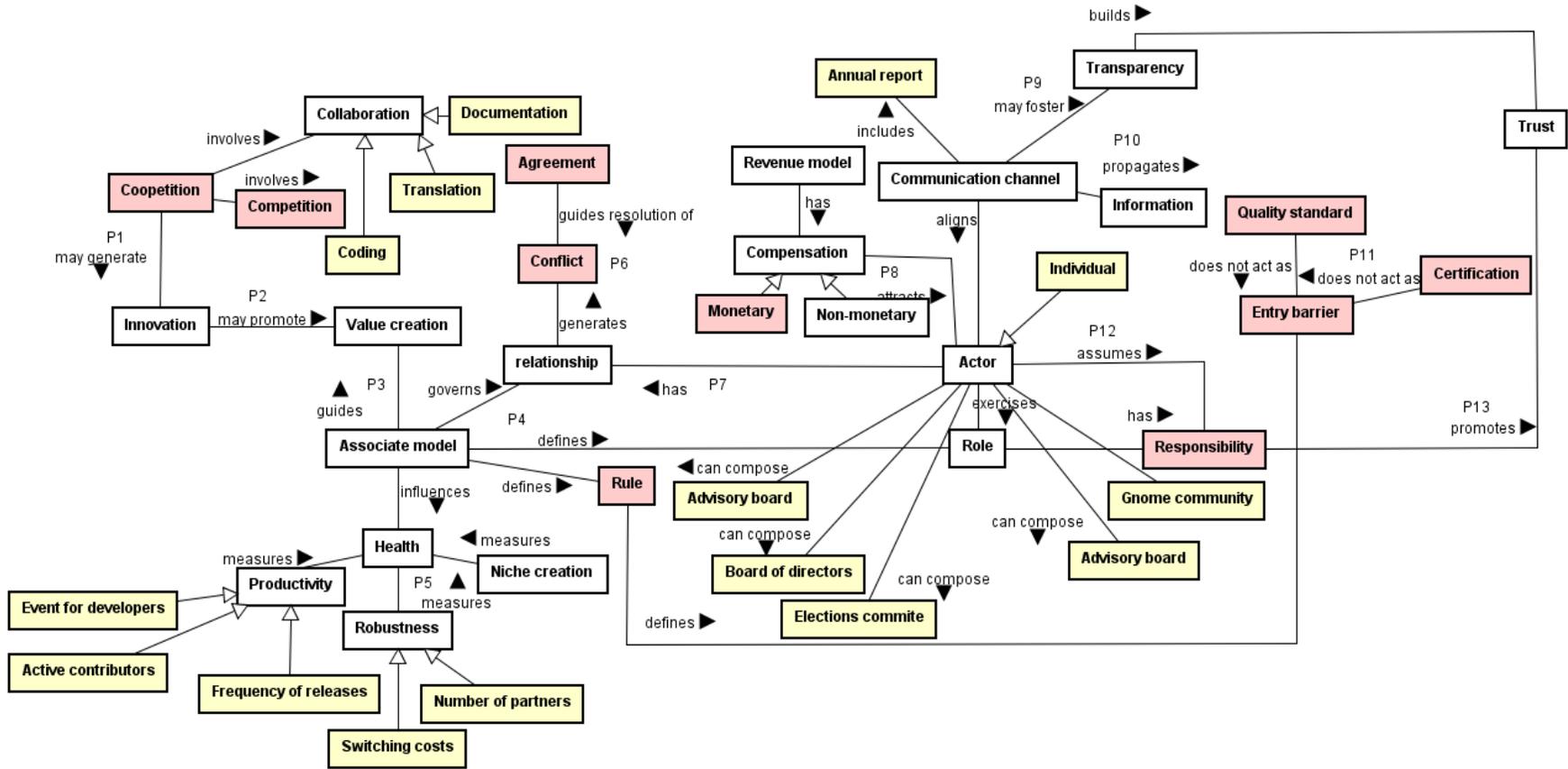


Figura 22: Aplicação do modelo conceitual para o GNOME

7.4 Estudo de caso III – ecossistema de software SAP

7.4.1 Visão geral

O SAP é um ecossistema de software que visa o desenvolvimento e a comercialização de produtos e serviços no âmbito tecnológico (Figura 23). SAP foi fundado em 1972. Atualmente possui 378,000 clientes em mais de 180 países e possui mais de 88,000 funcionários. O principal foco do SAP é desenvolver e comercializar soluções integradas de gestão, realizar consultoria e treinamento de implementação de e-commerce, além de promover a abertura e o avanço de empresas tecnológicas a nível internacional. O SAP produz software de gestão empresarial que fornece soluções de tecnologia da informação altamente integradas para planejamento e gerenciamento relacionados a recursos da empresa, relacionamentos com clientes, ciclos de vida de produtos, cadeias de suprimentos e relacionamentos com fornecedores (Gorman e Fischer, 2009).

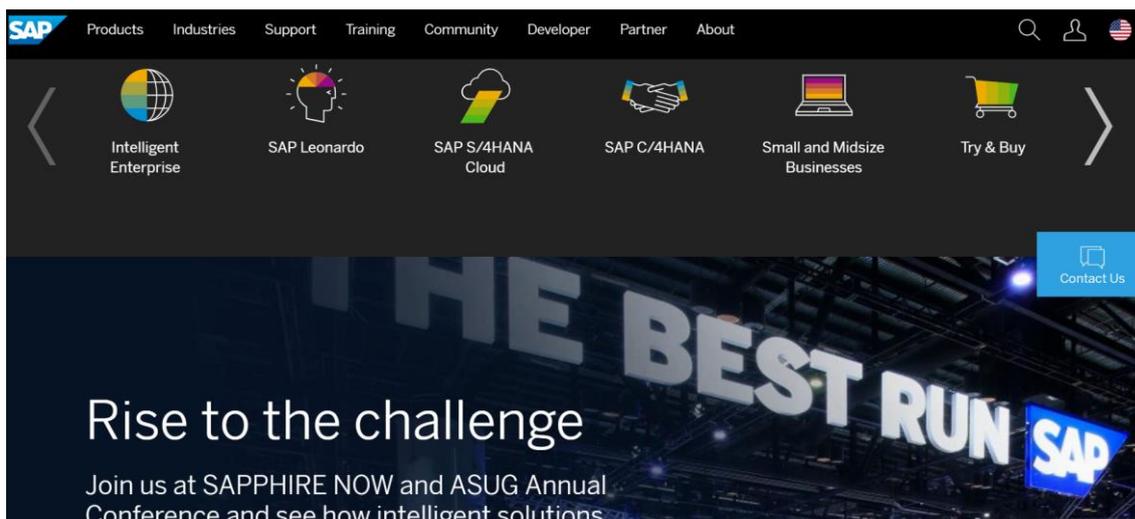


Figura 23: Página principal do ecossistema SAP

O SAP considera a governança como uma estratégia essencial para aumentar o valor e para atingir as metas corporativas (SAP, 2018a). Como o SAP é internacional com raízes Europeias, possui a base jurídica de uma empresa Europeia. Sendo sediada na Alemanha, está sujeita às leis Europeias e Alemãs. As principais características da estrutura de governança do SAP se baseiam no Código Alemão de Governança Corporativa. Como o SAP se expande aos Estados Unidos, o ecossistema também cumpre regras que se aplicam a empresas fora dos EUA listadas na Bolsa de Valores de Nova

York (NYSE). Isso inclui seguir os Padrões de Governança Corporativa da NYSE, da Lei Sarbanes-Oxley dos EUA de 2002 e da Securities and Exchange Commission (SEC) dos EUA (SAP, 2018b).

7.4.2 Aplicação das proposições do modelo conceitual para o SAP

P1: Coopetição pode gerar inovação. O SAP possui um Código de Conduta Empresarial que visa garantir o cumprimento das leis da referida organização por parceiros (SAP, 2018c). Dentre as regras estabelecidas, há um conjunto de instruções que regem a competição. O SAP foca em parcerias estratégicas que com base no código de conduta estabelecido e em regras de colaboração promovem a inovação (SAP, 2014). As inovações resultantes destas parcerias são percebidas pela geração de soluções relacionadas a Internet das Coisas, Big Data e inteligência artificial. Portanto a proposição é verdadeira.

P2: Inovação pode gerar criação de valor. Empresas como HSBC bank, Microsoft, Birchman Goup atuam em parceria com o SAP para focar na transformação digital e inovar. Os resultados da inovação muitas vezes culminam na criação de valor que é compartilhado entre estes parceiros. O SAP preconiza que o valor no ecossistema é criado partir da identificação das necessidades de negócio dos clientes que geram soluções inovadoras relacionadas a software, suporte e serviço. Portanto, para o SAP esta proposição é verdadeira.

P3: Um modelo de associação guia a criação de valor. O modelo de associação do SAP não é dividido por classes, como acontece no Eclipse e na Microsoft (SAP, 2018b). A hierarquia do SAP é composta por um quadro de diretores e por comissões que determinam a governança do ecossistema do SAP, mas não há classificação em prata, bronze ou ouro, por exemplo. Mesmo não sendo dividido em classes, o modelo de parceria do SAP guia a criação de valor porque cada papel orienta as funções e responsabilidades de cada membro. O SAP possui vasta experiência em selecionar parceiros de sucesso que direcionam o ecossistema para a criação e co-criação de valor (Ceccagnoli et al, 2012). O modelo de associação do ecossistema auxilia neste propósito. Portanto, para o SAP esta proposição é verdadeira.

P4: Um modelo de associação define um conjunto de papéis, define barreiras de entrada e governa relacionamentos influenciando a saúde do ecossistema. O

modelo de associação do SAP define direitos e deveres inerentes a cada papel assumido pelos atores do referido ecossistema (SAP, 2018b). No SAP o modelo de associação também define barreiras de entrada. Com relação ao quadro de diretores, por exemplo, apenas colaboradores do SAP podem fazer parte dele. Outros parceiros não possuem direito à representação no quadro. Apesar de restringir a atuação dos parceiros neste sentido, a definição do modelo de associação baseada em uma forte governança influencia a saúde do SAP que tem um histórico de sucesso desde sua fundação em 1972. Logo, a proposição é verdadeira.

P5: Saúde de ecossistemas de software pode ser mensurada por produtividade, robustez e criação de nicho. Anualmente, o SAP divulga resultados de mensuração do aumento de receita, da mudança de custos, além da satisfação do stakeholder e do colaborador. De acordo com os autores dos estudos (Jansen e Cusumano, 2012), (den Hartigh et al, 2006) e (Jansen, 2014), estas métricas se enquadram na classificação de robustez do ecossistema de software. O SAP também mensura a capitalização do mercado, que segundo os estudos (Knauss et al, 2014), (Jansen, 2014) e (Koch e Kerschbaum, 2014) se refere a métrica de criação de nicho. O SAP avalia a produtividade por meio da mensuração do número de patentes (Huang e Ceccagnoli, 2013), (Mhamdia, 2013), e da média de aumento de clientes (West e Wood, .2011), (Cardoso et al, 2013). Para realizar a mensuração, o SAP utiliza a solução SAP Digital Boardroom, um software analítico que permite o acompanhamento das métricas em tempo real e que gera relatórios sempre que solicitado. Portanto, a proposição é válida para o contexto do SAP.

P6: Acordos guiam a resolução de conflitos em relacionamentos em um ecossistema de software. O código de conduta do SAP determina regras para resolução de conflitos de interesse, mas na documentação encontrada não há menção à acordos como meio para resolução de conflitos sociais, técnicos ou de negócio, portanto para este contexto a proposição não pôde ser confirmada.

P7: Atores participam em um relacionamento) As parcerias são fundamentais para a estratégia de plataforma da SAP. Sua rede de provedores de soluções de software, revendedores, distribuidores e parceiros de tecnologia e serviços está entre os maiores da indústria (Ceccagnoli et al, 2012). O SAP possui mais de 17.000 parceiros em todo mundo que constroem, vendem, fornecem e executam serviços e soluções tecnológicas paras

empresas de médio e grande porte. O SAP preconiza que a rede de parceiros é um fator vital para a sustentabilidade do ecossistema (SAP, 2018b). Portanto, a proposição é válida para este contexto.

P8: Modelos de receita com compensações monetárias e não monetárias atraem atores. Compensações monetárias no SAP incluem participação nos lucros. Além disso membros dos conselhos recebem remuneração. Compensações não monetárias incluem direito a voto. Todos os parceiros atuantes e sem pendências do ecossistema do SAP tem direito a eleger membros dos conselhos por meio de voto. Os parceiros também podem utilizar o logotipo e o email do SAP. Logo, a proposição é verdadeira.

P9: Canais de comunicação trazem os atores a um alinhamento e pode permitir transparência para construir confiança. Prestação de contas, transparência e sustentabilidade são os princípios orientadores da governança corporativa no SAP que é baseada no código alemão de governança corporativa (SAP, 2018b). Os canais de comunicação auxiliam no alcance destes princípios e auxiliam na construção da confiança. O SAP divulga anualmente em seu site principal o nome dos participantes de reuniões do quadro de diretores. O SAP também divulga dados do ano fiscal do ecossistema por meio de um relatório de gestão. Este relatório é submetido ao conselho fiscal e ao auditor para análise de possíveis inconsistências. O resultado desta análise também é divulgado no site principal do SAP. Relatórios trimestrais e semestrais também são divulgados com notícias consideradas relevantes sobre os órgãos diretivos do SAP, sobre a governança corporativa, sobre negociações, além de dados financeiros, sociais e de pessoal. Situações excepcionais são informadas por meio de relatórios especiais divulgados no decorrer do ano. Logo, a proposição é verdadeira para o contexto do SAP.

P10: Canais de comunicação permite propagação da informação. Além do site principal do SAP, o ecossistema divulga produtos e serviços por meio de eventos como o Capital Markets Day realizado em Nova York, a feira CeBIT realizada na Alemanha e a conferência Sapphire Now, realizada na Flórida. Outros canais de comunicação incluem o twitter @sapinvestor, a revista trimestral SAP Investor e um serviço de mensagens de texto. Todos os eventos são divulgados por meio destes canais e todas as apresentações relevantes ocorridas nestes eventos são publicadas no site de relações com investidores. O marketing realizado pelo SAP nestes canais não só

possibilita a propagação da informação, mas fortalece o ecossistema de software e permite a transparência na prestação de contas. Logo, a proposição é verdadeira.

P11: Certificações e padrões de qualidade atuam como barreira de entrada para o ecossistema de software. O SAP é rígido no que se refere a certificações e padrões de qualidade e exige que as empresas parceiras também as adotem. Todos os relatórios de gestão do SAP são gerados em conformidade com a Declaração Financeira Internacional de Práticas Contábeis (IFRS). Dados e informações não financeiras seguem a Norma Internacional de Asseguração da Garantia (ISAE) 3000 e 3410, que são dois padrões de garantia para relatórios de sustentabilidade. Todas as regras e leis do SAP são baseados na lei Sarbanes-Oxley e todos os controles internos e relatórios financeiros seguem o padrão de governança corporativa do NYSE *listed Company Manual*. A certificação SAP significa que os engenheiros do SAP testaram e garantiram que o software do parceiro se integra ao SAP (Huang et al, 2009). Os parceiros certificados pelo SAP são os únicos que são exibidos no site do ecossistema do SAP e parceiros proeminentemente anunciam sua certificação por meio de comunicados à imprensa e em seus sites, como meio de sinalizar sua compatibilidade com o SAP (BasWare, 2005). Portanto, esta proposição para o contexto do SAP é verdadeira.

P12: O exercício de um papel por um ator traz responsabilidade. A hierarquia do SAP é bem definida com responsabilidades, direitos e deveres claramente informados no documento de governança da organização. Cada função possui um conjunto de regras que a detalha. Portanto, para o contexto do SAP esta proposição é verdadeira.

P13: Responsibility promotes trust in relationships. (Responsabilidade promove confiança em relacionamentos). A hierarquia do SAP é composta por uma diretoria executiva que fornece relatórios periódicos sobre a situação do ecossistema (SAP, 2018b). A diretoria executiva é supervisionada por um conselho formado por acionistas e colaboradores do SAP. Comitês responsáveis por coordenar remuneração, finanças, auditoria, tecnologia, riscos e pessoas complementam o quadro hierárquico do SAP que possui uma vasta documentação contendo as responsabilidades e os direitos associados a cada função. Com responsabilidades tão bem fundamentadas e definidas o ator se torna ciente do seu poder de atuação dentro do ecossistema o que aumenta a confiança entre os demais atores. Portanto a proposição é verdadeira.

7.4.3 Modelo Conceitual instanciado para o SAP

Na Figura 24 é mostrada a aplicação do modelo conceitual para o ecossistema do SA2P. Seguindo o mesmo formato das aplicações dos casos anteriores, os elementos específicos do contexto do SAP são apresentados em amarelo. Em branco são apresentados os elementos que também existem no modelo conceitual proposto. Em rosa é representado o elemento *acordo* ‘*agreement*’ que não pôde ser confirmado no SAP por insuficiência de informações dadas pela documentação investigada.

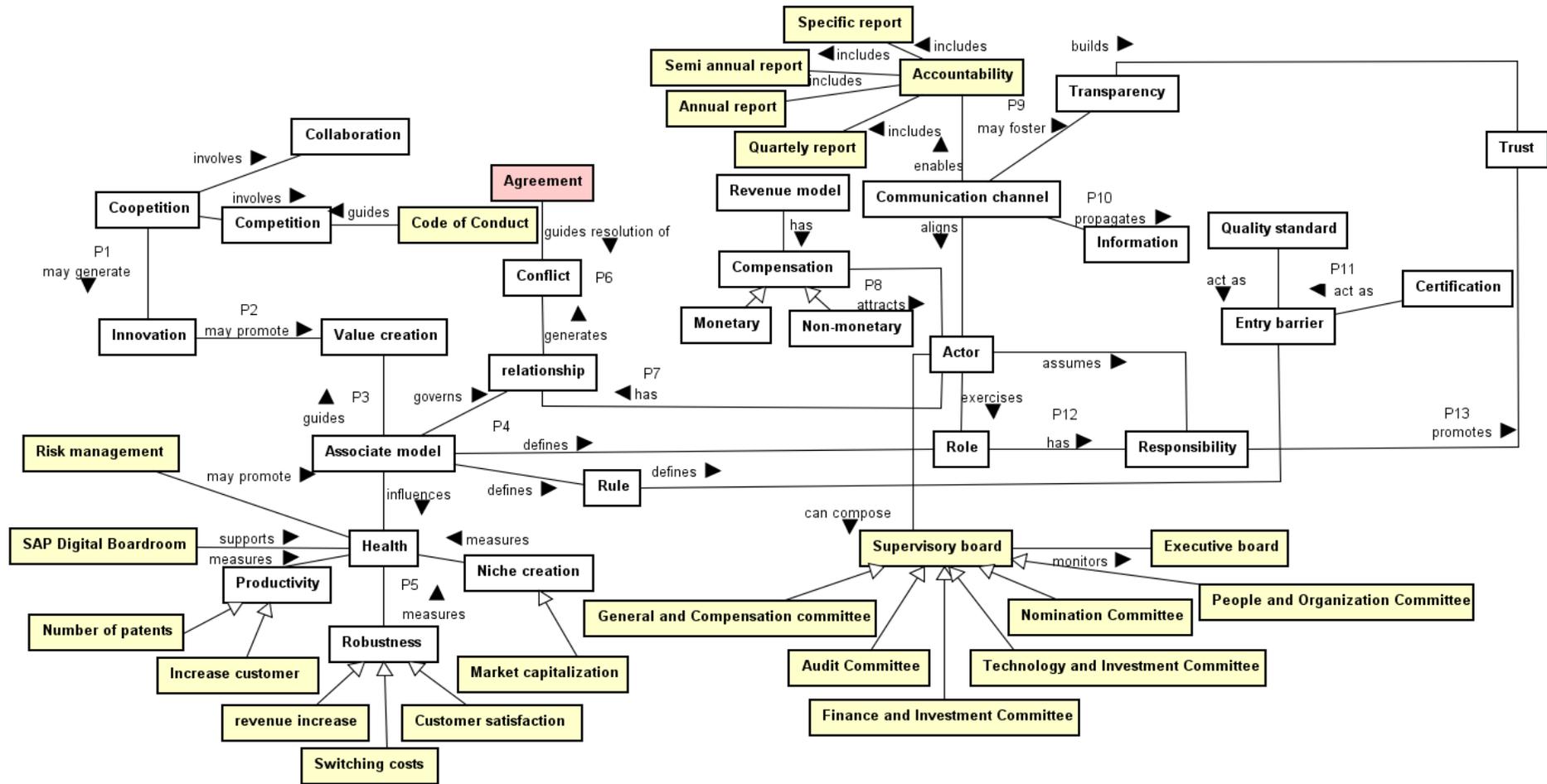


Figura 24: Aplicação do modelo conceitual para o SAP

7.5 Avaliação da implementação por meio dos estudos de caso

Nesta seção é realizada a avaliação da implementação do modelo proposto através dos estudos de caso realizados nos ecossistemas Eclipse, GNOME e SAP. Vale salientar que os estudos de caso discutem como o modelo proposto pode ser instanciado nos ecossistemas estudados. Em particular, cada proposição do modelo foi avaliada se ela é válida ou não para o contexto dos ecossistemas analisados.

A disponibilidade da ampla documentação da governança do Eclipse e do SAP possibilitou que quase todas as proposições fossem confirmadas, com exceção respectivamente das proposições P11 e P6. Em contrapartida, apesar do GNOME disponibilizar informações a respeito da sua governança, a documentação não é tão estruturada. Das 13 proposições do modelo conceitual, apenas seis puderam ser totalmente confirmadas. Cinco proposições não puderam ser instanciadas para este contexto por ausência de insumos que a comprovassem como verdadeiras e três foram confirmadas em parte. Mesmo inferindo que algumas proposições podem ser caracterizadas como falsas para o contexto de um ecossistema *open source* (ex: P11), achou-se prudente não utilizar este termo sem uma evidência explícita que permitisse refutar a proposição para o ecossistema estudado. Na Tabela 18 são apresentados os elementos conceituais identificados em cada caso e a Tabela 19 apresenta uma síntese da aplicação do modelo conceitual para cada ecossistema investigado considerando cada proposição.

Tabela 18: Elementos conceituais de cada estudo de caso

Elemento	Eclipse	Gnome	SAP
Ator (Actor)	C	C	C
Acordo (Agreement)	C	NC	NC
Modelo de associação (Associate model)	C	C	C
Certificação (Certification)	NC	NC	C
Coopetição (Coopetition)	C	C	C
Colaboração (Collaboration)	C	C	C
Competição (Competition)	C	NC	C
Canal de comunicação (Communication channel)	C	C	C

Conflito (Conflict)	C	NC	C
Barreira de entrada (Entry barrier)	C	NC	C
Saúde (Health)	C	C	C
Produtividade (Productivity)	C	C	C
Criação de nicho (Niche creation)	C	C	C
Robustez (Robustness)	C	C	C
Informação (Information)	C	C	C
Inovação (Innovation)	C	C	C
Padrão de qualidade (Quality standard)	NC	NC	C
Relacionamento (Relationship)	C	C	C
Responsabilidade (Responsibility)	C	NC	C
Modelo de receita (Revenue model)	C	C	C
Papel (Role)	C	C	C
Transparência (Transparency)	C	C	C
Confiança (Trust)	C	C	C
Criação de valor (Value creation)	C	C	C

Legenda:

C – Confirmado

NC – Não confirmado para o contexto

Considerando a quantidade de proposições válidas instanciadas, percebe-se uma maior conformidade do modelo conceitual proposto com os contextos do Eclipse e do SAP, que se caracterizam respectivamente como ecossistema de software híbrido e proprietário. A quantidade de proposições que não puderam ser comprovadas no contexto do GNOME, que é um ecossistema de software aberto, indica que outros estudos precisam ser realizados observar como estes ecossistemas se comportam a fim de identificar um padrão na sua dinâmica.

Os pontos comuns para todos os ecossistemas que surgiram a partir dos estudos de caso e que não foram abordados no modelo conceitual precisam de maior

embasamento para serem transformados em elementos ou relacionamentos e posteriormente adicionados ao modelo conceitual proposto.

Tabela 19: Resultado da aplicação do modelo conceitual para os ecossistemas investigados

Proposição	Eclipse	GNOME	SAP
P1: Coopetição pode gerar inovação.	V	PV	V
P2: Inovação pode gerar criação de valor.	V	V	V
P3: Um modelo de associação guia a criação de valor.	V	NC	V
P4: Um modelo de associação define um conjunto de papéis, define barreiras de entrada e governa relacionamentos influenciando a saúde do ecossistema.	V	PV	V
P5: Saúde de ecossistemas de software pode ser mensurada por produtividade, robustez e criação de nicho.	V	V	V
P6: Acordos guiam a resolução de conflitos em relacionamentos em um ecossistema de software.	V	NC	NC
P7: Atores participam em um relacionamento.	V	V	V
P8: Modelos de receita com compensações monetárias e não monetárias atraem atores.	V	PV	V
P9: Canais de comunicação trazem os atores a um alinhamento e pode permitir transparência para construir confiança.	V	V	V
P10: Canais de comunicação permite propagação da informação.	V	V	V
P11: Certificações e padrões de qualidade atuam como barreira de entrada para o ecossistema de software.	NC	NC	V
P12: O exercício de um papel por um ator traz responsabilidade.	V	NC	V
P13: Responsabilidade promove confiança em relacionamentos	V	NC	V

Legenda:

V – Verdadeira

PV – Em Parte verdadeira

NC – Não confirmada para o contexto

Os estudos de caso mostraram a diversidade existente entre cada tipo de ecossistema de software (proprietário, aberto e híbrido). Enquanto para o Eclipse e para o GNOME é estratégico definir um quadro de diretores composto pelos parceiros para gerenciar a governança, para outros ecossistemas de software outras estratégias, como a centralização da governança em um único *keystone*, como no caso do SAP onde os membros do quadro de diretores são apenas colaboradores do SAP, podem ser mais

efetivas por motivos distintos (Ex. quantidade de parceiros, tipo e tamanho do ecossistema, etc.). O mesmo ocorre com a *política antitruste e o código de conduta de competição*. Enquanto estes elementos são relevantes respectivamente para o Eclipse e para o SAP, para o GNOME pode não ser necessário dado o ambiente de baixa competição e alta colaboração que identifica ecossistemas abertos.

Alguns pontos adicionais foram observados ao se investigar o contexto de cada ecossistema de software a partir da documentação disponibilizada que podem ser úteis para futuros estudos:

- **Licença:** Em todos os três ecossistemas de software há um cuidado com relação ao uso de licenças de software. O Eclipse é muito rigoroso quanto a análise de incompatibilidade ou de uso indevido de licenças. Se uma ação de colaboração fere os direitos e as obrigações de Propriedade Intelectual, ela imediatamente é suspensa pelo Eclipse (Eclipse, 2016).

O GNOME também possui um acordo de licença que orienta sobre os direitos e deveres associados ao uso da sua logomarca e dos produtos desenvolvidos no ecossistema¹. A maior restrição é que eles podem ser reproduzidos e distribuídos, mas não podem ser utilizados para propósitos comerciais (GNOME, 2018b). Apesar de haver um processo de revisão de todas as colaborações, na documentação encontrada não há referência a existência de um processo específico de análise rigorosa de compatibilidade de licenças, como há no Eclipse. Também não foram encontrados documentos que especifiquem cada etapa envolvida na revisão das colaborações, portanto não há clareza sobre quais são os critérios utilizados para descartar uma colaboração devido ao uso indevido ou incompatível de licença.

O SAP, assim como o Eclipse e o GNOME também tem enfoque em licenças, patentes e propriedade intelectual (SAP, 2018b). Entretanto, na documentação encontrada não há menção a um processo de legalidade entre licenças, como no Eclipse, e não há um detalhamento sobre como a propriedade intelectual é gerida. É possível que documentos relacionados a estes elementos existam mas não estejam disponíveis ao público externo.

¹ <https://wiki.gnome.org/FoundationBoard/Resources/LicensingAgreement>

As observações oriundas dos estudos sobre licenças a partir dos estudos de caso é que: i) licenças precisam ser compatíveis entre si; ii) licenças precisam ter um conjunto de direitos e deveres bem definidos; iii) licenças precisam ser guiadas por um processo de análise de compatibilidade para evitar a ilegalidade da colaboração; iv) o uso de licenças precisa ser apoiado por políticas de Propriedade Intelectual e de patentes.

- **Governança compartilhada:** Um ponto em comum entre o GNOME e o Eclipse é a governança compartilhada por um quadro de diretores composto por parceiros de diversas organizações e a existência de uma comunidade que nutre o ecossistema por meio da colaboração e inovação. O Eclipse, entretanto, apresenta uma diversidade de comissões que possuem uma divisão profundamente detalhada com relação as responsabilidades de cada uma delas. O SAP, em contrapartida, possui a governança fechada. Todos os membros do quadro de diretores são funcionários do SAP e apenas os acionistas, além do quadro de diretores, pode interferir na governança do ecossistema, o que identifica o SAP como um *keystone*.
- **Gerenciamento de riscos:** Um dos destaques do SAP consiste no grande enfoque ao gerenciamento de riscos (SAP, 2018b). O ecossistema possui um robusto conjunto de regras e instruções para tratar riscos de caráter social, tecnológico e de negócio. Toda a documentação relacionada a riscos segue um padrão e é continuamente avaliada para monitoramento da eficácia do controle aos riscos. Certamente o gerenciamento de riscos é um dos fatores que promove a saúde do ecossistema SAP.

7.6 Síntese do capítulo

Este capítulo apresentou os resultados da avaliação da implementação do modelo conceitual, que foi realizada por meio de aplicações para os ecossistemas Eclipse, GNOME e SAP. Alguns pontos foram observados durante a investigação da documentação dos ecossistemas que foram foco dos estudos de caso. Entre estes pontos destacaram-se a necessidade de gerenciar licenças de software, de definir o tipo de colaboração com relação à governança (compartilhada ou fechada) e de se gerenciar riscos sociais, técnicos e de negócio no ecossistema de software. Entretanto, concluiu-se

que para que novos elementos como *licença* serem agregados ao modelo futuramente, eles devem ser mais profundamente estudados em outros ecossistemas de software.

8 DISCUSSÃO

O modelo conceitual proposto permite que acadêmicos e profissionais obtenham uma visão comum dos elementos envolvidos na governança de ecossistemas de software, bem como compreendam as relações entre eles. O modelo descreve 28 elementos envolvidos na governança de um ecossistema de software. Três desses elementos - "ator", "saúde" e "modelo de associação" - têm um grande número de relacionamentos com outros elementos. Nas seções seguintes, são discutidas as implicações para pesquisa e prática deste trabalho. Finalmente, a pesquisa é comparada com estudos relacionados sobre governança de ecossistemas de software.

8.1 Implicações teóricas

Sob uma perspectiva teórica, os resultados da Revisão Sistemática da Literatura (Capítulo 5), a lista de mecanismos de governança (Tabela 3), o glossário de elementos (Tabela 7) e o modelo conceitual (Figura 15) são contribuições relevantes que aumentam o corpo de conhecimento na área de governança de ecossistemas de software. Os resultados da Revisão Sistemática da Literatura além de fornecer uma visão geral sobre métricas de saúde de ecossistemas de software aponta *gaps* relevantes a serem abordados por pesquisadores da área. A síntese fornecida pela revisão da literatura permitiu uma análise mais aprofundada sobre o papel futuro da governança de ecossistema de software em organizações produtoras de software. Foram identificadas necessidades, que são requisitos que devem ser cumpridos para avançar no campo da governança de ecossistemas de software.

A pesquisa resultou em um conjunto de contribuições de teor teórico. A lista de mecanismos resultante da revisão sistemática da literatura consiste em um importante artefato para que se compreenda quais pontos são passíveis de serem governados em um ecossistema de software. O glossário fornece uma descrição unificada e abrangente dos elementos de governança. O modelo conceitual fornece uma referência sistemática para que os pesquisadores entendam os elementos envolvidos na governança do ecossistema de software. Além disso, as etapas seguidas para construir o modelo conceitual e representadas em UML através de uma estrutura (Figura 14) podem ser usadas pelos pesquisadores para gerar outros modelos conceituais. A partir da estrutura representada na Figura 14 se infere que extraíndo da literatura elementos relacionados a qualquer área, definindo-se proposições com base em evidências primárias e representando-as por meio

de uma notação é possível propor um modelo conceitual que explicará um determinado fenômeno que não necessariamente esteja relacionado com a governança de ecossistemas de software ou com a área de Tecnologia da Informação, mas com qualquer domínio para o qual seja necessário explicar e obter uma visão holística da sua dinâmica.

8.2 Implicações práticas

Em um ecossistema de software, as organizações trabalham em colaboração para permanecerem lucrativas e sobreviverem às mudanças do mercado. Para obter sucesso em suas parcerias, as organizações precisam se unir a um ecossistema de software sem se apropriarem de vantagens que desestabilizem a saúde geral do ecossistema. Ao adotar um modelo de governança, as empresas podem alcançar esse equilíbrio. Portanto, os profissionais podem usar o modelo conceitual como ponto de partida para entender os elementos de governança e seus respectivos relacionamentos. O modelo também pode ser útil para formular estratégias de governança para aumentar a saúde do ecossistema. Por exemplo, o modelo conceitual proposto destaca que modelos híbridos de receita, com fluxos monetários e não monetários, podem atrair atores para o ecossistema (Proposição P8). Com base nessa proposição, as empresas podem aplicar estratégias de governança envolvendo diferentes tipos de modelos de receita para expandir sua rede e aumentar a robustez do ecossistema. A partir da análise do modelo conceitual, os profissionais também podem determinar quais elementos e relacionamentos são relevantes para aumentar a capacidade de crescimento do ecossistema ao longo do tempo.

O escopo do modelo conceitual se limita a representação de elementos e relacionamentos da governança no contexto de ecossistemas de software. O modelo conceitual pode ser usado pelos profissionais da seguinte forma: i) verificar os elementos e relações entre os elementos no modelo conceitual, ii) verificar quais elementos e relações de modelo conceitual existem em seus ecossistemas de software, iii) verificar se o elemento existente/relacionamento atua como facilitador para aumentar a saúde do ecossistema como mostrado no modelo conceitual, iv) se os elementos são facilitadores, definir estratégias para potencializá-los, vi) se os elementos são barreiras para a saúde do ecossistema, definir estratégias para transformá-los em facilitadores. Na Figura 25, é mostrada uma maneira de expandir o modelo para agregar estratégias usando a notação UML. Foi utilizado o elemento de cooperação como exemplo e colocou-se as estratégias “definir políticas” e “definir o processo de verificação de conformidade” para apoiar a cooperação entre os atores. Cada etapa e documentos associados a essas estratégias podem

ser explicados e disponibilizados em um site, por exemplo, de acordo com as características do ecossistema de software.

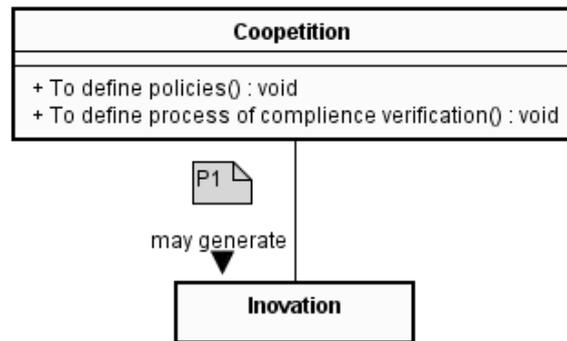


Figura 25: Estratégias para suportar o elemento coopetição

8.3 Trabalhos relacionados

Manikas et al. (2015) propuseram um framework para definir estratégias de tomada de decisão aplicadas à governança de um ecossistema de software. Segundo os autores, o objetivo do framework é analisar as estratégias de tomada de decisão dentro de cinco áreas, agrupando as principais decisões de governança em um ecossistema relacionadas a princípios, interação entre os atores, interação entre software, plataforma e produtos. Para cada área, as decisões são tomadas com base nos seguintes arquétipos: monarquia, coletivo, federal e anarquia. Esses arquétipos indicam se a decisão é tomada apenas pelo *keystone*, se a tomada de decisão é coletiva ou se cada ator do ecossistema toma suas próprias decisões sem a interferência de uma instância superior. A estrutura tem uma linha de raciocínio interessante, mas poderia ser mais bem explorada representando o impacto de cada arquétipo nas áreas de governança.

Christensen et al. (2014) modelaram ecossistemas de software de acordo com o conceito de arquitetura do ecossistema. Eles identificaram os aspectos importantes da estrutura organizacional, tais como: conjuntos de atores e elementos de software, limites do ecossistema, interações e coordenação entre atores e elementos de software. Embora não especificamente voltado para o contexto de governança, o modelo aborda elementos relevantes a serem governados em um ecossistema de software.

Jansen e Cusumano (2013) propuseram um modelo para medir o grau de abertura de uma organização desenvolvedora de software. Esse modelo lista as áreas em que a organização pode ser mais aberta de acordo com três pontos de vista: estratégico, tático e

operacional. Uma dessas áreas é a governança, onde as diferentes opções para a abertura da governança dos três pontos de vista são discutidas.

Baars e Jansen (2012) definiram uma estrutura para a análise da governança de ecossistemas de software. O objetivo é ajudar as empresas a obter uma vantagem sobre os parceiros, na medida em que possam analisar e melhorar seu ecossistema de maneira estruturada, levando a um melhor desempenho e saúde. A estrutura aborda as questões de responsabilidade, abertura, compartilhamento de conhecimento e abertura nos ecossistemas de software, bem como a medição da saúde do ecossistema. O framework pode ser usado para descrever, avaliar e comparar políticas de governança de ecossistemas de software. Além disso, o próprio trabalho indica a falta de validação do modelo pelos especialistas como uma limitação.

Fontão et al. (2018) propôs um processo de governança para apoiar Ecossistemas de Software Móveis (MECO), com foco em mecanismos para reutilizar conteúdo, como código e modelos, a partir de repositórios. Eles identificaram as principais atividades de governança, os artefatos e o papel do contexto do ecossistema de software móvel e validaram o processo proposto com gerentes. O processo de governança tem oito atividades: especificar a plataforma, definir diretrizes do MECO, fornecer diretrizes de interface, fornecer ferramentas de desenvolvimento, fornecer site de gerenciamento de aplicativos, definir critérios de certificação de aplicativos, criar políticas de incentivo e fornecer repositório de aplicativos. Essas atividades têm um conjunto de documentos relacionados que precisam ser gerados durante a execução do processo de governança.

O modelo proposto aqui nesta tese difere dos mencionados acima porque tem uma perspectiva conceitual e tem uma forte base teórica. Foi adotado o método de Design Science para construir o modelo conceitual a partir de uma visão dedutiva. Além disso, o modelo foi validado com especialistas e instanciado por meio de estudos de caso para ecossistemas do tipo híbrido, *open source* e proprietário. Esses aspectos são um diferencial desta pesquisa. O modelo ajuda pesquisadores e profissionais envolvidos no domínio de ecossistemas de software a entender como a governança funciona nesse contexto. O fato de o modelo ter sido descrito usando UML também consiste em uma vantagem, já que essa é uma notação amplamente conhecida e facilmente compreendida até mesmo por pessoas que nunca a estudaram. Elementos conceituais foram identificados em outros estudos no campo dos ecossistemas de software, mas não são formalmente definidos e analisados de maneira abrangente. Os elementos conceituais definidos na

Tabela 7 podem auxiliar a sanar esta lacuna, servindo como um parâmetro inicial para a formalização dos elementos da governança de ecossistemas de software.

8.4 Síntese do capítulo

Este capítulo discutiu as implicações teóricas (8.1) e práticas (8.2) deste estudo e apresentou trabalhos que possuem semelhança com a pesquisa descrita nesta tese (8.3). As implicações teóricas indicam que este estudo possui relevantes contribuições como a lista de mecanismos de governança, o glossário de elementos de governança, o modelo conceitual e uma estrutura para que outros pesquisadores possam criar seus próprios modelos conceituais a partir deste estudo. Todos estes artefatos são úteis por ampliar o corpo de conhecimento da área e por servirem de insumos para que novas pesquisas sobre governança de ecossistemas de software sejam geradas. Com relação à implicação prática, o modelo conceitual consiste em uma importante contribuição por ser um ponto de partida para que os praticantes possam compreender a dinâmica de governança em seus contextos, formular guias para governar os seus elementos e assim aumentar a saúde do ecossistema de software.

9 CONCLUSÃO

Plataformas de software como Salesforce, AppExchange, Netsuite, SuiteApp ou SAP Commercial Platform estão mudando a maneira como software e serviços são distribuídos, consumidos e comprados (Weiblen et al, 2012). As relações entre os parceiros nesses contextos precisam ser governadas para viabilizar a sustentabilidade. Esta tese propôs um modelo conceitual para governança de ecossistemas de software construído através do método de Design Science Research. O estudo tem caráter dedutivo (Sjøberg, 2008), pois os elementos do modelo conceitual emergiram de uma Revisão Sistemática da Literatura, previamente realizada. O modelo é composto por proposições testáveis (ou seja, relações entre elementos), que derivaram de estudos primários. Além disso, o modelo conceitual foi avaliado por meio de uma pesquisa com dezessete especialistas. Ele também foi instanciado através de entrevistas com profissionais que atuam em diferentes ecossistemas e foram realizados estudos de caso para implementar o modelo conceitual em um ecossistema híbrido (Eclipse), um aberto (GNOME) e outro fechado (SAP). Estes estudos empíricos permitiram uma reflexão sobre a relevância do modelo para ecossistemas na prática e sobre quais pontos devem ser alterados para melhorar sua aplicabilidade.

9.1 Limitações e ameaças à validade

Os métodos qualitativos que apoiaram o ciclo de Design Science Research (survey com especialistas, entrevistas com praticantes e estudos de caso) apresentaram diversas limitações. Estes métodos foram conduzidos sob uma perspectiva epistemológica, interpretativista e não positivista (Flick, 2009). Foram baseados epistemologicamente porque procurou-se entender a opinião dos especialistas e a vivência dos praticantes considerando suas características individuais. Foi interpretativista porque acredita-se que diferentes percepções sobre o tema investigado resultam em diferentes modelos conceituais para governança de ecossistemas de software. Não é positivista porque não há apenas uma explicação para o fenômeno estudado, que é a governança de ecossistemas de software. Prova disso é que algumas hipóteses construídas foram refutadas pelos especialistas e outras não puderam ser confirmadas por todos os praticantes e pelos estudos de caso.

Segundo Flick (2009), o paradigma interpretativo preconiza que o pesquisador e o fenômeno em estudo são interativos, ou seja, da mesma forma que a visão particular de

determinado pesquisador afeta a descrição de um fenômeno, o próprio fenômeno afeta o pesquisador de modo que a análise do fenômeno sofre algumas modificações devido a essa influência. Portanto, uma limitação do estudo é o viés inerente à interpretação das evidências extraídas para compor as proposições. No entanto, esse viés foi mitigado pela execução de vários ciclos de revisão das proposições e do modelo conceitual pelo orientador e pela execução do survey com especialistas.

Outra limitação do estudo é o número limitado de especialistas e praticantes que participaram do survey e das entrevistas para avaliar o modelo conceitual. Entretanto, a qualidade dos dados coletados atenua o problema do baixo número de participantes.

A escolha dos especialistas para participar do survey também consistiu em uma limitação. As características dos especialistas e dos profissionais influenciam sua visão sobre o fenômeno da governança. Especialistas em ecossistemas abertos tendem a ver a governança de forma mais flexível, com menos regras e com tomadas de decisão compartilhadas por partes interessadas. Especialistas de ecossistemas proprietários têm uma visão mais rígida focada no poder exercido pelo *keystone*. As duas visões geram modelos de governança antagônicos. Esta diferença é perceptível ao se observar o modelo instanciado para o SAP e o modelo instanciado para o GNOME. O modelo do GNOME é simples e pouco restritivo, enquanto o do SAP é baseado em uma hierarquia rígida com altas barreiras de entrada. A maioria dos especialistas tem uma perspectiva baseada em ecossistemas proprietários e avaliou o modelo em favor dessa dimensão. Em futuros estudos será possível gerar camadas para o modelo conceitual a partir dessas duas perspectivas diferentes. Desta forma, será possível aumentar o poder de generalização da pesquisa, uma vez que as peculiaridades de cada tipo de ecossistema seriam abordadas mitigando esta limitação.

Alguns especialistas relataram que o tempo para completar a pesquisa foi superior ao estimado, que foi de vinte minutos. Isso adicionou riscos relacionados à maturação e mortalidade das respostas. Segundo Yu e Ohlund (2010), a maturação refere-se aos processos em que os sujeitos agem em função da passagem do tempo. A mortalidade consiste na perda de sujeitos (Yu e Ohlund, 2010), ou seja, que o número de sujeitos que concluíram o questionário é menor que o número de sujeitos que passaram a respondê-lo. Apenas três especialistas não responderam a todas as perguntas do survey. Mais especificamente, eles não responderam à pergunta sobre o que é governança de ecossistemas de software. Entretanto, acredita-se que as limitações relacionadas à

maturação e mortalidade não afetaram o estudo com base na alta quantidade e qualidade de respostas válidas. Com relação a entrevistas com os praticantes, buscou-se elaborar um protocolo simples com perguntas semi-estruturadas que foram sendo ajustadas de acordo com o decorrer de cada entrevista. As entrevistas tiveram a duração média de meia hora para não prejudicar a maturação das respostas.

Abaixo são discutidas as principais ameaças à validade e confiabilidade deste estudo, de acordo com Merriam (2009) e Wohlin et al. (2012).

- **Validade interna:** Consiste na equivalência dos resultados obtidos com a realidade ou no grau de credibilidade do estudo. Uma das estratégias propostas por Merriam (2009) para aumentar a validade interna da pesquisa consiste em utilizar múltiplas técnicas de coleta para permitir a triangulação de dados. Nesta pesquisa os dados foram coletados por meio de questionários estruturados, de entrevistas semiestruturadas e de documentação dos ecossistemas investigados. Além de se diversificar os instrumentos de coleta de dados, buscou-se também diversificar o perfil dos participantes do estudo, que agregou pesquisadores e praticantes de variados ecossistemas de software com funções respectivamente de gerente de vendas, desenvolvedor e gerente de projetos.
- **Validade externa:** refere-se à capacidade de generalização do estudo para outro contexto. Outro termo utilizado para caracterizar a validade externa consiste na transferibilidade. Merriam (2009) afirma que nem sempre o pesquisador conhece as características do ambiente para o qual os resultados da pesquisa serão transferidos, mas os que irão aplicar estes resultados conhecem. Portanto, uma estratégia mencionada por Merriam (2009) para aumentar a validade externa consiste em prover uma rica descrição dos dados para tornar a transferibilidade possível. Assim, cabe ao interessado na pesquisa decidir se ela é ou não aplicável para o seu contexto. Para diminuir a ameaça à validade externa buscou-se descrever com detalhes as circunstâncias sob as quais a pesquisa foi validada e instanciada. Outra estratégia proposta por Merriam (2009) consiste na definição de hipóteses i) que descrevam situações específicas, ii) que ofereçam aos praticantes a opção de escolher quais delas se adequam ao seu contexto e iii) que possam ser monitoradas e avaliadas com o propósito de orientar a tomada de decisão. As proposições formuladas nesta pesquisa servem a estes propósitos visto que podem ser selecionadas de acordo com o tipo de ecossistema do praticante e

trabalhadas para guiar a tomada de decisão sobre a governança e saúde do ecossistema de software.

- **Confiabilidade:** refere-se capacidade de repetição ou replicação dos resultados, ou seja, de que de que o mesmo resultado seria encontrado ao se refazer o estudo. Merriam (2009) afirma que uma forma de diminuir a ameaça à confiabilidade consiste em realizar ‘trilhas de auditoria’, ou seja detalhar bem como os dados foram coletados e analisados, quais foram os métodos de pesquisa utilizados, quais decisões foram tomadas no processo de investigação e como se chegou aos resultados. A condução do ciclo de Design Science Research, a documentação de todos os passos da pesquisa, a realização de teste piloto e a inserção de informações sobre a pesquisa em um site auxiliaram no aumento da confiabilidade deste estudo.

9.2 Trabalhos futuros

Novos refinamentos do modelo conceitual são necessários para que o mesmo atenda as especificidades de ecossistemas de diferentes tipos tais como, proprietários, abertos e híbridos. Ainda se torna necessário validar os modelos oriundos dos estudos de casos com atores atuantes na governança de cada ecossistema investigado (GNOME, Eclipse e SAP). Além disso, pontos adicionais identificados nos estudos de caso (ex: licenças, patentes e Propriedade Intelectual) podem ser passíveis de inclusão no modelo conceitual, mas para tal faz-se necessário realizar um estudo aprofundado sobre estes aspectos. Para expandir o campo da governança de ecossistemas de software, é proposta a seguinte agenda de pesquisa:

- **Incorporar os princípios de governança corporativa ao modelo conceitual de governança para ecossistemas de software:** os princípios da governança corporativa inspiraram estudos sobre governança em diferentes campos, como Arquitetura Orientada a Serviços e Tecnologia da Informação. A governança corporativa envolve conceitos relacionados à transparência, direitos, interesses das partes interessadas e ética. A governança do SAP, por exemplo, é fortemente baseada na governança corporativa. A análise desses princípios no contexto de ecossistemas de software pode ser útil para ampliar o corpo de conhecimento da área.
- **Definir um Vocabulário Comum na Governança de Ecossistemas de Software:** O número de publicações neste domínio enfatiza que a governança de

ecossistemas de software está amadurecendo. Uma quantidade cada vez maior de trabalhos se posiciona sobre definições dos conceitos centrais de ecossistemas de software: saúde (Manikas e Hansen, 2013b), governança (este trabalho), ecossistemas *open source* (Silva et al, 2014) e qualidade em ecossistemas de software (Axelsson e Skoglund, 2016), cada um desses conceitos está se estabelecendo como um termo de discursão na área de ecossistema de software. Foi identificado que vários estudos adotam termos relacionados, como gerenciamento e orquestração, para se referir a mecanismos de governança. Portanto, é sugerida a necessidade de estabelecer um glossário comum e um quadro conceitual que colete essas definições sobre governança de ecossistemas.

- **Definir Orientações Práticas de Governança:** Embora exista um vasto conhecimento sobre governança de ecossistemas de software, é difícil para os profissionais extrair orientação práticas e estratégicas das obras em estudo. Existe uma necessidade de conhecimentos mais adotáveis e práticos para os profissionais. Estudos relevantes para profissionais interessados na criação de painéis de saúde de ecossistemas incluem a coleta de informações sobre o progresso de determinados ecossistemas por meio do GitHub (Goeminne e Mens, 2013). Ferramentas semelhantes ao GitHub podem constituir mecanismos e instrumentos de avaliação maduros para grandes ecossistemas de software *open source* e proprietários.
- **Analisar a Interação entre Mecanismos de Governança e Métricas de Saúde:** Ao selecionar métricas de saúde adequadas, os atores podem governar o ecossistema rumo à um caminho sustentável. Continua a existir um desafio quanto à forma de implementar a governança para fomentar a inovação e incentivar comportamentos autônomos para a diversidade, sem prejudicar a qualidade do software e a prestação de contas das ações dos atores (Jansen et al, 2012). A tensão entre controle e autonomia deve ser devidamente equilibrada. Entender como a implementação de mecanismos de governança específicos afeta o sucesso dos ecossistemas e a plataforma empresarial subjacente é um problema interessante para os estudiosos da área.
- **Entender a Governança de Ecossistemas de Desenvolvedores:** O impacto dos desenvolvedores e nichos de mercado nos ecossistemas são ampliados pelo sucesso do ecossistema. Exemplos como Farmville para Facebook e Angry Birds para iOS ilustram como os ecossistemas crescem imensamente através do sucesso

de seus constituintes. Os desenvolvedores são o ponto de partida para qualquer ecossistema de software. Isto justifica o recente aumento de interesse em ecossistemas de desenvolvedores. Há uma necessidade de entender melhor as expectativas e comportamentos dos desenvolvedores (Yu et al, 2007). Barreiras à entrada e formas de adesão são os fatores que requerem pesquisa adicional. Os orquestradores devem entender as motivações e expectativas dos desenvolvedores para adotar mecanismos de governança apropriados.

- **Estudar a Governança em Ecossistemas de Software *Open Source*:** Os ecossistemas *open source* apresentam propriedades diferentes dos ecossistemas proprietários mais tradicionais. A abertura de uma plataforma permeia todos os aspectos de um ecossistema, seja sobre a propriedade do código ou sobre mecanismos em torno de ferramentas de suporte. Essas questões de abertura também desempenham um papel na arquitetura de uma plataforma: sem uma arquitetura de plataforma aberta não é possível estendê-la. Na revisão sistemática da literatura não foi encontrado nenhum estudo que apresentasse uma análise comparativa dos mecanismos de governança empregados pelos ecossistemas *open source* versus ecossistemas proprietários. Esta é uma linha de pesquisa promissora.
- **Compreender as Interações entre os Ecossistemas:** Mesmo a boa governança pode levar ao desaparecimento de um ecossistema devido a fatores externos. Ao olhar para a governança e saúde do ecossistema Symbian em 2007, teria sido difícil prever sua morte. Pode-se especular sobre o seu apoio à Nokia e sobre falhas fundamentais no modelo de negócios da Symbian. No entanto, é difícil ignorar o infortúnio iminente que veio do iPhone após 2007: sua alta taxa de adoção e tecnologia superior simplesmente explodiu o ecossistema Symbian. O desafio para a pesquisa de governança na próxima década será analisar e compreender a interação entre grandes ecossistemas. Ecossistemas antigos ou novos podem ser destruídos ou crescer exponencialmente através do funcionamento de outros ecossistemas. Deve-se, portanto, desenvolver ferramentas de governança e práticas de gestão que se concentrem na robustez dos ecossistemas de software para prepara-los para sobreviver em tais adversidades.

9.3 Síntese do capítulo

Este capítulo apresentou as limitações do estudo, as ameaças a validade com base em Merriam (2009) e Wohlin et al. (2012) e os trabalhos futuros. As principais limitações da pesquisa consistiram no viés inerente à autora desta tese com relação a interpretação das evidências extraídas para compor as proposições e dos dados obtidos a partir dos métodos qualitativos conduzidos. O número de participantes do survey e das entrevistas também consistiu em uma limitação.

Conclui-se esta tese afirmando que a dinâmica de governança de ecossistemas de software é composta por um conjunto de elementos que se comportam de modo diverso dependendo de como são gerenciados, do tipo e tamanho do ecossistema, dos papéis que possuem e das regras que os regem. Além disso, ressalta-se que nesta dinâmica governança e saúde são temas indissociáveis visto que a saúde está diretamente relacionada com a capacidade de sustentabilidade e de evolução do ecossistema de software.

REFERÊNCIAS

- Aarnoutse, F., Renes C., Snijders, R., and Jansen, S. 2014. The Reality of an Associate Model: Comparing Partner Activity in the Eclipse Ecosystem. ECSAW 2014: Proceedings of the 2014 European Conference on Software Architecture Workshops, ACM, New York, NY, USA, pp. 8-6.
- Alami, D., Rodríguez, M., & Jansen, S. 2015. Relating health to platform success: Exploring three e-commerce ecosystems. In Proceedings of the 2015 European Conference on Software Architecture Workshops (p. 43). ACM.
- Albert. B.E., Santos, R.P e Werner, C. 2013. Software ecosystems governance to enable IT architecture based on software asset management. In Proceedings of the 7th IEEE International Conference on Digital Ecosystems and Technologies (DEST'13), pp.55-60.
- Allen, G.N. and March, S.T. 2012. A research note on representing part-whole relations in conceptual modeling', MIS Quarterly, Vol. 36, No. 3, pp.945–964.
- Alves, A, M., Pessôa, M. 2010. Brazilian public software: beyond sharing. MEDES'10. International Conference on Management of Emergent Digital EcoSystems.
- Alves, C., Oliveira, J., and Jansen, S. 2017a. Software Ecosystems Governance-A Systematic Literature Review and Research Agenda. In ICEIS 2017: Proceedings of the 19th International Conference on Enterprise Information Systems, Vol. 3, pp. 26-29.
- Alves, C., Oliveira, J., and Jansen, S. 2017b. Understanding Governance Mechanisms and Health in Software Ecosystems: A Systematic Literature Review.' In ICEIS 2017: proceedings of In International Conference on Enterprise Information Systems, pp. 517-542. Springer, Cham.
- Axelsson, J., and Skoglund, M. 2016. Quality assurance in software ecosystems: A systematic literature mapping and research agenda. Journal of Systems and Software, Volume 114, pp. 69-81.
- Baars, A., e Jansen, S. 2012. A framework for software ecosystem governance. In Proceedings of the Third International Conference on Software Business (ICSOB'12), pp.168-180. 2012.
- Bachmann, R. 2001. Trust, Power and Control in Trans-Organizational Relations. Organization Studies, 22 (2), 337-365.
- Barbosa, O., Alves, C. 2011. A Systematic Mapping Study on Software Ecosystems. In proceedings of the Third International Workshop on Software Ecosystems (IWecossistema de software'11).
- Barbosa, O., Santos, R., Alves, C., Werner, C., Jansen, S. 2013. A Systematic Mapping Study on Software Ecosystems through a Three-dimensional Perspective. In Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry, S. Jansen, M. Cusumano, S. Brinkkemper (eds.) UK, Edward Elgar Publishing, 59-81.
- BasWare. 2005. Basware achieves SAP certification, http://www.basware.com/News_and_Events/News/Pages/BasWareAchievesSAPCertification.aspx.

- Baxter, P. e Jack, S. 2008. Qualitative case study methodology: Study design and implementation for novice researchers. *The qualitative report*, 13(4), 544-559.
- Bird, F. 2001. Good governance: A philosophical discussion of the responsibilities and practices of organizational governors. *Canadian Journal of Administrative Sciences/Revue Canadienne des Sciences de l'Administration*, 18(4), 298-312.
- Bosch Sijtsema, P., e Bosch, J. 2015. User Involvement throughout the Innovation Process in High Tech Industries. *Journal of Product Innovation Management*, 32(5), 793-807.
- Bosch, J. 2009. From software product lines to software ecosystems. In *Proceedings of the 13th International Software Product Line Conference (SPLC'09)*. Carnegie Mellon University, Pittsburgh, PA, USA. 111-119.
- Bosch, J. 2012. Software ecosystems: Taking software development beyond the boundaries of the organization. *Journal of Systems and Software*, Volume, 85, Issue, 7, pp. 1453-1454.
- Bosch, J., & Bosch-Sijtsema, P. (2014, June). ESAO: A holistic ecosystem-driven analysis model. In *International Conference of Software Business* (pp. 179-193). Springer, Cham.
- Brereton, P., Kitchenham, B., Budgen, D., Turner, M., Khalil, M. 2007. Lesson from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, Volume 80, pp. 571–583.
- Brhel, M., Meth, H., and Maedcher, A. 2015. Exploring principles of user-centered agile software development: A literature review. *Information and Software Technology*, Volume 61, 163-181.
- Brown, W. A., Moore, G., & Tegan, W. 2006. *SOA Governance-IBM's approach*. Somers, NY.
- Burkard, C., Widjaja, T., e Buxmann, P. 2012. Software Ecosystems. *Business & Information Systems Engineering*, volume 4, Issue. 1, pp. 41-44.
- Cardoso, J.L., Barbin, F. A., Silva Filho, O.S. 2013. The public software ecosystem: exploratory survey. In *Proceedings of the Fifth International Conference on Management of Emergent Digital EcoSystems (MEDES '13)*, ACM. 289-296.
- Carvalho, I., Campos, F., Braga, R., David, J. M. N., Stroelle, V., & Araújo, M. A. 2017. HEAL ME: an architecture for health software ecosystem evaluation. In *Proceedings of the Joint 5th International Workshop on Software Engineering for Systems-of-Systems and 11th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems* (pp. 59-65). IEEE Press.
- Ceccagnoli, M., Forman, C., Huang P. and Wu, D.J. 2012. Cocreation of Value in a Platform Ecosystem: The Case of Enterprise Software. *MIS Quarterly*, 36 (1), 263-290.
- Che, M., Perry, D.E. 2014. Architectural Design Decisions in Open Software Development: A Transition to Software Ecosystems. In *ASEC: Proceedings of the 23rd Australian Software Engineering Conference*, pp.58-61.
- Christensen, H.B., Hansen, K.M., Kyng, M., and Manikas, K. (2014). Analysis and design of software ecosystem architectures towards the 4telemedicine ecosystem. *Information and Software Technology*, Vol. 56 No.11, pp. 1476-1492.
- Christensen, H.B., Hansen, K.M., Kyng, M., Manikas, K. 2014. Analysis and design of software ecosystem architectures towards the 4telemedicine ecosystem. *Information and Software Technology*, 0(0).

- Cross, N. 2001. Design/science/research: developing a discipline. In Fifth Asian Design Conference: International Symposium on Design Science, Su Jeong Dang Printing Company, Seoul, Korea.
- Cruzes D. e Dybå, T. 2011. Recommended steps for thematic synthesis in software engineering. In Proceedings of International Symposium on Empirical Software Engineering and Measurement (ESEM'11), pp. 275-284.
- Den Hartigh, E., Tol, M., and Visscher, W. 2006. The health measurement of a business ecosystem.' ECCON 2006: Proceedings of the European Network on Chaos and Complexity Research and Management Practice Meeting, pp. 1-39.
- Den Hartigh, E., Visscher, W., Tol, M., and Salas, A. J. 2013. Measuring the health of a business ecosystem, in Jansen, S. et al (Eds.), Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry, Edward Elgar Pub, pp. 221–246.
- Dhungana, D., Groher, I., Schludermann, E., Biffel, S. Software ecosystems vs. natural ecosystems: learning from the ingenious mind of nature. In Proceedings of the Fourth European Conference on Software Architecture: Companion Volume (ECSA '10), Carlos E. Cuesta (Ed.). ACM, New York, NY, USA. 96-102. 2010.
- dos Santos, R. P., e Werner, C. 2011. Treating business dimension in software ecosystems. In Proceedings of the International Conference on Management of Emergent Digital EcoSystems (pp. 197-201). ACM.
- Earl, M. J. 1994. Experiences in strategic information systems planning, MIS Quarterly, Vol. 17, No. 1, 1-24.
- Eckhardt, E., Kaats, E., Jansen, S., Alves, C. 2014. The Merits of a Meritocracy in Open Source Software Ecosystems. In Proceedings of European Conference on Software Architecture Workshops (ECSAW '14), ACM, pp.7.
- Eclipse Foundation. 2002. Antitrust policy. https://eclipse.org/org/documents/Eclipse%20ANTITRUST%202003_11_10%20Final.pdf
- Eclipse Foundation. 2008. Recognizing Project Contributions Policy. Disponível em http://www.eclipse.org/org/documents/Recognizing_Project_Contributions_Policy_2008.pdf. Acesso em 30 de agosto de 2016.
- Eclipse Foundation. 2008. Recognizing Project Contributions Policy. Disponível em http://www.eclipse.org/org/documents/Recognizing_Project_Contributions_Policy_2008.pdf.
- Eclipse Foundation. 2010. Membership Agreement. Disponível em http://www.eclipse.org/org/documents/Eclipse%20MEMBERSHIP%20AGMT%202010_01_05%20Final.pdf. Acesso em 30 de agosto de 2016.
- Eclipse Foundation. 2015. Bylaws of Eclipse Foundation. Disponível em http://www.eclipse.org/org/documents/Eclipse%20BYLAWS%202011_08_15%20Final.pdf. Acesso em 05 de setembro de 2016.
- Eclipse Foundation. 2016. Eclipse IP Policy. Disponível em http://www.eclipse.org/org/documents/Eclipse_IP_Policy.pdf. Acesso em 30 de agosto de 2016.
- Flick, U. 2009. An Introduction to Qualitative Research, 4 nd ed., S. Netz., Sage Publications, London.

- Fontão, A., Estácio, B., Fernandes, J., dos Santos, R. P., & Dias-Neto, A. C. 2018. Which factors affect the evangelist's support during training sessions in mobile software ecosystems?. In Proceedings of the 12th European Conference on Software Architecture: Companion Proceedings (p. 22). ACM.
- Fotrousi, F., Fricker, Samuel A., Fiedler, M., Le-Gall, F.: KPIs for software ecosystems: a systematic mapping study. In: Lassenius, C., Smolander, K. (eds.) ICSOB 2014. LNBP, vol. 182, pp. 194–211. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08738-2_14
- Fox, J. 2007. The uncertain relationship between transparency and accountability. *Development in practice*, 17(4-5), 663-671.
- Franco-Bedoya, O., Ameller, D., Costal D., and Franch, X. 2014. Queso: A quality model for open source software ecosystems. In Proceedings of the 9th International Conference on Software Engineering and Applications (ICSOFTEA'14), pp. 209-221.
- Fricker, S. 2009. Specification and Analysis of Requirements Negotiation Strategy in Software Ecosystems. In proceedings of Workshop on Software Ecosystems. 19-33.
- Fricker, S. 2010. Requirements Value Chains: Stakeholder Management and Requirements Engineering in Software Ecosystems. In proceedings of Requirements Engineering: Foundation for Software Quality (REFSQ'10), volume 6182. 60-66.
- Garcia, R., e R. Calantone. 2002. A Critical Look at Technological Innovation Typology and Innovativeness Terminology: A Literature Review," *The Journal of Product Innovation Management*, vol. 19, pp. 110 – 132.
- Gawer, A., & Henderson, R. 2007. Platform owner entry and innovation in complementary markets: Evidence from Intel. *Journal of Economics & Management Strategy*, 16(1), 1-34.
- Geerts, G. L. 2011. A design science research methodology and its application to accounting information systems research. *International Journal of Accounting Information Systems*, 12(2), 142-151.
- Gillies, A. 2011. *Software quality: theory and management*. 3rd edition. Lulu. com.
- GNOME Bylaws. 2002. Bylaws of GNOME Foundation. <https://www.gnome.org/foundation/governance/attachment/bylaws-2/>.
- GNOME. Annual Report. 2017. <https://www.gnome.org/wp-content/uploads/2018/08/GAR2017-18-07-2018-final-web.pdf>.
- GNOME. Contribution Process. 2018a. <https://wiki.gnome.org/Community/Contribute/Process>
- GNOME. Foundation Board. 2018b. <https://wiki.gnome.org/FoundationBoard/Minutes>
- GNOME.LicensingAgreement.2018c.<https://wiki.gnome.org/FoundationBoard/Resources/LicensingAgreement>.
- Gorman, A., & Fischer, G. 2009. Toward an analytic framework for understanding and fostering peer-support communities in using and evolving software products. In Proceedings of the fourth international conference on Communities and technologies (pp. 1-10). ACM.
- Gregor, S., e Jones, D. 2007. The anatomy of a design theory. *Journal of the Association for Information Systems*, 8(5), 312.

- Gronroos, C. 2008. Service Logic Revisited: Who Creates Value? And Who Co-Creates? *European Business Review* 20(4): 298–314.
- Hansen, G., Alves C., e Bosch, J. 2014. Special Issue editorial: Understanding Software Ecosystems. *Information and Software Technology*, Volume 56.
- Hansen, G., Dyba, T.: Theoretical foundations of software ecosystems. *International Workshop on Software Ecosystems* (2012)
- Hevner, A. R. 2007. A three cycle view of design science research. *Scandinavian journal of information systems*, 19(2), 4.
- Hevner, A., e Chatterjee, S. 2010. *Design science research in information systems*. 9-22. Springer US.
- Hojaji, F., & Shirazi, M. R. A. 2010a. Developing a more comprehensive and expressive SOA governance framework. In *The 2nd IEEE International Conference on Information Management and Engineering (ICIME)*. 563-567. IEEE.
- Hojaji, F., & Shirazi, M. R. A. 2010b. A comprehensive SOA governance framework based on COBIT. In *6th World Congress on Services* (pp. 407-414). IEEE.
- Holmström, J., Ketokivi, M., e Hameri, A. P. 2009. Bridging practice and theory: A design science approach. *Decision Sciences*, 40(1), 65-87.
- Hoogervorst, J. A. P. 2009. Corporate Governance. In: *Enterprise Governance and Enterprise Engineering*. Springer. 153–188.
- Huang, P., Ceccagnoli, M., Forman, C., & Wu, D. J. (2009). When do ISVs join a platform ecosystem? Evidence from the enterprise software industry. *ICIS 2009 Proceedings*, 161.
- Huang, P., Ceccagnoli, M., Forman, C., Wu, D.J.: 2013 Appropriability mechanisms and the platform partnership decision: Evidence from enterprise software. *Management Science*, Vol 59, Issue 1.
- Hurni, T., e Huber, T. 2014. The interplay of power and trust in platform ecosystems of the enterprise application software industry. *Twenty ecosystem de software and European Conference on Information Systems*.
- Hyrnsalmi, S., Weal Seppanen, M., Nokkala, T., Suominen, A., and Jarvi, A. 2015. Wealthy, Healthy and/or Happy — What does ‘Ecosystem Health’ Stand for? J.M. Fernandes et al. (Eds.): *ICSOB 2015*, Springer, Heidelberg, pp. 272–287.
- Iansiti, M., and Levien, R. 2004. *Keystones and Dominators: Framing Operating and Technology Strategy in a Business Ecosystem*. Harvard Business School, Working Paper, 03-061.
- IBGC. 2013. *Origem da boa Governança*. Instituto Brasileiro de Governança Corporativa - IBGC. Disponível em: <http://www.ibgc.org.br/index.php/governanca/origens-da-governanca>. Último acesso: Dez/2016.
- Iivari, J., e Venable, J. 2007. Action research and design science research—seemingly similar but decisively dissimilar. In *17th European Conference on Information Systems* (pp. 1-13). 2009. in *BPM Governance. 1st International Workshop on BPM Governance*. Queensland University of Technology. *International Workshop on Software Ecosystems*, 34–48.
- ISACA. 2007. *Cobit 4.1 - Portuguese*. ISACA / IT Governance Institute.

- Jansen, S. Brinkkemper, S., Souer, J. e Luinenburg, L. 2012. Shades of gray: Opening up a software producing organization with the open software enterprise model. *Journal of System and Software*. Volume 85, Issue 7, pp.1495-1510.
- Jansen, S., Brinkkemper, S., Finkelstein, A. 2009. Business Network Management as a Survival Strategy: A Tale of Two Software Ecosystems. In *Proceedings of the 4th International Workshop on Software Ecosystems (IWecosistema de software '12)*.
- Jansen, S., e Cusumano, M. 2012. Defining Software Ecosystems: A Survey of Software Platforms and Business Network Governance. In *Proceedings of the 4th International Workshop on Software Ecosystems (IWecosistema de software '12)*. 41–58.
- Jansen, S.: 2014. Measuring the health of open source software ecosystems: Beyond the scope of project health. *Information and Software Technology*, Volume, 56, Issue 11, pp. 1508–1519.
- Järvinen, P. 2007. Action research is similar to design science. *Quality & Quantity*, 41(1), 37-54.
- Jazayeri, B., Zimmermann, O., Engels, G., and Kundisch, D. 2017. A Variability Model for Store-Oriented Software Ecosystems: An Enterprise Perspective. In *ICSOC 2017: Proceedings of International Conference on Service-Oriented Computing, Lecture Notes in Computer Science*, Vol. 10601, pp. 573-588. Springer, Cham.
- Jensen, M., Meckling, W., 1976. Theory of the firm: managerial behavior, agency costs, and ownership structure. *Journal of Financial Economics* 3, 305-360.
- Jergensen, C., Sarma, A., & Wagstrom, P. 2011. The onion patch: migration in open source ecosystems. In *Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering* (pp. 70-80). ACM.
- Josuttis, N. M. (2007). *SOA in practice: the art of distributed system design*. " O'Reilly Media, Inc."
- Kaler, J. 2002. Responsibility, accountability and governance. *Business Ethics: A European Review*, 11(4), 327-334.
- Kingsford, R., Dunn, L. and Cooper, J. 2003. Information Systems, IT Governance and Organisational Culture, in *14th Australasian Conference on Information Systems Perth, Western Australia*.
- Kitchenham, B e Charters, S. 2007. Guidelines for Performing Systematic Literature Review in Software Engineering. *EBSE Technical Report*, 2.3, Keele University.
- Kitchenham, B. A., and Pfleeger, S. L. (2008). 'Personal opinion surveys.' In Shull et al (Eds.), *Guide to Advanced Empirical Software Engineering*, Springer, London, pp. 63-92.
- Kittlaus, H.-B., & Clough, P. 2009. *Software Product Management and Pricing. Key Success Factors for Software Organizations*. Berlin, Heidelberg: Springer-Verlag.
- Knauss, E., Damian, D.A. Knauss, A. Borici, A. 2014. Openness and requirements: Opportunities and tradeoffs in software ecosystems. In *Proceedings of the IEEE 22nd International Requirements Engineering Conference (RE'14)*, 2014, pp.213-222.
- Koch, S., Kerschbaum, M.: 2014. Joining a smartphone ecosystem: Application developers' motivations and decision criteria. *Information and Software Technology*, Volume 56, Issue 11, pp. 1423–1435.

- Korhonen, J. 2007. On the lookout for organizational effectiveness—requisite control structure in BPM governance. In 1st International Workshop on BPM Governance—WoGo.
- Kouters, E., Vasilescu, B., Serebrenik, A., & van den Brand, M. G. 2012. Who's who in Gnome: Using LSA to merge software repository identities. In Software Maintenance (ICSM), 2012 28th IEEE International Conference on (pp. 592-595). IEEE.
- Kuechler, B., e Vaishnavi, V. 2008. On theory development in design science research: anatomy of a research project. *European Journal of Information Systems*, 17(5), 489-504.
- La Porta, R., Lopez-de-Silanes, F., Shleifer, A., & Vishny, R. 2000. Investor protection and corporate governance. *Journal of financial economics*, 58(1), 3-27.
- Lakatos, E. M. e MARCONI, M. A. 2010. *Fundamentos de Metodologia Científica*. São Paulo: 7ª Edição. Atlas.
- Lantz, M., & Weijden, O. 2013. Software Ecosystem Governance and Participation-a Case Study at Axis Communications AB. 255/2013.
- Linåker, J., Munir, H., Wnuk, K., and Mols, C. E. 2018. Motivating the contributions: An Open Innovation perspective on what to share as Open Source Software. *Journal of Systems and Software*, Vol 135, pp. 17-36.
- Manikas K., e Hansen, K.M. 2013. Reviewing the health of Software Ecosystem – A conceptual framework proposal. In proceedings of International Workshop on Software Ecosystems (IWecosistema de software'13), pp. 33-44.
- Manikas, K e Hansen, K.M. 2013. Software Ecosystems – A Systematic Literature Review. *Journal of Systems and Software*, Volume 86, Issue 5, pp. 1294-1306.
- Manikas, K. 2016. Revisiting software ecosystems research: a longitudinal literature study. *Journal of Systems and Software*, 117, 84-103.
- Manikas, K., Wnuk, K., & Shollo, A. 2015. Defining decision making strategies in software ecosystem governance. Department of Computer Science, University of Copenhagen.
- March, S. T., & Smith, G. F. 1995. Design and natural science research on information technology. *Decision support systems*, 15(4), 251-266.
- McGregor, D. J.: A method for analyzing software product line ecosystems. In Proceedings of the Fourth European Conference on Software Architecture: Companion Volume (ECSA'10), Carlos E. Cuesta (Ed.). ACM, New York, NY, USA, pp. 73–80. 2010.
- Mead, L. A., Mayer, A. R., Bobholz, J. A., Woodley, S. J., Cunningham, J. M., Hammeke, T. A., & Rao, S. M. 2002. Neural basis of the Stroop interference task: response competition or selective attention?. *Journal of the International Neuropsychological Society*, 8(6), 735-742.
- Mens, T. M. Claes, M, P. Grosjean. 2014. ECOS: Ecological studies of open source software ecosystems," In Proceedings of IEEE Conference on Software Maintenance, Reengineering and Reverse Engineering (CSMR-WCRE'14). 403-406.
- Mens, T., Claes, M., Grosjean, P., & Serebrenik, A. 2014. Studying evolving software ecosystems based on ecological models. In *Evolving Software Systems* (pp. 297-326). Springer, Berlin, Heidelberg.
- Merriam, S. B. 2002. Introduction to qualitative research. *Qualitative research in practice: Examples for discussion and analysis*, 1, 1-17.

- Messerschmitt, D. G., e Szyperski, C. 2005. Software ecosystem: understanding an indispensable technology and industry. MIT Press Books, 1.
- Mhamdia, A., B., H.:2013 Performance measurement practices in software ecosystem. *International Journal of Productivity and Performance Management*, Volume 62, Issue 5, pp. 514–533. 2013.
- Mistrik, I., Grundy, J., van der Hoek, A., Whitehead, J. 2010. Collaborative software engineering: challenges and prospects. In *Collaborative Software Engineering*. pp.389-403. Springer Berlin Heidelberg.
- Mizushima, K., and Ikawa, Y. 2011. A structure of co-creation in an open source software ecosystem: A case study of the eclipse community. In *Proceedings of Technology Management in the Energy Smart World (PICMET'11)*. 1-8.
- Monteith, J.Y., McGregor, J.D., Ingram, J.E. 2014. Proposed metrics on ecosystem health. In *Proceedings of the 2014 ACM international workshop on Software-defined ecosystems (BigSystem'14)*, ACM, pp.33-36.
- Moore, C. W. 2014. *The mediation process: Practical strategies for resolving conflict*. John Wiley & Sons.
- Musil, J., Musil, A., & Biffl, S. 2013. Elements of software ecosystem early-stage design for collective intelligence systems. In *Proceedings of the 2013 International Workshop on Ecosystem Architectures* (pp. 21-25). ACM.
- Msiska, B., & Nielsen, P. 2018. Innovation in the fringes of software ecosystems: the role of socio-technical generativity. *Information Technology for Development*, Vol. 24, No 2, pp. 398-421.
- Nabiollahi, A., e bin Sahibuddin, S. 2008. Considering service strategy in ITIL V3 as a framework for IT Governance. In *International Symposium on Information Technology* (Vol. 1, 1-6). IEEE.
- Niemann, M; Eckert, J; Repp, N Steinmetz, R. 2008. Towards a generic governance model for service-oriented architectures. In: *Proceedings of the Fourteenth Americas Conference on Information Systems*.
- Novak, J.D. and Cañas, A.J. 2008. *The Theory Underlying Concept Maps and How to Construct and Use Them*, Florida Institute for Human and Machine Cognition, Parana, Brazil, Vol. 5, No. 1, pp.9–29.
- Noy, N. F.; McGuinness, D. L. 2001. 'Ontology Development 101: A Guide to Creating Your First Ontology.' Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880. https://protege.stanford.edu/publications/ontology_development/ontology101.pdf. (Accessed 01 December 2017).
- OECD. 2004. *OECD Principles of Corporate Governance*. OECD - Organisation for Economic Co-operation and Development. <http://www.oecd.org/daf/ca/oecdprinciplesofcorporategovernance.htm>. on *Academic Software Development Tools and Techniques (WASDeTT)*. Acesso em 01 de fevereiro de 2017.
- Oruganti, V., and Mangematin, V. 2018. Firm Participation in a Platform Ecosystem–Value from Data and Competition. In *DRUID: Proceedings of Druid Academy Conference*, University of Southern Denmark, Odense, Denmark, pp. 17-19.

- Ostrowski, L., Helfert, M., & Xie, S. 2012. A conceptual framework to construct an artefact for meta-abstract design knowledge in design science research. In 45th Hawaii International Conference on System Science .4074-4081. IEEE.
- Papatheocharous, E., Andersson, J., & Axelsson, J. 2015. Ecosystems and Open Innovation for Embedded Systems: A Systematic Mapping Study. In International Conference of Software Business. 81-95. Springer International Publishing.
- Parker, G., Van Alstyne, M., and Jiang, X. 2017. Platform Ecosystems: How Developers Invert the Firm. *MIS Quarterly*, Vol. 41, No 1, pp. 255-266.
- Petersen, K., Vakkalanka, S., Kuzniarz, L., 2015. Guidelines for Conduction Systematic Mapping Studies in Software Engineering: An Update. *Information and Software Technology*. Vol 56/11. 2015.
- Popp, K.M. 2011. Hybrid revenue models of software companies and their relationship to hybrid business models. In proceedings of Third International Workshop on Software Ecosystems (IWecosistema de software'11). 77-88.
- Pries-Heje, J., Baskerville, R., e Venable, J. 2008. Strategies for design science research evaluation. *ECIS 2008 proceedings*, 1-12.
- Santos, R., Valença, G., Viana, D., Estácio, B., Fontão, A., Marczak, S., Werner, C., Alves, C., Conte, T., e Prikładnicki, R. 2014. Qualidade em ecossistemas de software: Desafios e oportunidades de pesquisa. In Proceedings of VIII Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems, pages41–44.
- Rahim, M. A. 2002. Toward a theory of managing organizational conflict. *The International Journal of Conflict Management*, 13, 206-235.
- Santana, A. F. 2015. BPMG – Um Modelo Conceitual para Governança em BPM. Tese de doutorado em Ciência da Computação. Universidade Federal de Pernambuco, Recife.
- SAP. Agreement of involvement of SAP SE. 2014. <https://www.sap.com/corporate/en/investors/governance.html#pdf-asset=6a4198f5-537c-0010-82c7-eda71af511fa&page=1>.
- SAP. Articles of Incorporation of SAP SE. 2018a. <https://www.sap.com/corporate/en/investors/governance.html#pdf-asset=98a5b998-0b7d-0010-87a3-c30de2ffd8ff&page=2>.
- SAP. Code of Business Conduct. 2018 c. <https://www.sap.com/corporate/en/investors/governance.html#pdf-asset=e090dc55-997c-0010-82c7-eda71af511fa&page=1>.
- SAP. Governance Statement. 2018b. <https://www.sap.com/docs/download/investors/2018/sap-2018-cgs.pdf>
- Schepers, T. G., Iacob, M. E., e Van Eck, P. A. A lifecycle approach to SOA governance. In Proceedings of the 2008 ACM symposium on Applied computing. 1055-1061. ACM.2008.
- Schreieck, M., Wiesche, M., e Krcmar, H. 2016. Design and Governance of Platform Ecosystems–Key Concepts and Issues for Future Research. In Twenty-Fourth European Conference on Information Systems (ECIS).
- Shin, M., Holden, T., & Schmidt, R. A. 2001. From knowledge theory to management practice: towards an integrated approach. *Information processing & management*, 37(2), 335-355.

- Silva A., S., de Almeida, E. S., McGregor, J. D., e von Flach G Chavez, C. 2014. When ecosystems collide: making systems of systems work. In Proceedings of the 2014 European Conference on Software Architecture Workshops (p. 29). ACM.
- Sjøberg, D.I.K., Dybå, T., Anda, B. C., and Hannay, J. E. 2008. Building theories in software engineering. In Shull et al (Eds.): Guide to Advanced Empirical Software Engineering. Springer, London, pp. 312–336.
- Slinger Jansen and Michael Cusumano. 2013. Defining software ecosystems: a survey of software platforms and business network governance. In Slinger Jansen, Michael Cusumano, & Sjaak Brinkkemper, eds. Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry. Edward Elgar Publishers.
- Sortica, E. A., Clementi, S., e Carvalho, T. 2004. Governança de TI: Comparativo entre COBIT e ITIL. In Anais do Congresso Anual de Tecnologia da Informação-CATI.
- Spauwen, R., and Jansen, S.: 2013. Towards the roles and motives of open source software developers. In Proceedings of the 5th International Workshop on Software Ecosystems (IWSECO'13), p. 62.
- Sutherland, S. and Katz, S. 2005. Concept mapping methodology: a catalyst for organizational learnings, Evaluation and Program Planning, Vol. 28, No. 3, pp.257–269.
- Tilakaratna, P., and Rajapakse, J. 2017. Evaluation of the Ontological Completeness and Clarity of Object-Oriented Conceptual Modelling Grammars. Journal of Database Management, Vol. 28 No 2, pp. 1-26.
- Tilson, D., Kalle Lyytinen, C. Sørensen. 2010. Digital infrastructures: The Missing 15 Research Agenda. Information Systems Research 21(4) p.748–759.
- Tiwana, B., Konsynski, B. and A. Bush, A. A. 2010. Platform evolution: Coevolution of platform architecture, governance, and environmental dynamics. Information Systems Research, Volume 21, Issue 4, 675–687.
- Valenca, G., Alves, C., Heimann, V., Jansen, S., Brinkkemper, S. 2014. Competition and collaboration in requirements engineering: A case study of an emerging software ecosystem. In Proceedings of the IEEE 22nd International Requirements Engineering Conference (RE'14), pp.384-393.
- van Angeren, J., Jansen, S., e Alves C. 2016. Can We Ask You To Collaborate? Analyzing App Developer Relationships in Commercial Platform Ecosystems. Journal of Systems and Software, Volume 113, pp. 430-445. 2016.
- van Angeren, J., Jansen, S., Brinkkemper, S. 2014. Exploring the relationship between partnership model participation and interfirm network structure: An analysis of the office365 ecosystem, In Proceedings of Software Business (ICSOB'14). Towards Continuous Value Delivery, Springer, pp. 1-15.
- van Angeren, J., Kabbedijk, J., Jansen, S., Popp, K.M. 2011. A Survey of Associate Models used within Large Software Ecosystems. In proceedings of the Third International Workshop on Software Ecosystems. 1-13.
- Van Angeren, J., Kabbedijk, J., Popp, K. M., and Jansen, S. 2013. Managing software ecosystems through partnering. in Jansen, S. et al (Eds.), Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry, Edward Elgar Pub, pp. 85-102.

- van den Berk, I., Jansen, S., e Luinenburg, L. 2010. Software ecosystems: a software ecosystem strategy assessment model. In Proceedings of the Fourth European Conference on Software Architecture: Companion Volume. 127-134). ACM.
- Van Lingen, S., Palomba, A., e Lucassen, G. 2013. On the software ecosystem health of open source content management systems. In proceedings of the 5th International Workshop on Software Ecosystems (IWecosistema de software 2013), pp. 38.
- Varadan, R., Channabasavaiah, K., Simpson, S., Holley, K., Allam, A. 2008. Increasing business flexibility and SOA adoption through effective SOA governance. *IBM Systems Journal*, 47(3), 473-488.
- Venable, J., Pries-Heje, J., & Baskerville, R. 2012. A comprehensive framework for evaluation in design science research. In *International Conference on Design Science Research in Information Systems*. 423-438. Springer Berlin Heidelberg.
- Voas, J. M., and Ghosh, A. K. 2005. Patent and Trademark Office. U.S. Patent No. 6,862,696. Washington, DC: U.S
- Walls, J. G., Widmeyer, G. R., & El Sawy, O. A. 1992. Building an information system design theory for vigilant EIS. *Information systems research*, 3(1), 36-59.
- Wareham, J., Fox, P. B., e Cano Giner, J. L. 2014. Technology ecosystem governance. *Organization Science*, 25(4), 1195-1215.
- Webb, P., Pollard, C., e Ridley, G. 2006. Attempting to define IT governance: Wisdom or folly? Proceedings of the 39th Annual Hawaii International Conference on System Sciences (Vol. 8, 194a-194a). IEEE.
- Weiblen, T., Giessmann, A., Bonakdar, A., and Eisert, U. 2012. Leveraging the software ecosystem: Towards a business model framework for marketplaces.' pp. 187-193.
- Weill, P., & Ross, J. W. 2004. *IT governance: How top performers manage IT decision rights for superior results*. Harvard Business Press.
- West, J., and Wood, D. 2011. Tradeoffs of Open Innovation Platform Leadership: The Rise and Fall of Symbian Ltd. In proceedings of Social Science and Technology Seminar Series.
- West, J., e Wood, D. 2008. *Creating and Evolving an Open Innovation Ecosystem: Lessons from Symbian Ltd*. Available at SSRN Working Paper Series.
- Wieringa, R. J. 2014. What Is Design Science? In *Design Science Methodology for Information Systems and Software Engineering*. 3-11). Springer Berlin Heidelberg.
- Wieringa, R., Maiden, N., and Mead, N., 2006. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirements Engineering Journal*, Vol 11, Issue 1.
- Wnuk, K., Manikas, K., Runeson, P., Lantz, M., Weijden, O., Munir, H.: 2014. Evaluating the governance model of hardware-dependent software ecosystems — a case study of the axis ecosystem. In Lassenius, C., Smolander, K., eds.: *Software Business. Towards Continuous Value Delivery*. Volume 182 of *Lecture Notes in Business Information Processing*. Springer International Publishing. 212–226.
- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., and Wesslén, A. 2012. *Experimentation in Software Engineering*, 1nd ed., Springer Science & Business Media.

- Yu, Chong-ho & Barbara Ohlund. 2010. Threats to validity of Research Design. Retrieved January, Vol 12.
- Yu, L., Ramaswamy, S., e Bush, J. 2007. Software evolvability: An ecosystem point of view. In Third International IEEE Workshop on Software Evolvability.75-80). IEEE.

APÊNDICE A – SITE COM INSTRUÇÕES PARA PARTICIPANTES DO SURVEY E DAS ENTREVISTAS

A - Tela inicial – Instruções sobre o survey

1. Instructions (Survey)	1. Instructions (Survey)
2. Software Ecosystem Governance Model (Before survey)	This survey aims to evaluate a conceptual model of Software Ecosystem Governance proposed as part of a PhD thesis. The survey should take around 20 minutes to complete.
3. Additional Info	A conceptual model is the representation of a set of entities and relationships between these entities, which are part of a given knowledge domain and which helps to understand the domain in question. The model was built using UML (Unified Modelling Language). In this link , you can see UML notation. The model represents a holistic view of the elements involved in the governance of a software ecosystem and the respective relationships among the elements. We present the definitions of the elements here . The model is based on the assumption that understanding governance elements and applying them in practice increases the chances of survival of a software ecosystem.
3.1. UML Notation	
3.2. Glossary	
3.3. References	A set of propositions is proposed to explain the constructs and relationships of the Software Ecosystem Governance Model. Each proposition includes evidence from studies published in the area that was synthesized from a Systematic Literature Review. These studies can be checked in the references .
4. Survey	
5. Software Ecosystem Governance Model (After survey)	The survey is divided into three sections: (1) has questions about your experience in the software ecosystem governance area. (2) has questions that aim to evaluate each proposition of model based on evidence from the literature. To answer this section, you can follow these steps:
6. Instructions (Interview)	<ol style="list-style-type: none"> 1. Read the evidence. 2. Analyze the proposition associated with the evidence. 3. Check the part of the model that represents the proposition. 4. If you have doubt about any element definition, check it at the glossary. 5. Mark your answer (I agree/ I don't agree/ I don't now). 6. Justify your opinion with comments and examples. Please, help us to improve the conceptual model adding suggestions.
Sitemap	
	Finally, in section (3) evaluates all model based on criteria proposed by Sjöberg (2008): testability, empirical support, explanatory power, parsimony, generality, and utility.
	Your contribution is extremely important for our research. We are very grateful for your kind collaboration!

B- Tela com modelo conceitual antes do survey

1. Instructions (Survey)	2. Software Ecosystem Governance Model (Before survey)
2. Software Ecosystem Governance Model (Before survey)	We present the Software Ecosystem Governance Model. Please analyze the complete model to answer the third section of the survey. You can download the image here .
3. Additional info	<ul style="list-style-type: none"> • Check the glossary to read the definition of each element. • Check the evidence presented in the survey to see how the elements are related. • If you are not familiar with UML, check the notation.
3.1. UML Notation	Click on the image to see it full.
3.2. Glossary	
3.3. References	
4. Survey	
5. Software Ecosystem Governance Model (After survey)	
6. Instructions (Interview)	
Sitemap	

C- Tela com informações adicionais – Notação UML

1. Instructions (Survey)

2. Software Ecosystem Governance Model (Before survey)

3. Additional info

3.1. UML Notation

3.2. Glossary

3.3. References

4. Survey

5. Software Ecosystem Governance Model (After survey)

6. Instructions (Interview)

Sitemap

3. Additional info >

3.1. UML Notation

The conceptual model uses symbols of UML notation. More advanced concepts such as aggregation, inheritance, cardinality may be used in the future. Currently, we only adopt the concept of association.

Notational Symbols	Description
	A is an element of the conceptual model.
	The line indicates the relationship between elements. R is the name of the relationship. In the conceptual model R is always represented by a verb in the third person singular in the present tense.
	The element A has relation R with the element B.
	D and E are subtypes of C.
	The arrow represents an inheritance.
P	P is a proposition based on evidence from the literature that explains the relationships among elements.

D -Tela de informações adicionais – Glossário de elementos

1. Instructions (Survey)

2. Software Ecosystem Governance Model (Before survey)

3. Additional info

3.1. UML Notation

3.2. Glossary

3.3. References

4. Survey

5. Software Ecosystem Governance Model (After survey)

6. Instructions (Interview)

Sitemap

3. Additional info >

3.2. Glossary

- **Actor:** Organization or individual that participates in the software ecosystem assuming one or more roles (Manikas, 2014).
- **Agreement:** Set of rules that define the process, strategies, and limitations of conflict resolution in order to attend all parties involved (Moore, 2014).
- **Associate model:** A set of strategies to maintain, manage, group and expand the partners of a software ecosystem (van Ageren et al, 2011).
- **Certification:** Certification is a process that validates conformance to an expectation or standard, demonstrating that software was produced in a particular manner (Voas, 2002).
- **Coopetition:** collaboration between business competitors, in the hope of mutually beneficial results (Valença et al, 2014).
- **Collaboration:** the recursive process in which two or more people or organizations work together towards an intersection of common goals (Mistrik et al, 2010).
- **Communication channel:** Describe how a company communicates with and reaches its customer segments to deliver a value proposition (Axelson et al, 2014). It consists of means for sending and receiving messages and information dissemination that allow the participation of all interested parties in discussions of work in progress, regardless of their location (Knauss, 2014) (Rong et al, 2015).
- **Competition:** The act of seeking or endeavoring to gain what another is endeavoring to gain at the same (Mead, 2002).
- **Conflict:** Interactive process manifested in incompatibility, disagreement or discordance within or between entities (Rahim, 2002).
- **Distribution strategy:** Ways of allocating resources by analyzing their availability and estimating the number of consumers (Arrow, 1962).
- **Entry barrier:** It consists of a set of requirements under which a stakeholder is assessed before formally joining a group (Gawer and Henderson, 2007).
- **Health:** The ability of an organization to remain sustainable and productive over time (Manikas and Hansen, 2013).
- **Knowledge:** Accumulation of information organized from data that represent a domain and product of the interaction between individual cognition and reality (Krogh, 1998), (Shin et al, 2001).

E -Tela com informações adicionais – Referências dos estudos incluídos na revisão sistemática da literatura

<ul style="list-style-type: none"> 1. Instructions (Survey) 2. Software Ecosystem Governance Model (Before survey) 3. Additional info 3.1. UML Notation 3.2. Glossary 3.3. References 4. Survey 5. Software Ecosystem Governance Model (After survey) 6. Instructions (Interview) Sitemap 	<p>3. Additional info ></p> <h3>3.3. References</h3> <p>S1. Albert, B.E., Santos, R.P., Werner, C.: Software ecosystems governance to enable IT architecture based on software asset management. In Proceedings of the 7th IEEE International Conference on Digital Ecosystems and Technologies (2013).</p> <p>S2. Alspaugh, T.A., Asuncion, H. U., Scacchi, W.: The Role of Software Licenses in Open Architecture Ecosystems. In proceedings of the 1st International Workshop on Software Ecosystems (IWSECO'09), pp. 4–18 (2010).</p> <p>S3. Axelsson, J., Papatheocharous, E., Andersson, J.: Characteristics of software ecosystems for Federated Embedded Systems: A case study. Information and Software Technology, Volume 56, Issue 11, pp.1457–1475 (2014).</p> <p>S4. Baars, A., Jansen, S.: A framework for software ecosystem governance. In Proceedings of the Third International Conference on Software Business (ICSOB'12), pp.168–180 (2012).</p> <p>S5. Bosch, J., Bosch-Sijtsema, M.P., Esao: A Holistic Ecosystem-Driven Analysis Model. In Proceedings of the 5 th International Conference on Software Business (ICSOB 2014), Springer Verlag, pp. 179–193 (2014).</p> <p>S6. Burkard, C., Widjaja, T., Buxmann, P., Software ecosystems. Business & Information Systems Engineering, volume 4, Issue. 1, pp. 41–44 (2012).</p> <p>S7. Campbell, P., Ahmed, F.: A three-dimensional view of software ecosystems. In Proceedings of the Fourth European Conference on Software Architecture: Companion Volume (ECSA '10), CarlosE. Cuesta (Ed.). ACM, New York, NY, USA, pp.81–84 (2010).</p> <p>S8. Ceccagnoli, M., Forman, C., Huang, P., Wu, D.J.: Co-creation of value in a platform ecosystem: The case of enterprise software. MIS Quarterly, Forthcoming (2011).</p> <p>S9. Che, M., Perry, D.E.: Architectural Design Decisions in Open Software Development: A Transition to Soft/ware Ecosystems. In Proceedings of the 23rd Australian Software Engineering Conference, pp.58–61 (2014).</p> <p>S10. Den Hartigh, E., Tol, M., Visscher, W.: The health measurement of a business ecosystem. In proceedings of the European Network on Chaos and Complexity Research and Management Practice Meeting, pp. 1–39 (2006).</p> <p>S11. Dittrich, Y.: Software engineering beyond the project - Sustaining software ecosystems. Information and Software Technology, Volume 56, Issue 11, pp. 1436–1456 (2014).</p> <p>S12. Dubinsky, Y., Kruchten, P.: Software development governance (sdg): report on 2nd workshop. ACM SIGSOFT Software Engineering Notes, Volume 34, pp. 46–47 (2009).</p> <p>S13. Eisenmann, T.R., Parker, G., M.W.: Van Alstyne, M.W.: Opening platforms: how, when and why? In Platforms, Markets and Innovation. Gawer A (ed). Edward Elgar: Cheltenham, UK, pp.131–162 (2008).</p> <p>S14. Eklund, U., Bosch, J.: Introducing software ecosystems for mass-produced embedded systems. In Software Business. Springer Berlin Heidelberg, pp. 248–254 (2012).</p> <p>S15. Fricker, S.: Requirements Value Chains: Stakeholder Management and Requirements Engineering in Software Ecosystems. In proceedings of Requirements Engineering: Foundation for Software</p>
---	---

F - Tela de acesso ao survey

<ul style="list-style-type: none"> 1. Instructions (Survey) 2. Software Ecosystem Governance Model (Before survey) 3. Additional info 3.1. UML Notation 3.2. Glossary 3.3. References 4. Survey 5. Software Ecosystem Governance Model (After survey) 6. Instructions (Interview) Sitemap 	<h3>4. Survey</h3> <p>Click here to open the survey on another page.</p> <div style="border: 1px solid #ccc; padding: 10px;"> <h2 style="text-align: center;">Software Ecosystem Governance Model</h2> <p style="color: red; text-align: center;">* Required</p> <p>1. Name (Your name will not be revealed).</p> <p>Your answer <input type="text"/></p> <p>2. Email address</p> <p>Your answer <input type="text"/></p> <p>3. What is your highest academic degree? *</p> <p><input type="radio"/> Postdoctoral</p> <p><input type="radio"/> Ph.D. Degree</p> <p><input type="radio"/> Ph.D. Student</p> </div>
---	---

G - Tela de acesso ao modelo conceitual após a realização do survey

1. Instructions (Survey)	<h3>5. Software Ecosystem Governance Model (After survey)</h3> <p>We present the Software Ecosystem Governance Model. Please analyze the complete model to answer the third section of the survey. You can download the image here.</p> <ul style="list-style-type: none"> Check the propositions below. Check the conceptual model below. Check the glossary to read the definition of each element. If you are not familiar with UML, check the notation. <h4>Propositions</h4> <p>P1: <i>Coopetition</i> may generate <i>innovation</i>.</p> <p>P2: <i>Innovation</i> may promote <i>value creation</i>.</p> <p>P3: An <i>associate model</i> guides <i>value creation</i>.</p> <p>P4: An <i>associate model</i> defines a set of <i>roles</i>, defines <i>entry barriers</i>, and governs <i>relationships</i>, influencing the <i>health</i> of the ecosystem.</p> <p>P5: Software ecosystems health can be measured by <i>productivity</i>, <i>robustness</i>, and <i>niche creation</i>.</p> <p>P6: <i>Agreements</i> can guide the resolution of <i>conflicts</i> in <i>relationships</i> in a software ecosystem.</p> <p>P7: <i>Actors</i> participate in a <i>relationship</i>.</p> <p>P8: <i>Revenue model</i> with <i>monetary</i> and <i>non-monetary</i> compensation streams attract <i>actors</i>.</p> <p>P9: <i>Communication channels</i> bring actors into alignment and may foster <i>transparency</i> to build <i>trust</i>.</p> <p>P10: <i>Communication channels</i> enable <i>information</i> propagation.</p> <p>P11: <i>Certifications</i> and <i>quality standards</i> act as an <i>entry barrier</i> to the software ecosystem.</p> <p>P12: The exercise of a <i>role</i> by an <i>actor</i> carries <i>responsibility</i>.</p> <p>P13: <i>Responsibility</i> promotes <i>trust</i> in <i>relationships</i>.</p>
2. Software Ecosystem Governance Model (Before survey)	
3. Additional info	
3.1. UML Notation	
3.2. Glossary	
3.3. References	
4. Survey	
5. Software Ecosystem Governance Model (After survey)	
6. Instructions (Interview)	
Sitemap	

H -Tela de acesso à instruções para os participantes das entrevistas

1. Instructions (Survey)	<h3>6. Instructions (Interview)</h3> <p>The interviews aim to evaluate the conceptual model of Software Ecosystem Governance validated by experts. The interview should take around 30 minutes to response and we will perform it through Skype, Hangouts or WhatsApp. To facilitate the collection of data we will record the interviews if you give us permission. The answers you provide are never right or wrong. We are looking for an answer to your experience.</p> <p>A conceptual model is the representation of a set of entities and relationships between these entities, which are part of a given knowledge domain and which helps to understand the domain in question. The model was built using UML (Unified Modelling Language). In this link, you can see UML notation. The model represents a holistic view of the elements involved in the governance of a software ecosystem and the respective relationships among the elements. We present the definitions of the elements here. The model is based on the assumption that understanding governance elements and applying them in practice increases the chances of survival of a software ecosystem.</p> <p>A set of propositions is proposed to explain the constructs and relationships of the Software Ecosystem Governance Model. Each proposition includes evidence from studies published in the area that was synthesized from a Systematic Literature Review (SLR) and validated by experts in the software ecosystem area. The studies of the SLR can be checked in the references.</p> <p>The interview is divided into two sections: (1) has questions about your experience in the software ecosystem area. (2) has questions that aim evaluate each proposition of the conceptual model. To answer the interview, you can follow these steps:</p> <ol style="list-style-type: none"> Read each proposition of the conceptual model. Check the part of the model that represents the proposition. If you have doubt about any element definition, check it at the glossary. If you have doubt about the UML notation, check the here. Answer the interview justifying your opinion with comments and examples. Please, help us to improve the conceptual model adding suggestions. <p>Your contribution is extremely important for our research. We are very grateful for your kind collaboration!</p>
2. Software Ecosystem Governance Model (Before survey)	
3. Additional info	
3.1. UML Notation	
3.2. Glossary	
3.3. References	
4. Survey	
5. Software Ecosystem Governance Model (After survey)	
6. Instructions (Interview)	
Sitemap	

APÊNDICE B – QUESTIONÁRIO COM ESPECIALISTAS

26/01/2019

Software Ecosystem Governance Model

Software Ecosystem Governance Model

* Required

1. Name (Your name will not be revealed).

2. Email address

3. What is your highest academic degree? *

Mark only one oval.

- Postdoctoral
- Ph.D. Degree
- Ph.D. Student
- Master Degree
- Master Student
- Bachelor Degree

4. Where do you work? *

Mark only one oval.

- Industry
- Academy
- Both

5. How many years do you conduct research in software ecosystems? *

Mark only one oval.

- 0-2 years
- 2-4 years
- 4-6 years
- 6 and more years

6. How many papers did you publish in software ecosystem area? *

Mark only one oval.

- 1-2 papers
- 2-4 papers
- 4-6 papers
- 6 and more papers

26/01/2019

Software Ecosystem Governance Model

7. Do you have practical experience in software ecosystem area? (If yes, tell us about your experience.) *

Mark only one oval.

No

Yes

8. In your opinion, what is software ecosystem governance?

9. What is the importance of governance of software ecosystems?

Evaluation of Software Ecosystem Governance Model propositions

To answer the following questions, you can follow these steps:

1. Read the evidence. The evidence was obtained from primary studies [S1..S63] of a systematic literature review.
2. Analyze the proposition associated with the evidence.
3. Check the part of the model that represents the proposition.
4. If you have doubt about any element definition, check it at the glossary.
5. **Mark your answer (I agree/ I don't agree/ I don't now).**
6. Justify your opinion with comments and examples. Please, help us to improve the conceptual model by adding suggestions.

Propositions

Evidence 1

In a software ecosystem, participants simultaneously compete and cooperate with each other [S40]. The complex interplay of competition and collaboration strategies continues during the lifecycle of an ecosystem. Coopetition (the collaboration between business competitors, in the hope of mutually beneficial results) is the main driver of innovation and performance [S52]. Coopetition takes the

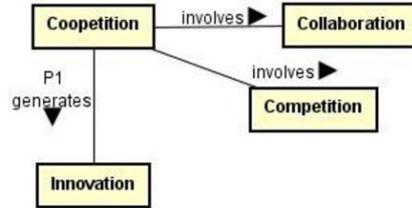
26/01/2019

Software Ecosystem Governance Model

relationship between the companies to a new level, where actors work together to identify innovative requirements and deliver new solutions that address market needs[S52].

Proposition 1

Coopetition generates innovation.



Mark only one oval.

- I agree
- I do not agree
- I do not know

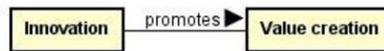
Comments/ Suggestions/ Examples

Evidence 2

Innovation can produce win-win situations for the platform leader, product manufacturers and customers [S19][S60]. The more innovation there is the more value is created for the platform [S17].

Proposition 2

Innovation promotes value creation.



Mark only one oval.

- I agree
- I do not agree
- I do not know

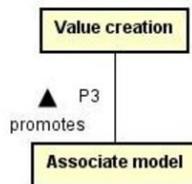
Comments/ Suggestions/ Examples

Evidence 3

An Associate model enables actors to join in and creates value for the ecosystem according to specific roles and positions in the ecosystem[S28][S53].

Proposition 3

An Associate Model promotes value creation.



Mark only one oval.

- I agree
- I do not agree
- I do not know

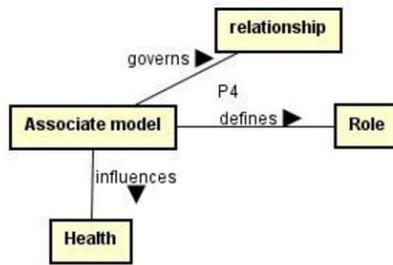
Comments/ Suggestions/ Examples

Evidence 4

Complex relationships are constructed among actors inside and outside of the ecosystem [S40]. Relationships with weak actors can play a strong role in destabilizing the ecosystem [S28]. To manage relationships among actors is a key success factor for the healthy evolution of a software ecosystem [S10][S25][S54]. The introduction of a associate model can strengthen weak actors and bring high potential partners closer to the ecosystem [S28].

Proposition 4

An Associate Model defines a set of roles and governs partnerships, it directly influences the health of the ecosystem.



Mark only one oval.

- I agree
- I do not agree
- I do not know

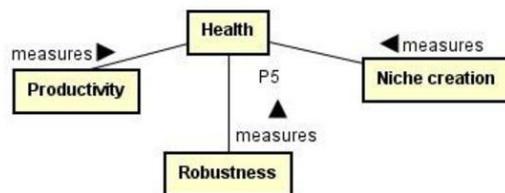
Comments/ Suggestions/ Examples

Evidence 5

The health of an ecosystem is determined by the capability of an ecosystem to persistently produce meaningful outputs (productivity), survive market disruptions (robustness) and create niches in the ecosystem (niche creation) [S10][S25].

Proposition 5

Productivity, robustness and niche creation measures health.



Mark only one oval.

- I agree
- I do not agree
- I do not know

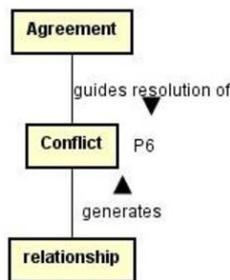
Comments/ Suggestions/ Examples

Evidence 6

Conflicts are very common in software ecosystems when relationships have not clearly established power and coordination mechanisms are incipient or even inexistent [S52]. According to Fricker [S15], agreements, that are results of successful negotiation among interested parties, are a way to solve conflicts.

Proposition 6

Agreements can guide the resolution of conflicts in relationships in a software ecosystem.



Mark only one oval.

- I agree
- I do not agree
- I do not know

Comments/ Suggestions/ Examples

Evidence 7

A software ecosystem is a set of actors functioning as a unit and interacting with a shared market for software and services, together with the relationships among them [S28][S29][S54][S23]. These relationships are frequently underpinned by a common technological platform or market and operate through the exchange of information, resources and artifacts[S28][S29].

26/01/2019

Software Ecosystem Governance Model

Proposition 7

Actor has relationship.

*Mark only one oval.*

- I agree
 I do not agree
 I do not know

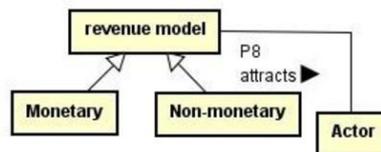
Comments/ Suggestions/ Examples

Evidence 8

According to Hyrynsalmi et al [S23], a hybrid revenue model with monetary and non-monetary revenue streams represents a competitive advantage for software ecosystem because it facilitates the adhesion of different actors in relationships, allowing the ecosystem expansion in many directions.

Proposition 8

Hybrid revenue models with monetary and non-monetary streams attract actor.

*Mark only one oval.*

- I agree
 I do not agree
 I do not know

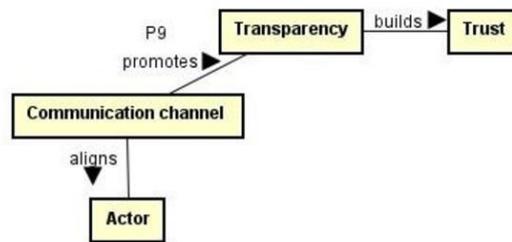
Comments/ Suggestions/ Examples

Evidence 9

According to Knauss [S37], communication channels allow the alignment of several actors and shape the attractiveness of the ecosystem to new actors. Effective communication channels require an elevated level of transparency [S37]. If a valuable information is removed from the communication channel, the transparency will be threatened and trust can be broken [S37], [S16].

Proposition 9

Communication channels align actors, and promote transparency that builds trust.



Mark only one oval.

- I agree
- I do not agree
- I do not know

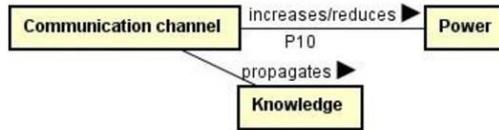
Comments/ Suggestions/ Examples

Evidence 10

The interaction among partners raises the need for knowledge management to support communication and efficient propagation of information [S52]. Communication channels are designed to enable knowledge sharing among actors [S29]. According to Fricker [S16], actions to increase power include better integration of an actor into the communication channel. The power can be indirectly reduced by isolating the actor of the communication channels [S16].

Proposition 10

Communication channels enable knowledge sharing and can be used to increase or reduce the power of an actor.



Mark only one oval.

- I agree
- I do not agree
- I do not know

Comments/ Suggestions/ Examples

Evidence 11

According to Axelsson et al [S3], the main resources used by the actors of the ecosystem include products, services, key people and platforms. Actors are selected and admitted to the relationships because they can provide desired resources [S42]. In a software ecosystem, actors share resources that include not only technical resources but also tacit knowledge of all parties [S9]. Thus, management is required to deal with the diverse resources distributed in multiple actors [S9] [S15].

Proposition 11

Resources are efficiently shared among actors with strategies of resource distribution.



Mark only one oval.

- I agree
- I do not agree
- I do not know

Comments/ Suggestions/ Examples

Evidence 12

According to Axelsson et al. [S3], resources is regard as a main entry barrier to SECO. For example, an actor may have to contribute a number of full-time developers (human resource) to platform development. Another actor may have to pay regular fees (financial resource) to join to software ecosystem [S3].

Proposition 12

Resources act as entry barrier to software ecosystem.



Mark only one oval.

- I agree
- I do not agree
- I do not know

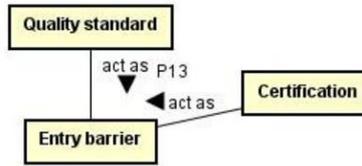
Comments/ Suggestions/ Examples

Evidence 13

According to [S32] and [S53], to become a partner, an organization may have to certify its product or service so that they comply with ecosystem guidelines [S53]. A software ecosystem can encourage the formation of relationships by handling out certifications based on quality criteria [S32]. According to [S32], [S50] and [S62], certifications and quality standards will inform the quality level expected from services and products of partners, acting as entry barriers into the ecosystem software.

Proposition 13

Certifications and quality standards act as entry barrier into the software ecosystem.



Mark only one oval.

- I agree
- I do not agree
- I do not know

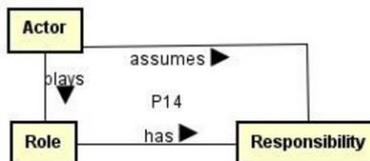
Comments/ Suggestions/ Examples

Evidence 14

Responsibility is an important factor in ecosystem governance [S4]. Without appointed members of the organization being responsible for the ecosystem, the correct execution of the governance strategy cannot be guaranteed [S4].

Proposition 14

The adoption of a role by an actor brings responsibility that must be formally assumed to ensure the correct execution of the software ecosystem governance.



Mark only one oval.

- I agree
- I do not agree
- I do not know

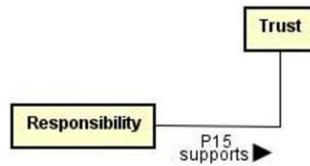
Comments/ Suggestions/ Examples

Evidence 15

If trust is broken in a relationship, it can be repaired by taking personal responsibility for the breach [S16]. Trust, however, can only be repaired if the breach is an isolated event and if the risk of deception is effectively mitigated [S16].

Proposition 15

Responsibility supports trust in a relationship among actors in a software ecosystem.



Mark only one oval.

- I agree
- I do not agree
- I do not know

Comments/ Suggestions/ Examples

Evaluation of all elements and relations of the Software Ecosystem Governance Model

In the next questions, you can to evaluate the complete software ecosystem governance model according to following criteria (Sjorberg, 2008): testability, empirical support, explanatory power, parsimony, generality, and utility.

Testability

It is the degree to which a theory is constructed such that empirical refutation is possible (Sjorberg, 2008).

26/01/2019

Software Ecosystem Governance Model

1. Are the elements and relations of the model understandable?

Mark only one oval.

- Yes
- No
- I do not know

Comments/Suggestions/Examples:

2. Are the elements and relations of the model internally consistent (without contradictory relations) and free from ambiguities?

Mark only one oval.

- Yes
- No
- I do not know

Comments/ Suggestions/ Examples

3. Are there elements and relations empirically refutable in the model?

Mark only one oval.

- Yes
- No
- I do not know

Comments/Suggestions/Examples:

Empirical support

It is the degree to which a theory is supported by empirical studies that confirm its validity;

26/01/2019

Software Ecosystem Governance Model

1. Are there empirical studies that confirm the validity of elements and relations of the model?

Mark only one oval.

- Yes
 No
 I do not know

Comments/ Suggestions/ Examples

Explanatory power

The degree to which a theory accounts for and predicts all known observations within its scope, is simple in that it has few ad hoc assumption, and relates to that which is already well understood.

1. Are the elements and relationships of the model already well understood by the community of software ecosystems?

Mark only one oval.

- Yes
 No
 I do not know

Comments/ Suggestions/ Examples

2. Are there ad hoc assumptions related to the elements and relationships?

Mark only one oval.

- Yes
 No
 I do not know

26/01/2019

Software Ecosystem Governance Model

Comments/ Suggestions/ Examples

3. Is the software ecosystem governance model simple to understand?

Mark only one oval.

- Yes
- No
- I do not know

Comments/ Suggestions/ Examples

Parsimony

It is the degree to which a theory is economically constructed with a minimum of concepts and propositions.

1. Were used the minimum of elements and relationships to build the model?

Mark only one oval.

- Yes
- No
- I do not know

Comments/ Suggestions/ Examples

Generality

The breadth of the scope of a theory and the degree to which the theory is independent of specific settings.

26/01/2019

Software Ecosystem Governance Model

1. Is the software ecosystem governance model generalizable to different kinds of software ecosystems, such as open source and proprietary?

Mark only one oval.

- Yes
- No
- I do not know

Comments/ Suggestions/ Examples

Utility

It is degree to which a theory supports the relevant areas of the software industry.

1. Has the model practical and/or theoretical utility?

Mark only one oval.

- Yes
- No
- I do not know

Comments/ Suggestions/ Examples

Powered by
 Google Forms

APÊNDICE C – ROTEIRO DE ENTREVISTAS COM PRATICANTES

A - Questões demográficas

1. Qual o seu nome?
2. Qual o seu email?
3. Em qual empresa você trabalha?
4. Em qual função você atua?
5. Você desenvolve para qual (quais) ecossistema (s) de software (SAP, Eclipse, IOS, Android, Linux, Ruby, AppStore, GNOME...)
6. Há quanto tempo você atua neste ecossistema de software?

B - Questões relacionadas ao modelo conceitual para governança de ecossistemas de software

1. Como é o ambiente de competição no ecossistema no qual você atua?
2. Quais atividades você realiza no ecossistema no qual você atua?
3. Como o ecossistema no qual você atua estimula a inovação?
4. Como ações de inovação e de geração de valor são percebidas no ecossistema no qual você atua?
5. Quais são os requisitos necessários para que um desenvolvedor atue neste ecossistema?
6. Existem classes diferentes de desenvolvedores com responsabilidades distintas no ecossistema no qual você atua? Caso positivo, quais são estas classes?
7. Existem formas de mensuração de produtividade no ecossistema no qual você atua?

8. Como os conflitos são gerenciados no ecossistema no qual você atua?
9. Como os desenvolvedores são recompensados por suas atividades no ecossistema no qual você atua?
10. Como a comunicação é gerenciada no ecossistema no qual você atua?
11. Quais certificações e padrões de qualidade são necessários para ingressar no ecossistema no qual você atua?
12. Houve algum momento em que você se sentiu afetado por decisões da alta gestão do ecossistema no qual você atua?

ANEXO A – ESTUDO INCLUÍDOS NA REVISÃO SISTEMÁTICA DA LITERATURA

- S1. Albert, B.E., Santos, R.P., Werner, C.: Software ecosystems governance to enable IT architecture based on software asset management. In Proceedings of the 7th IEEE International Conference on Digital Ecosystems and Technologies (2013).
- S2. Alsbaugh, T.A., Asuncion, H. U., Scacchi, W.: The Role of Software Licenses in Open Architecture Ecosystems. In proceedings of the 1st International Workshop on Software Ecosystems (IWSECO'09), pp. 4–18 (2010).
- S3. Axelsson, J., Papatheocharous, E., Andersson, J.: Characteristics of software ecosystems for Federated Embedded Systems: A case study. *Information and Software Technology*, Volume 56, Issue 11, pp. 1457–1475 (2014).
- S4. Baars, A., Jansen, S.: A framework for software ecosystem governance. In Proceedings of the Third International Conference on Software Business (ICSOB'12), pp. 168–180 (2012).
- S5. Bosch, J., Bosch-Sijtsema, M.P.: Esao: A Holistic Ecosystem-Driven Analysis Model. In Proceedings of the 5 th International Conference on Software Business (ICSOB 2014), Springer Verlag, pp. 179–193 (2014).
- S6. Burkard, C., Widjaja, T., Buxmann, P., Software ecosystems. *Business & Information Systems Engineering*, volume 4, Issue. 1, pp. 41–44 (2012).
- S7. Campbell, P., Ahmed, F.: A three-dimensional view of software ecosystems. In Proceedings of the Fourth European Conference on Software Architecture: Companion Volume (ECSA '10), CarlosE. Cuesta (Ed.). ACM, New York, NY, USA, pp. 81–84 (2010).
- S8. Ceccagnoli, M., Forman, C., Huang, P., Wu, D.J.: Co-creation of value in a platform ecosystem: The case of enterprise software. *MIS Quarterly*, Forthcoming (2011).
- S9. Che, M., Perry, D.E.: Architectural Design Decisions in Open Software Development: A Transition to Soft/ware Ecosystems. In Proceedings of the 23rd Australian Software Engineering Conference, pp. 58–61 (2014).

- S10. Den Hartigh, E., Tol, M., Visscher, W.: The health measurement of a business ecosystem. In proceedings of the European Network on Chaos and Complexity Research and Management Practice Meeting. pp. 1–39 (2006).
- S11. Dittrich, Y.: Software engineering beyond the project – Sustaining software ecosystems. *Information and Software Technology*, Volume 56, Issue 11, pp. 1436–1456 (2014).
- S12. Dubinsky, Y., Kruchten, P.: Software development governance (sdg): report on 2nd workshop. *ACM SIGSOFT Software Engineering Notes*, Volume 34, pp. 46–47 (2009).
- S13. Eisenmann, T.R., Parker, G., M.W.; Van Alstyne, M.W.: Opening platforms: how, when and why? In *Platforms, Markets and Innovation*. Gawer A (ed). Edward Elgar: Cheltenham, UK, pp. 131–162 (2008).
- S14. Eklund, U., Bosch, J.: Introducing software ecosystems for mass-produced embedded systems. In *Software Business*. Springer Berlin Heidelberg, pp. 248– 254 (2012).
- S15. Fricker, S.: Requirements Value Chains: Stakeholder Management and Requirements Engineering in Software Ecosystems. In proceedings of Requirements Engineering: Foundation for Software Quality (REFSQ'10), volume 6182, pp. 60–66 (2010).
- S16. Fricker, S.: Specification and Analysis of Requirements Negotiation Strategy in Software Ecosystems. In proceedings of Workshop on Software Ecosystems, pp. 19–33 (2009).
- S17. Gawer, A., Cusumano, M.A.: Industry platforms and ecosystem innovation. *Journal of Product Innovation Management*, Volume 31, Issue 3, pp. 417–433 (2014).
- S18. Gawer, A., Henderson, R.: Platform owner entry and innovation in complementary markets: Evidence from Intel. *Journal of Economics & Management Strategy*, Volume 16, Issue 1, pp. 1–34 (2007).
- S19. Gawer, A., Cusumano, M.A.: Platform leadership: How Intel, Microsoft, and Cisco drive industry innovation. Harvard Business School Press, pp. 29–30 (2002).
- S20. Ghazawne, A., Henfridsson, O.: Governing third-party development through platform boundary resources. In proceedings of International Conference on Information Systems (ICIS' 10), AIS Electronic Library (AISeL), pp 1–18 (2010).

- S21. Goldbach, T., Kemper, V.K.: Should I Stay or should I go? the effects of control mechanisms on app developers' intention to stick with a platform. In proceedings of European Conference on Information Systems (ECIS'S3) (2014).
- S22. Haile, N., Altmann, J.: Value creation in software service platforms. *Future Generation Computer Systems*, Volume 55, pp. 495–509 (2016).
- S23. Hyrynsalmi, S., Suominen, A., Makila, T., Jarvi, A., Knuutila, T.: Revenue models of application developers in android market ecosystem. In M. Cusumano, B. Iyer, N. Venkatraman, eds.: *ICSOB 2012. Number 114 in Lecture Notes in Business Information Processing*. Springer Heidelberg (2012).
- S24. Huang, P., Ceccagnoli, M., Forman, C., Wu, D.J.: Appropriability mechanisms and the platform partnership decision: Evidence from enterprise software. *Management Science*, Vol 59, Issue 1 (2013).
- S25. Iansiti, M., Levien, R.: Strategy as ecology. *Harvard business review*. Vol 82, Issue 3 (2004).
- S26. Iyer, B., Lee, C.H., Venkatraman, N.: Managing a small world ecosystem: Some lessons from the software sector. *California Management Review*, Volume 48, Issue 3, pp. 28–47 (2006).
- S27. James, W.P., and De Meyer, A.: Ecosystem advantage: how to successfully harness the power of partners. *California Management Review*, Volume 55, Issue 1 (2012): 24–46 (2012).
- S28. Jansen, S., Cusumano, M.: Defining Software Ecosystems: A Survey of Software Platforms and Business Network Governance. In *Proceedings of the 4th International Workshop on Software Ecosystems (IWSECO '12)*, pp. 41–58 (2012).
- S29. Jansen, S., Finkelstein, A., Brinkkemper, S.: A sense of community: A research agenda for software ecosystems. In *Proceedings of the 31st Conference on Software Engineering*, pp. 187–190 (2009).
- S30. Jansen, S., Brinkkemper, S., Finkelstein, A.: Business Network Management as a Survival Strategy: A Tale of Two Software Ecosystems. In *proceedings of the 1st International Workshop on Software Ecosystems (IWSECO'09)*. pp. 34–48 (2009).
- S31. Jansen, S., Brinkkemper, S., Souer, J., Luinenburg, L.: The open software enterprise model: how open is my software business? In *Software Ecosystems: Analyzing and*

- Managing Business Networks in the Software Industry. UK: Edward Elgar Publishing, pp. 159–186 (2013).
- S32. Jansen, S., Brinkkemper, S., Souer, J., Luinenburg, L.: Shades of gray: Opening up a software producing organization with the open software enterprise model. *Journal of System and Software*. Volume 85, Issue 7, pp. 1495–1510 (2012).
- S33. Jansen, S.: Opening the Ecosystem Flood Gates: Architecture Challenges of Opening Interfaces Within a Product Portfolio. In proceedings of European Conference on Software Architecture (ECSA'15), Springer International Publishing, pp. 121–136 (2015).
- S34. Jansen, S.: Measuring the health of open source software ecosystems: Beyond the scope of project health. *Information and Software Technology*, Volume, 56, Issue 11, pp. 1508–1519 (2014).
- S35. Kapoor, R., Lee, J.M.: Coordinating and competing in ecosystems: How organizational forms shape new technology investments. *Strategic Management Journal*, Volume 34, Issue 3, pp. 274–296 (2013).
- S36. Kim, H.J., Kim, L., Lee, H.: Third-party mobile app developers' continued participation in platform-centric ecosystems: An empirical investigation of two different mechanisms. *International Journal of Information Management*, Volume 36, Issue 1, pp. 44–59 (2016).
- S37. Knauss, E., Damian, D., Knauss, A., Borici, A.: Openness and requirements: Opportunities and tradeoffs in software ecosystems. In *Proceedings of the IEEE 22nd International Requirements Engineering Conference (RE'S3)*, 2014, pp. 213–222 (2014).
- S38. Koch, S., Kerschbaum, M.: Joining a smartphone ecosystem: Application developers' motivations and decision criteria. *Information and Software Technology*, Volume 56, Issue 11, pp. 1423–1435 (2014).
- S39. Manikas, K., Klaus, M.H., and Kyng, M.K.: Governance mechanisms for healthcare apps. In *Proceedings of the 2014 European Conference on Software Architecture Workshops*. ACM, p. 10 (2014).
- S40. Mizushima, K., Ikawa, Y.: A structure of cocreation in an open source software ecosystem: A case study of the eclipse community. In *Proceedings of Technology Management in the Energy Smart World (PICMET'11)*. pp. 1–8 (2011).

- S41. Monteith, J.Y., McGregor, J.D., Ingram, J.E.: Proposed metrics on ecosystem health. In Proceedings of the 2014 ACM international workshop on Softwaredefined ecosystems (BigSystem'S3), ACM, pp. 33–36 (2014).
- S42. Mukhopadhyay, S., de Reuver, M., and Bouwman, H.: Effectiveness of control mechanisms in mobile platform ecosystem. *Telematics and Informatics*, Volume 33, Issue 3, pp. 848–859 (2016).
- S43. Olsson, H., and Bosch, J. Strategic Ecosystem Management: A multi-case study on challenges and strategies for different ecosystem types. In proceedings of Software Engineering and Advanced Applications (SEAA'15), pp. 398–401 (2015).
- S44. Pilinkienė, V., Mačiulis, P.: Comparison of Different Ecosystem Analogies: The Main Economic Determinants and Levels of Impact. *Procedia – Social and Behavioral Sciences*, Volume 156, Issue 26, pp. 365–370.
- S45. Popp, K.M.: Goals of Software Vendors for Partner Ecosystems – A Practitioner's View. In proceedings of First International Conference on Software Business (ICSOB'10). Springer: pp. 181–186 (2010).
- S46. Sadi, M.H., Yu, E.: Designing Software Ecosystems: How Can Modeling Techniques Help? In proceedings of Enterprise, Business-Process and Information Systems Modeling. Springer International Publishing, pp. 360–375 (2015).
- S47. Sadi, H., Yu, E.: Analyzing the evolution of software development: From creative chaos to software ecosystems. In Proceedings of the IEEE Eighth International Conference on Research Challenges in Information Science (RCIS'S3), pp. 1–11 (2014).
- S48. Santos, R., Werner, C.: Treating social dimension in software ecosystems through ReuseECOS approach. In Proceedings of the 6th IEEE International Conference on Digital Ecosystems Technologies (DEST'12), pp. 1–6 (2012).
- S49. Schönberger, A., Elsner, C.: Modeling Partner Networks for Systematic Architecture Derivation in Software Ecosystems. In: Linnhoff-Popien, Claudia; Zaddach, Michael; Grahl, Andreas (Ed.): *Marktplätze im Umbruch – Digital e Strategien für Services in Mobilen Internet*. Springer Berlin Heidelberg, pp. 655–665 (2015).
- S50. Schultis, K.B., Elsner, C., Lohmann, D.: Architecture challenges for internal software ecosystems: a large-scale industry case study. In Proceedings of the 22nd ACM SIGSOFT

- International Symposium on Foundations of Software Engineering (FSE'S3), ACM, pp. 542–552 (2014).
- S51. Tiwana, A. Konsynski, B. Bush, A.: Platform evolution: Coevolution of platform architecture, governance, and environmental dynamics. *Information Systems Research*. Volume 21, Issue 4, pp. 675–687 (2010).
- S52. Valenca, G., Alves, C., Heimann, V., Jansen, S., Brinkkemper, S.: Competition and collaboration in requirements engineering: A case study of an emerging software ecosystem. In *Proceedings of the IEEE 22nd International Requirements Engineering Conference (RE'S3)*, pp. 384–393 (2014).
- S53. van Angeren, J., Kabbedijk, K., Popp, K., Jansen, S.: Managing software ecosystems through partnering. In *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry*. Cheltenham, UK: Edward Elgar Publishing, pp. 85–102 (2013).
- S54. van Angeren, J., Alves, C., Jansen, S.: Can We Ask You To Collaborate? Analyzing App Developer Relationships in Commercial Platform Ecosystems. *Journal of Systems and Software*, Volume 113, pp. 430–445 (2016).
- S55. van Angeren, J., Jansen, S., Brinkkemper, S.: Exploring the relationship between partnership model participation and interfirm network structure: An analysis of the office365 ecosystem, In *Proceedings of Software Business (ICSOB'S3)*. Springer, pp. 1–15 (2014).
- S56. Ven, K., Mannaert, H.: Challenges and strategies in the use of Open Source Software by Independent Software Vendors. *Information and Software Technology*, Volume 50, Issues 9–10, pp. 991–1002 (2008).
- S57. Viljainen, M., Kauppinen, M.: Framing Management practices for keystones in platform ecosystems. In *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry*. Cheltenham, UK: Edward Elgar Publishing, pp. 121–137 (2013).
- S58. Viljainen, M., Kauppinen, M. *Software Ecosystems: A Set of Management Practices for Platform Integrators in the Telecom Industry*. In *proceedings of 2nd International Conference on Software Business*. Springer Berlin Heidelberg, pp. 32–43 (2011).

- S59. Wareham, J., Fox, P.B., CanoGiner, J.L. Technology ecosystem governance. *Organization Science* Volume 25, Issue 4, pp. 1195–1215 (2014).
- S60. West, J., and Wood, D.: *Creating and Evolving an Open Innovation Ecosystem: Lessons from Symbian Ltd.* Available at SSRN Working Paper Series (2008).
- S61. West, J., and Wood, D. Tradeoffs of Open Innovation Platform Leadership: The Rise and Fall of Symbian Ltd. In *proceedings of Social Science and Technology Seminar Series* (2011).
- S62. Wnuk, K., Manikas, K., Runeson, P., Lantz, M., Weijden, O., Munir, H.: Evaluating the governance model of hardware-centric software ecosystems: a case study of the axis ecosystem. In *proceedings of 5th International Conference on Software Business*, pp. 212–226 (2014).
- S63. Yu, E., Deng, S.: *Understanding Software Ecosystems: A Strategic Modelling Approach.* In *International Workshop on Software Ecosystems (IWSECO'10)* (2010)
- S64. Aarnoutse, F., Renes, C., Snijders, R., and Jansen, S.: *The Reality of an Associate Model: Comparing Partner Activity in the Eclipse Ecosystem.* In *Proceedings of the 2014 European Conference on Software Architecture Workshops (ECSAW '14)*, ACM, pp. 6–8. (2014).
- S65. Alami, D., Rodríguez, M., and Jansen, S.: *Relating Health to Platform Success: Exploring Three E-commerce Ecosystems.* In *Proceedings of the 2015 European Conference on Software Architecture Workshops (ECSAW '15)*, ACM, p. 43. (2015).
- S66. Ben Hadj Salem Mhamdia, A.: *Performance measurement practices in software ecosystem.* *International Journal of Productivity and Performance Management*, Volume 62, Issue 5, pp. 514–533. 2013.
- S67. Cardoso, J.L., Barbin, S.E., and Silva Filho, O.S.: *The public software ecosystem: exploratory survey.* In *Proceedings of the Fifth International Conference on Management of Emergent Digital EcoSystems (MEDES '13)*, ACM, pp. 289–296 (2013).
- S68. Costanza, R.: *Toward an operational definition of ecosystem health. Ecosystem health: New goals for environmental management*, Island Press, pp. 239–256 (1992).
- S69. Den Hartigh, E., Visscher, W., Tol, M., and Salas, A.: *Measuring the health of a business ecosystem.* In *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry*. Cheltenham, UK: Edward Elgar Publishing, 221–246 (2013).

- S70. Dhungana, D., Groher, I., Schludermann, E., and Biffel, S.: Software ecosystems vs. natural ecosystems: learning from the ingenious mind of nature. In Proceedings of the Fourth European Conference on Software Architecture: Companion Volume (ECSA '10), Carlos E. Cuesta (Ed.). ACM, New York, NY, USA, pp. 96–102 (2010).
- S71. Eckhardt, E., Kaats, E., Jansen, S., and Alves, C.: The Merits of a Meritocracy in Open Source Software Ecosystems. In Proceedings of European Conference on Software Architecture Workshops (ECSAW '14), ACM, p. 7 (2014).
- S72. Eklund U., and Bosch, J.: Using architecture for multiple levels of access to an ecosystem platform. In Proceedings of the 8th international ACM SIGSOFT conference on Quality of Software Architectures (QoSA '12), ACM, pp. 143–148 (2012).
- S73. German, D. M., Adams, B., Hassan, A. E.: The Evolution of the R Software Ecosystem. In Proceedings of the 17th European Conference on Software Maintenance and Reengineering (CSMR'13), pp. 243–252 (2013).
- S74. Hoving, R., Slot, G., Jansen, S.: Python: Characteristics identification of a free open source software ecosystem. In Proceedings of the 7th IEEE International Conference on Digital Ecosystems and Technologies (DEST'13), pp. 13–18 (2013).
- S75. Hyrynsalmi, S., Seppänen, M., Suominen, A.: Sources of value in application ecosystems. *Journal of Systems and Software*, Volume 96, pp. 61–72 (2014).
- S76. Iansiti, M., and Richards, G.L.: Information technology ecosystem: Structure, health, and performance. *The Antitrust Bull.*, Volume 51, p. 77 (2006).
- S77. Iansiti, M., and Levien, R.: Keystones and Dominators: Framing Operating and Technology Strategy in a Business Ecosystem. Harvard Business School, Working Paper, p. 03–061 (2004).
- S78. Kabbedijk, J., and Jansen, S.: Unraveling Ruby ecosystem dynamics: a quantitative network analysis. In *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry*. Cheltenham, UK: Edward Elgar Publishing, pp. 322–332 (2013).
- S79. Kilamo, T., Hammouda, I., Mikkonen, T and Aaltonen, T.: Open source ecosystems: a tale of two cases. In *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry*. Cheltenham, UK: Edward Elgar Publishing, pp. 276–206 (2013).

- S80. Lucassen, G., van Rooij, K., Jansen, S.: Ecosystem health of cloud paas providers. In *Software Business. From Physical Products to Software Services and Solutions*. Springer Berlin Heidelberg, pp. 183–194 (2013).
- S81. Lundell, B., Forssten, B., Gamalielsson, J., Gustavsson, H., Karlsson, R., Lennerholt, C., and Olsson, E.: Exploring health within OSS ecosystems. In *proceedings of the First International Workshop on Building Sustainable Open Source Communities (OSCOMM'09)* (2009).
- S82. Manikas, K., and D., Kontogiorgos, D.: Characterizing Software Activity: The Influence of Software to Ecosystem Health. In *Proceedings of the 2015 European Conference on Software Architecture Workshops (ECSAW '15)*, ACM, p. 46 (2015).
- S83. Manikas, K and Hansen, K.M. Reviewing the Health of Software Ecosystems-A Conceptual Framework Proposal. In *proceedings of the 5th International Workshop on Software Ecosystems (IWSECO'13)*, in ICSOB, pp. 33–44 (2013).
- S84. McGregor, D. J.: A method for analyzing software product line ecosystems. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume (ECSA'10)*, Carlos E. Cuesta (Ed.). ACM, New York, NY, USA, pp. 73–80. 2010.
- S85. Spauwen, R., and Jansen, S.: Towards the roles and motives of open source software developers. In *Proceedings of the 5th International Workshop on Software Ecosystems (IWSECO'13)*, p. 62 (2013).
- S86. Stefi, A., Berger, M., and Hess, T.: What Influences Platform Provider's Degree of Openness?—Measuring and Analyzing the Degree of Platform Openness. In *proceedings of Software Business. Towards Continuous Value Delivery*. Springer International Publishing, pp. 258–272 (2014).
- S87. Syed, S and Jansen, S.: On Clusters in Open Source Ecosystems. In *proceedings of International Workshop on Software Ecosystems (IWSECO'13)*, in ICSOB, pp. 19–32. (2013).
- S88. Teixeira, J., Robles, G., González-Barahona, J.M.: Lessons learned from applying social network analysis on an industrial Free/Libre/Open Source Software ecosystem. *Journal of Internet Services and Applications*, Volume 6, Issue 1, pp. 1–27 (2015).

- S89. Van Lingen, S., Palomba, A., and Lucassen, G.: On the software ecosystem health of open source content management systems. In proceedings of the 5th International Workshop on Software Ecosystems (IWSECO 2013), p. 38 (2013).